rollNW

unknown

May 19, 2024

GETTING STARTED

rollNW is an homage to Neverwinter Nights in C++ and Python.

See the docs and tests for more info, or open an IDE in browser in the quickstart section below.

This library is a work-in-progress. There will be serious refactoring and until there is a real release, it should be assumed the library is in flux.

CHAPTER

ONE

FEATURES

- The beginnings of a novel Rules System designed for easily adding, overriding, expanding, or removing any rule and reasonable performance
- A combat engine that's very nearing being able to simulate melee battles.
- Objects (i.e. Creatures, Waypoints, etc) are implemented at a toolset level. Or in other words their features cover blueprints, area instances, with support for effects and item properties. They are still missing some new EE things. Player Characters are read only, for now.
- A recursive decent NWScript Parser
- Implementations of pretty much every NWN File Format
- An binary and ASCII Model Parser. See also the mudl model viewer side project.
- A Resource Manager that can load all NWN containers (e.g. erf, key, nwsync) and also Zip files.
- An implementation of NWN's Localization System focused on utf8 everywhere.

CHAPTER

TWO

GOALS

- aims to implement an RPG engine inspired by NWN, excluding graphics and networking.
- focuses on usage, instead of doing things the Aurora Engine Way.
- follows utf8 everywhere.
- hews as close to C++ Core Guidelines as possible.
- aims to be as easily bindable as possible to other languages. I.e. only library specific or STL types at API boundaries.

CHAPTER

THREE

QUICKSTART - OPEN VS CODE IN YOUR BROWSER

Github Codespaces is available to those in the beta.

CHAPTER

FOUR

HISTORY

A lot of what's here was written in the 2011-2015 range as part of personal minimalist toolset, modernized and with new EE stuff added. In some sense, it's a work of historical fiction - it's what I'd have suggested at the start of NWN:EE: get the game and the community on the same set of libraries. Similarly to an older project that asked "what if Bioware had stuck with Lua?". The answer to that was pretty positive: a decade ahead, at least, of nwscript.

CHAPTER

FIVE

MOONSHOTS

You make ask yourself, why? But, to paraphrase Tennyson, ours isn't to question why, it's but to do and die and learn and maybe make neat things. In that spirit, here is a list of crazy projects that this library hopes to facilitate and that all fly in the face of "WHY?".

- A nwscript Language Server
- A modern, cross-platform nwexplorer.
- And, of course, the ever elusive open source NWN Toolset, with plugins, scripting, and a built-in console.

CHAPTER

SIX

CREDITS

- Bioware, Beamdog - The game itself
- abseil - Foundational
- Catch2 - Testing
- glm - Mathematics
- loguru, fmt - Logging
- stbi_image, NWNExplorer, SOIL2 - Image/Texture loading.
- inih - INI, SET parsing
- nholmann_json - JSON
- toml++ - For settings.tml
- libiconv, boost::nowide - i18n, string conversion
- doxygen, sphinx, breathe - documentation

6.1 building

rollnw uses cmake as its build system and more specifically CMakePresets.json.

To build the library, all one needs to do is use the following cmake commands. This example also builds tests which are not enabled by default.

linux

macOS

windows

```
$ cd path/to/rollnw
$ cmake --preset linux -DROLLNW_BUILD_TESTS=ON
$ cmake --build --preset default
$ ctest --preset default
```

Note: The deployment target is currently set to 12.

```
$ cd path/to/rollnw
$ cmake --preset macos -DROLLNW_BUILD_TESTS=ON
$ cmake --build --preset default
$ ctest --preset default
```

The main windows builds are for Visual Studio 2022 (Community Edition), but mingw-64, later versions of Visual Studio will added.

For now it's probably easiest to open the x64 Visual Studio Developer Command Prompt. If you have ninja installed, it's highly recommended to use the `windows-ninja` configuration preset.

```
$ cd path/to/rollnw
$ cmake --preset windows -DROLLNW_BUILD_TESTS=ON
$ cmake --build --preset default
$ ctest --preset default
```

In order to run all of the tests, you can help the library locate Neverwinter Nights installation paths by exporting the following ENV vars, if your install is in a non-standard location:

Linux / MacOS

Windows

```
$ export NWN_ROOT=path/to/game
$ export NWN_USER=path/to/nwn-user-dir
```

```
set NWN_ROOT=path\to\game
set NWN_USER=path\to\nwn-user-dir
```

6.2 using

While the library is far from done, basic usage would be as follows.

Python

C++

```
pip install rollnw
```

```
import rollnw

rollnw.kernel.start()
mod = rollnw.kernel.load_module("mymodule.mod")
for area in mod:
    # Do neat things
```

```
#include <nw/kernel/Kernel.hpp>
#include <nw/log.hpp>

int main(int argc, char* argv[])
{
    // Initialize logger
    nw::init_logger(argc, argv);
```

(continues on next page)

(continued from previous page)

```
    // Say this application is specific to 1.69.
    // This must be set before the initialize call below.  The default is NWN:EE, so in␣
↪that case,
    // ``set_version`` need not be called.. NOTE: THIS also controls which profile is␣
↪loaded..
    nw::kernel::config().set_version(GameVersion::v1_69);

    // Sets config for the system, paths, version, etc.
    nw::kernel::config().initialize();

    // Initializes all systems
    nw::kernel::services().start();

    auto mod = nw::kernel::load_module("path/to/modules/your_module.mod");

    // Do neat stuff

    nw::kernel::unload_module();

    return 0;
}
```

6.3 differences

6.3.1 Changed

1. All resource names (i.e. resrefs, extensions) and resource paths are coerced to lower-case. On macOS, Windows, this generally makes no difference. On Linux, converting filenames, paths, etc, to lower-case has always been the best policy.

6.3.2 Unsupported

1. NWN(:EE or 2) configuration files for a couple reasons:
   - NWN:EE introduced a lot of needless complexity with `settings.tml` and it wasn't a particularly good choice to begin with.
   - If values are necessary they can be read first by some consumer of the library.
2. The concept of path aliases, i.e. "HAK:", "HD0:", etc.

6.4 formats

One thing that makes NWN(:EE) so great is that it's a local optima of power and simplicity. It's file formats are no different many of them - at the reader level - can be implemented as a thin wrapper over a handful of casts.

Where necessary the reading will be implemented separately from the writing and likewise from the 'rithmatic, unless it doesn't affect the usage or performance characteristics of one or the other. E,g. tlk can easily be made read/write/modifiable with the exact same performance characteristics, Gff cannot. It's better to separate them than convolute the implementations of all of them.

6.4.1 2da

Status: read/write

The 2da parser is one of the more optimized parts of the library. It can parse ~300MBps and all default 2das in under half of a second.

Example - Load a TwoDA:

Python

C++

```python
#! /usr/bin/env python

from rollnw import TwoDA
import os


for f in os.listdir():
    if os.path.isfile(f) and os.path.splitext(f)[1].lower() == '.2da':
        tda = TwoDA(f)
        if not tda.valid():
            print(f"failed parsing: {f}")
            continue
        print(tda[0][0])
```

```cpp
nw::TwoDA feat("feat.2da");
if(feat.is_valid()) {
    std::cout << fmt::format("feat.2da has {} rows", feat.rows()) << "\n";
    std::cout << *feat.get<std::string_view>(4, 0) << "\n";
    std::cout << *feat.get<int32_t>(0, "FEAT") << "\n";
}
```

6.4.2 gff

See serialization docs

6.4.3 images formats

Status:

- bmp: read/write
- dds (standard): read/write
- dds (bioware): read/write
- jpg: read/write
- gif: read/write
- plt: unsupported
- png: read/write
- tga: read/write

The library can do conversions between image formats and can do anything that stbi_image can, however, this has no goal of being any kind of useful conversion library.

bmp, gif, jpg, png, tga are supported thanks to stbi_image. dds (standard & bioware) is supported thanks to SOIL2.

Python

C++

```
from rollnw import Image
img = Image("my_texture.dds") # Can be Bioware DDS or standard
img.write_to("my_texture.png")
```

```
// TODO
```

6.4.4 ini

Status: read

Supported thanks to inih

Python

C++

```
from rollnw import Ini
ini = Ini("userpatch.ini")
if ini.get_str("Patch/PatchFile000"):
    # User has patch files defined
    pass
```

```
// TODO
```

6.4.5 json

Status: read/write

Supported thanks to nholmann_json

6.4.6 mdl

See model docs

6.4.7 mtr

Status: unsupported

Python

C++

```
# TODO
```

```
// TODO
```

6.4.8 set

Status: read

Supported thanks to inih

Python

C++

```
# TODO
```

```
// TODO
```

6.4.9 ssf

Status: unsupported

Python

C++

```
# TODO
```

```
// TODO
```

6.4.10 tml

Status: read/write (c++), unsupported (python)

Supported thanks to toml++.

6.4.11 txi

Status: unsupported

Python

C++

```
# TODO
```

```
// TODO
```

6.5 i18n

The **i18n** module provides support for internationalization, conversions between default NWN encodings and UTF-8 (see below).

6.5.1 Philosophy

The module follows the principles of UTF-8 everywhere. Or in other words, ordinary C++ string types, `std::string`, `std::string_view`, etc. *must* be in UTF-8. The only exception are:

- wide character types (`std::wstring`, `u16string`, `u32string`, etc) which are never used.
- `std::filesystem::path` which is, per the standard, stored in native encoding. E.g., on Linux, UTF-8; on windows, UCS-2; etc. Some platform specific conversions to UTF-8 are therefore necessary.

There is *no* caching or fixed caching policies at this layer of the library.

6.5.2 Neverwinter Nights Languages and Encodings

- English (CP1252)
- French (CP1252)
- German (CP1252)
- Italian (CP1252)
- Spanish (CP1252)
- Polish (CP1250)
- Korean (CP949) - Unsupported by NWN:EE
- Chinese Traditional (CP936) - Unsupported by NWN:EE
- Chinese Simplified (CP950) - Unsupported by NWN:EE
- Japanese (CP932) - Unsupported by NWN:EE

6.5.3 kernel service

The *Strings* kernel service provides access to dialog/custom TLKs and localized strings. It also provides a mechanism for interning commonly used strings.

Example - Intern a String

```
auto str = nw::kernel::strings().intern("This is a Test");
if(str == "This is a Test") {
   // This will occur
}
```

6.6 kernel

The `kernel` module provides submodules for handling global resources and services. It's designed around some explicit goals:

- Every service should be easily overrideable to allow for parallel implementation.
- Every service should be decoupled from the kernel itself.
- Any function or object that can modify global state must be contained in this module for easy search/grepability.

6.6.1 Services

Config

The `Config` service provides access to installation info.

6.7 model

As mentioned elsewhere, the goal of this library is not to render graphics, but maybe? Loading NWN models is for the purpose of conversion or in some other asset pipeline.

See the mudl model viewer side project for work on a NWN model viewer.

6.7.1 ASCII Models

Most default NWN models can be parsed without errors.

The parser can parse at about 100mb/s and can read pretty much all ascii models in ~20s on 2015 MacBook Pro.

6.7.2 Binary Models

The beginnings of binary model parsing is in the library. It's hard to tell what's right and what's wrong until there is more rendering experience.

6.7.3 Examples

Python

C++

```python
#! /usr/bin/env python

from rollnw.model import Mdl
import os

mdl = Mdl.from_file("my_ascii_model.mdl")
if not mdl.valid():
    print(f"failed parsing: {f}")
print(mdl.supermodel_name)
```

```
// TODO
```

6.8 objects

rollNW is all about live objects and *not* serialized file formats.

6.8.1 definitions

ObjectID
Unlike NWN an ObjectID does not provide a one-to-one mapping to an object. Rather, it's an index into a structure containing objects.

ObjectHandle
An object handle maps to a specific object it consists of an ObjectID, the objects type, and an unsigned 16 bit integer indicating the object's version. To be valid, an object handle must match what is in the object array.

ObjectBase
The base class of all objects

6.8.2 kernel service

Any object that is loaded via the Objects service, belongs to the service and must be deleted through it.

Example - Loading and Deleting a Creature

Python

C++

```python
import rollnw

rollnw.kernel.start()
obj = rollnw.kernel.objects().creature('nw_chicken.utc')
rollnw.kernel.objects().destroy(obj.handle())
# After this point accessing obj is undefined behavior and its handle if stored elsewhere
# will no longer be valid
```

```cpp
auto obj = nw::kernel::objects().load<nw::Creature>(fs::path("a/path/to/nw_chicken.utc
↪"));
nw::kernel::objects().destroy(obj->handle());
// After this point accessing obj is undefined behavior and its handle if stored␣
↪elsewhere
// will no longer be valid
```

6.8.3 area

6.8.4 creature

Python

C++

```python
from rollnw import Creature

# The file can also be rollnw JSON format, it doesn't matter.
cre = Creature.from_file("a/path/to/nw_chicken.utc")
if cre.scripts.on_attacked == "nw_c2_default5":
   cre.scripts.on_attacked = "nw_shakenbake"
```

```
// TODO
```

6.8.5 door

6.8.6 encounter

6.8.7 item

6.8.8 module

6.8.9 placeable

6.8.10 sound

6.8.11 store

6.8.12 trigger

6.8.13 waypoint

6.9 resources

6.9.1 kernel service

The resource services provides access the file system and resources stored in NWN container files. The main goals mostly satisfied: the ability to read all NWN(:EE) containers. The ability to add new container types is limited in utility due to NWNX:EE's lack of a ResMan plugin, even so the ability to load files from a Zip file is included.

Example - Demanding a resource from resman

Python

C++

```python
import rollnw
from rollnw.kernel import resman

rollnw.kernel.start()
assert resman().contains('nw_chicken.utc')
data = resman().demand('nw_chicken.utc')
data2 = resman().demand(rollnw.Resource('nw_chicken', rollnw.ResourceType.utc))
assert data == data2
```

```cpp
nw::kernel::start();
// Assumes that NWN root directory was found.
if (nw::kernel::resman().contains({"nw_chicken"sv, nw::ResourceType::utc})) {
    auto utc = nw::kernel::resman().demand({"nw_chicken"sv, nw::ResourceType::utc});
    // Do something with this chicken.
}
```

6.9.2 containers

Directory

Status: Read, Write (file format dependant)

Erf

Status: Read

Example - Load an Erf and Print Contents

Python

C++

```
import rollnw

erf = rollnw.Erf("tests/test_data/user/hak/hak_with_description.hak")
print(erf.name(), erf.size())
for rd in erf.all():
    print(rd.name.filename(), rd.size)
```

```
#include <nw/resources/Erf.hpp>
// ...
Erf e("MyModule.mod");
if (e.valid()) {
    std::cout << fmt::format("{} has {} resources", e.name(), e.size()) << "\n";
    for (const auto& rd : e.all()) {
        std::cout << fmt::format("File: {}, Size: {}", rd.name.filename(), rd.size) << "\
↪n";
    }
}
```

Key/Bif

Status: Read

NWSync

Status: Read

Example - Loading and Reading From NWSync Manifest

Python

C++

```
import rollnw

nws = rollnw.NWSync("path/to/nwsync")
if nws.is_loaded():
    # One of the curated modules
```

(continues on next page)

(continued from previous page)

```
    manifest = nws.get('9a84e512dd3971eb215d6f9b0816a2e3ae2fee54')
    if manifest:
        tga = manifest.demand('002metal.tga')
        # Do something with this image..
```

```
#include <nw/resources/NWSync.hpp>
#include <nw/kernel/Kernel.hpp>

auto path = "path/to/nwsync"
auto n = nw::NWSync(path);
if(!n.is_loaded()) {
    throw std::runtime_error("a fit");
}

auto manifests = n.manifests();
if (manifests.size() > 0) {
    auto manifest = n.get(manifests[0]);
    auto resource = manifest->all();
    if(resource.size() > 0) {
        // Extract the first resource found
        manifest->extract(std::regex(resource[0].name.filename()), "tmp/");
    }
}
```

Zip

Status: Read

6.10 rules

The `Rules` module presents some difficulties in the sense that if one was to sit down and design a system capable of expressing relatively arbitrary sets of rules and modifiers, it probably would not look much like NWN. Enhanced Edition's approach largely was to unhardcode *values*, but not systems[1].

rollNW has the elements of NWN's rule system builtin, which is itself an approximation of the Dungeons and Dragons 3rd Edition ruleset.

6.10.1 The Goals

- Rules must be overridable, expandable, removable either through configuration (2da) or at the very least programmatically. Nothing should be hardcoded.
- The rules system must be queryable. Example: Given one creature attacking one chair with one handaxe in one bar of Chicago, what are all the modifiers that affect this particular situation?
- Ideally, constants would be disassociated from 2da rows. Say a UUID <-> integer map, but that's both a configuration and serialization problem.

[1] There are some exceptions, parts of the custom spellcaster system.

6.10.2 Definitions

Profile
Profiles are a way of decoupling different rulesets from the rule system itself.

Type
A rule type is an attribute of the rule system, say a skill or an ability or a damage. The rule system defines the type and its invalid case, but leaves valid cases up to the rule profile. An example, armor class:

```
// Definition of an attribute in nw/rules/attributes.hpp
DECLARE_RULE_TYPE(ArmorClass)

// Somewhere else in a rule profile that uses the concept of armor class
constexpr nw::ArmorClass ac_dodge = nw::ArmorClass::make(0);
constexpr nw::ArmorClass ac_natural = nw::ArmorClass::make(1);
constexpr nw::ArmorClass ac_armor = nw::ArmorClass::make(2);
constexpr nw::ArmorClass ac_shield = nw::ArmorClass::make(3);
constexpr nw::ArmorClass ac_deflection = nw::ArmorClass::make(4);

// Then it's possible to refer to them as some opaque value for type safety:
auto res = get_armor_class(object, ac_shield);

// Or as their underlying value:
switch(*ac_type) {
case *ac_dodge: // ..
case *ac_natural: // ..
case *ac_armor: // ..
case *ac_shield: // ..
case *ac_deflection: // ..
}

// Or if it makes logical sense to think of a particular type as an index:
obj->ac_bonuses[ac_dodge.idx()]
```

Flag
nw::RuleFlag provides a mechanism for making flags out of rule types.

6.10.3 Modifiers

The foundation of the modifier system is just three types: `int32_t`, `float`, strings. It builds on the following abstractions to provide a dynamic, modifiable, queryable system. Modifiers are stored in a global table in *nw::kernel::Rules*. Note that Master Feat modifiers are special cased below.

The approach here is inspired by Solstice and Orth's NWNX:EE plugins Race, SkillRank, and Feat.

Note that the examples below are designed for simplicity, not things that should necessarily be done.

Definitions

Modifier Type
A modifier type is a rule type that is used to determine how to process the outputs of a modifier.

Modifier Source
A modifier source indicates the attribute of an object that modifier is associated with.

Modifier Input
An input is an `int`, `float`, or a version of a `ModifierFunction`[2].

Modifier Output
In the basic cases, an output is the input passed directly without modification. When a function is the modifier input, it is called and its result is the output

The output is then passed to a callback provided to one of the `nw::kernel::resolve_modifier` function overloads.

The meaning of these outputs are determined by the modifier type. The number of output parameters is limited to one. They currently have to be integer, floating point types, or *`nw::DamageRoll`*.

In most cases using `nw::kernel::sum_modifier` or `nw::kernel::max_modifier` can avoid having to deal with passing callbacks.

Example - Adding a Modifier:

```
// This is just an example, see "profiles/nwn1/modifiers.[ch]pp for real implementations␣
↪of rules.
auto mod2 = nwn1::mod::hitpoints(
   20, // Modifier value, if the below requirement is met
   "dnd-3.0-epic-toughness-01",
   nw::ModifierSource::feat
   { nwn1::qual::feat(nwn1::feat_epic_toughness_1) },
);

// Add it to the global modifier table
nw::kernel::rules().modifier.add(mod2);
```

Example - Pale Master Armor Class Bonus:

```
auto mod = nwk::load_module("test_data/user/modules/DockerDemo.mod");
auto ent = nwk::objects().load<nw::Creature>(fs::path("some/palemaster.utc"));

int res = 0;
nwk::resolve_modifier(ent, nwn1::mod_type_armor_class, nwn1::ac_natural,
   [&res](int value) { res += value; });
// res == 6

auto pm_ac_nerf = [](const nw::ObjectBase* obj) -> nw::ModifierResult {
   auto cre = obj->as_creature();
   if (!cre) { return 0; }
   auto pm_level = cre->levels.level_by_class(nwn1::class_type_pale_master);
   return ((pm_level / 4) + 1);
};
```

(continues on next page)

[2] One could imagine in a different context, say NWNX:EE, you could add a callback to nwnx_dotnet/lua/etc or a string for use with `ExecuteScriptChunk`.

(continued from previous page)

```
// Get rid of any requirement
nwk::rules().modifiers.replace("dnd-3.0-palemaster-ac", nw::Requirement{});
// Set nerf
nwk::rules().modifiers.replace("dnd-3.0-palemaster-ac", pm_ac_nerf);
res = 0;
REQUIRE(nwk::resolve_modifier(ent, nwn1::mod_type_armor_class, nwn1::ac_natural,
   [&res](int value) { res += value; }));
// res == 3

res = 0;
nwk::resolve_modifier(ent, nwn1::mod_type_armor_class, nwn1::ac_natural,
   [&res](int value) { res += value; });
// res == 0
```

6.10.4 Master Feats

Master feats and associated bonuses are set in the *nw::MasterFeatRegistry*. The master feat registry associates a particular rule element, say, a skill with a master feat and a feat corresponding to that skill.

Example - (Epic) Skill Focus: Discipline

```
auto& mfr = nw::kernel::rules().master_feats();
mfr->set_bonus(mfeat_skill_focus, 3);
mfr->set_bonus(mfeat_skill_focus_epic, 10);

mfr->add(skill_discipline, mfeat_skill_focus, feat_skill_focus_discipline);
mfr->add(skill_discipline, mfeat_skill_focus_epic, feat_epic_skill_focus_discipline)
```

Multiple feats are able to be associated with a rule element and masterfeat. Imagine in some universe, there is a class that has access to a generic Weapon Focus: Martial feat which provides Weapon Focus for all martial weapons.

Example - Multiple Associated Feats

```
auto& mfr = nw::kernel::rules().master_feats;
// Set up bonuses...
mfr->set_bonus(mfeat_weapon_focus, 1);
mfr->set_bonus(mfeat_weapon_focus_epic, 2);

// Register feats
mfr.add(baseitem_longsword, mfeat_weapon_focus, feat_weapon_focus_longsword);
mfr.add(baseitem_longsword, mfeat_weapon_focus, feat_weapon_focus_martial);
mfr.add(baseitem_longsword, mfeat_weapon_focus_epic, feat_epic_weapon_focus_longsword);
mfr.add(baseitem_longsword, mfeat_weapon_focus_epic, feat_epic_weapon_focus_martial);

// Process
auto callback = [](int value) { /* do something with value */ };
nw::kernel::resolve_master_feats<int>(cre, baseitem, callback,
   mfeat_weapon_focus, mfeat_weapon_focus_epic);

// Simple sums of master feat bonuses can be done as below.
```

(continues on next page)

(continued from previous page)

```
int value = nw::kernel::sum_master_feats<int>(cre, baseitem,
   mfeat_weapon_focus, mfeat_weapon_focus_epic);

// If you are only interested in resolving one master feat you can get that result
// directly:
int value2 = nw::kernel::resolve_master_feat<int>(cre, baseitem, mfeat_weapon_focus);
```

6.10.5 Requirements

Selector

A selector gets some piece of information from an entity.

Example:

```
auto s = nwn1::sel::ability(ability_strength);
// ...
auto str = nw::kernel::rules().select<int>(s, entity);
// ...
```

Qualifier

A qualifier is a selector with some constraints thereupon. In the example below any creature with an unmodified strength between [20, 40] inclusive would match.

```
auto q = nwn1::qual::ability(ability_strength, 20, 40);
// ...
if(nw::kernel::rules().match(q, creature)) {
   // ...
}
```

Requirement

A requirement is just a set of one or more Qualifiers.

Example:

Some thing a has requirement of level 4, wisdom between [12, 20], and a minimum appraise skill of 6.

```
auto req = nw::Requirement{{
   nwn1::qual::level(4),
   nwn1::qual::ability(ability_wisdom, 12, 20), // Min, Max
   nwn1::qual::skill(skill_appraise, 6),
}};
// ...
if(nw::kernel::rules().meets_requirement(req, creature)) {
   // ...
}
```

By default a requirement uses logical conjunction, to use disjunction pass `false` at construction.

```
auto req = nw::Requirement{{
   // Qualifiers ...
}, false};
```

6.11 script

The script module provides a lexer, recursive decent parser, and type-checker for NWScript.

Note: In the case of the Python API, the interface to the AST is read only.

6.11.1 examples

Basic Loading

Python

C++

```python
import rollnw
from rollnw.script import Nss, Context

# Start kernel, if you want to load game assets
rollnw.kernel.start()

# Create a context and to add include path, pass them into the Context constructor
ctx = Context(["includes/"])

# Load the script from a file
nss = Nss("path/to/myscript.nss", ctx)

# Parse
nss.parse()

# Preprocessing
nss.process_includes()

# Now all dependencies are available
deps = nss.dependencies()

# Ast resolution and type-checking
nss.resolve()

# Load a script from string
nss2 = Nss.from_string("void test_function(string s, int b);", ctx)

# To get any old script in the the context's resman use ``get``.  Note this
# parses and resolves the script, nothing further processing is needed.
raise_dead = ctx.get("nw_s0_raisdead")
```

```cpp
#include <nw/kernel/Kernel.hpp>
#include <nw/script/Nss.hpp>

// Start the kernel, if you want to load game assets
nw::kernel::config().initialize();
nw::kernel::services().start();
```

(continues on next page)

(continued from previous page)

```
auto ctx = std::make_unique<nw::script::Context>();
nw::script::Nss nss{nw::kernel::resman().demand({"nwscript"sv, nw::ResourceType::nss),␣
↪ctx.get(), true};

// Parse
nss.parse();

// Preprocessing
nss.process_includes()
// Now all dependencies are available
auto deps = nss.dependencies()

// Ast resolution and type-checking
nss.resolve()
```

Iterating Top-Level Declarations

```
import rollnw
from rollnw.script import Nss, FunctionDecl, Context

# Start kernel, if you want to load game assets
rollnw.kernel.start()

# Create a context..
ctx = Context()

# The default command script is "nwsscript"
nss = ctx.command_script()

# Iterate toplevel declarations and look for function declarations
# This is all functions WITHOUT bodies.
for decl in nss.ast():
    if isinstance(decl, FunctionDecl):
        # the identifier is token for now..
        print(f"function '{decl.identifier()}' has {len(decl)} parameter(s)")

# Or if you know what you're looking for.. the result is a rollnw.script.Symbol
int_to_string = nss.locate_export("IntToString", False)
```

6.11.2 performance

The parser currently parses at >100MBps on a 2015 MacBook Pro.

6.11.3 TODOs

- Decide how much to track NWN:EE NWScript changes, only raw strings isn't already done.
- Make the library more useful for NWScript successors (i.e Dragon Age or KoTOR)
- Whether to do optimizations or anything further than performance/usability improvements

6.11.4 credits

- Crafting Interpreters

6.12 serialization

6.12.1 Definitions

- **profile** - NWN has three different (de)serialization profiles:
  - **blueprint** - UTC, UTT, etc, etc. BIC is included here, though not a blueprint itself.
  - **instance** - Instances of blueprints stored in an area's GIT file.
  - **savegame** - All game and object state. This is outside of the scope of this library.. for now.
- **type** - C++ types corresponding to GFF serialization types.
  - `uint8_t` - Also convertible to `bool`
  - `int8_t`
  - `uint16_t`
  - `int16_t`
  - `uint32_t`
  - `int32_t`
  - `uint64_t`
  - `int64_t`
  - `float`
  - `double`
  - `std::string`
  - `Resref`
  - `LocString`
  - `ByteArray`
  - Scoped Enumerations are convertible when their underlying type matches the GFF type.

The library may support the lifting of numeric types, i.e. reading a `int16_t` into `int16_t` or `int32_t` or `int64_t`.

- **struct** is a collection of key-value pairs, where the key is a 16 character string and the value is one of the above types (almost).
- **list** is a list solely of structs, this follows the GFF pattern.
- **gffjson** refers to the nwn-lib/neverwinter.nim json format that mimics GFF. The extent to which this is supported by the library is an open issue.
- **json** refers specifically to rollnw json serialization. This very closely mimics the structure of a given object, such that if you load the JSON into another language, or a dynamic language that can construct arbitrary objects from JSON, the usage is identical or analogous to the C++ objects.

6.12.2 Examples

Example - How to build your own GFF

```
nw::GffBuilder gff{"GFF"};

// Add a field.  Note that the type of the field is determined by the value
// passed.
gff.top.add_field("DATA", 9);

// Add a list.  Note that in the GFF format lists contain only structs
auto& xs = gff.top.add_list("LIST");
// So when you push_back, you're creating a struct with a specific struct ID
auto& st = xs.push_back(0xBEEF);
// Now you can add fields to the struct
st.add_field("A", 1)
    .add_field("B", 12);

// Add a struct.  It's pretty rare that a gff field is a struct but if necessary
// you can add a struct with its struct ID, then add fields like above.
gff.top.add_struct("STRUCT", 42)
    .add_field("A", 1)
    .add_field("B", 12);

gff.build(); // This must be called after all fields have been added.
gff.write_to("mygff.gff");
```

6.12.3 Sample rollnw JSON serialization format

```
{
    "$type": "UTC",
    "$version": 1,
    "appearance": {
        "body_parts": {
            "belt": 0,
            "bicep_left": 1,
            "bicep_right": 1,
            "foot_left": 1,
            "foot_right": 1,
```

(continues on next page)

(continued from previous page)


```
            "forearm_left": 1,
            "forearm_right": 1,
            "hand_left": 1,
            "hand_right": 1,
            "head": 119,
            "neck": 1,
            "pelvis": 1,
            "shin_left": 1,
            "shin_right": 1,
            "shoulder_left": 0,
            "shoulder_right": 0,
            "thigh_left": 1,
            "thigh_right": 1,
            "torso": 1
        },
        "hair": 167,
        "id": 6,
        "phenotype": 0,
        "portrait_id": 65,
        "skin": 3,
        "tail": 0,
        "tattoo1": 1,
        "tattoo2": 1,
        "wings": 0
    },
    "bodybag": 0,
    "chunk_death": 0,
    "combat_info": {
        "ac_natural": 0,
        "special_abilities": [
            {
                "flags": 1,
                "level": 15,
                "spell": 120
            }
        ]
    },
    "common": {
        "comment": "",
        "locals": {
            "DIPType": {
                "integer": 3
            },
            "DeflectionAC": {
                "integer": 6
            },
            "DodgeAC": {
                "integer": 6
            },
            "OtherImmunes": {
                "integer": 1001945111
            },
```


(continues on next page)

(continued from previous page)


```
            "Soak": {
                "string": "15+5"
            },
            "VFXDur1": {
                "integer": 11
            },
            "rlgs_ss_1": {
                "string": "lt_agent_1"
            }
        },
        "object_type": 5,
        "palette_id": 0,
        "resref": "pl_agent_001",
        "tag": "pl_agent_001"
    },
    "conversation": "",
    "cr": 38.0,
    "cr_adjust": -36,
    "decay_time": 5000,
    "deity": "",
    "description": {
        "strings": [],
        "strref": 4294967295
    },
    "disarmable": 0,
    "equipment": {
        "arms": "handwish",
        "chest": "dk_agent_thread2",
        "creature_left": "pl_slam_1d2"
    },
    "faction_id": 1,
    "gender": 0,
    "good_evil": 100,
    "hp": 894,
    "hp_current": 894,
    "hp_max": 1014,
    "immortal": 0,
    "interruptable": 0,
    "inventory": [],
    "lawful_chaotic": 50,
    "levels": [
        {
            "class": 4,
            "level": 10,
            "spellbook": {
                "known": [
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
```


(continues on next page)

(continued from previous page)


```
                    [],
                    [],
                    [],
                    []
                ],
                "memorized": [
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    []
                ]
            }
        },
        {
            "class": 5,
            "level": 30,
            "spellbook": {
                "known": [
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    []
                ],
                "memorized": [
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    [],
                    []
                ]
            }
        }
    ],
    "lootable": 0,
    "name_first": {
```


(continues on next page)

(continued from previous page)


```
        "strings": [
            {
                "lang": 0,
                "string": "Agent"
            }
        ],
        "strref": 4294967295
    },
    "name_last": {
        "strings": [],
        "strref": 4294967295
    },
    "pc": 0,
    "perception_range": 11,
    "plot": false,
    "race": 6,
    "scripts": {
        "on_attacked": "mon_ai_5attacked",
        "on_blocked": "mon_ai_13blocked",
        "on_conversation": "mon_ai_4conv",
        "on_damaged": "mon_ai_6dmged",
        "on_death": "mon_ai_7death",
        "on_disturbed": "mon_ai_8disturb",
        "on_endround": "mon_ai_3ocre",
        "on_heartbeat": "mon_ai_1hb",
        "on_perceived": "mon_ai_2percep",
        "on_rested": "mon_ai_10rest",
        "on_spawn": "mon_ai_9spawn",
        "on_spell_cast_at": "mon_ai_11spcast",
        "on_user_defined": "mon_ai_12ud"
    },
    "soundset": 171,
    "starting_package": 4,
    "stats": {
        "abilities": [
            40,
            13,
            16,
            10,
            16,
            9
        ],
        "feats": [
            2,
            3,
            4,
            6,
            8,
            10,
            21,
            26,
            32,
```


(continues on next page)

(continued from previous page)


```
            41,
            45,
            46,
            49,
            206,
            207,
            208,
            209,
            211,
            212,
            214,
            215,
            216,
            258,
            260,
            289,
            290,
            291,
            292,
            297,
            391,
            392,
            408,
            755,
            756,
            757,
            971,
            1089
        ],
        "save_bonus": {
            "fort": 9,
            "reflex": 15,
            "will": 13
        },
        "skills": [
            0,
            1,
            0,
            40,
            11,
            30,
            30,
            1,
            30,
            0,
            20,
            0,
            30,
            0,
            0,
            0,
            0,
```


(continues on next page)

(continued from previous page)

```
            0,
            0,
            0,
            0,
            0,
            1,
            0,
            0,
            1,
            2,
            0
        ]
    },
    "subrace": "",
    "walkrate": 4
}
```

6.13 classes

6.13.1 nw::Ability

struct **Ability**

Public Functions

bool **operator==**(const *Ability* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Ability* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Ability* **make**(int32_t id)

Makes a rule type

static inline constexpr *Ability* **invalid**()

Returns an invalid rule type

6.13.2 nw::AbilityInfo

struct **AbilityInfo**

Public Functions

inline bool **valid**() const noexcept

Public Members

uint32_t **name** = 0xFFFFFFFF

InternedString **constant**

6.13.3 nw::Appearance

struct **Appearance**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

int32_t **phenotype** = 0

uint32_t **tail** = 0

uint32_t **wings** = 0

uint16_t **id** = 0

uint16_t **portrait_id**

BodyParts **body_parts**

uint8_t **hair** = 0

uint8_t **skin** = 0

uint8_t **tattoo1** = 0

uint8_t **tattoo2** = 0

6.13.4 nw::Area

struct **Area** : public nw::ObjectBase

Public Functions

Area()

inline virtual *Common* ***as_common**() override

inline virtual const *Common* ***as_common**() const override

inline virtual *Area* ***as_area**() override

inline virtual const *Area* ***as_area**() const override

virtual bool **instantiate**() override

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Versus* **versus_me**() const

virtual *InternedString* **tag**() const

inline virtual *Creature* ***as_creature**()

inline virtual const *Creature* ***as_creature**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Encounter* ***as_encounter**()

inline virtual const *Encounter* ***as_encounter**() const

inline virtual *Item* ***as_item**()

inline virtual const *Item* ***as_item**() const

inline virtual *Module* ***as_module**()

inline virtual const *Module* ***as_module**() const

inline virtual *Placeable* ***as_placeable**()

inline virtual const *Placeable* ***as_placeable()** const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

AreaScripts **scripts**

AreaWeather **weather**

std::vector<*Creature**> **creatures**

std::vector<*Door**> **doors**

std::vector<*Encounter**> **encounters**

std::vector<*Item**> **items**

std::vector<*Placeable**> **placeables**

std::vector<*Sound**> **sounds**

std::vector<*Store**> **stores**

std::vector<*Trigger**> **triggers**

std::vector<*Waypoint**> **waypoints**

std::string **comments**

LocString **name**

Resref **tileset**

std::vector<*Tile*> **tiles**

int32_t **creator_id** = 0

AreaFlags **flags**

int32_t **height** = 0

int32_t **id** = 0

int32_t **listen_check_mod** = 0

int32_t **spot_check_mod** = 0

uint32_t **version** = 0

int32_t **width** = 0

uint16_t **loadscreen** = 0

uint8_t **no_rest** = 0

uint8_t **pvp** = 0

uint8_t **shadow_opacity** = 0

uint8_t **skybox** = 0

Public Static Functions

static bool **deserialize**(*Area* *obj, const nlohmann::json &caf)

Deserialize from JSON

> **Note:** Note only supports does 'caf' style input/output, i.e. ARE + GIT + GIC.

static bool **deserialize**(*Area* *obj, const nlohmann::json &are, const nlohmann::json &git, const nlohmann::json &gic)

static void **serialize**(const *Area* *obj, nlohmann::json &archive)

Serialize to JSON.

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*area*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::caf

6.13.5 nw::AreaScripts

struct **AreaScripts**

Public Functions

AreaScripts() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

Resref **on_enter**

Resref **on_exit**

Resref **on_heartbeat**

Resref **on_user_defined**

6.13.6 nw::AreaWeather

struct **AreaWeather**

Public Functions

AreaWeather() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

int32_t **chance_lightning** = 0

int32_t **chance_rain** = 0

int32_t **chance_snow** = 0

uint32_t **color_moon_ambient** = 0

uint32_t **color_moon_diffuse** = 0

uint32_t **color_moon_fog** = 0

uint32_t **color_sun_ambient** = 0

uint32_t **color_sun_diffuse** = 0

uint32_t **color_sun_fog** = 0

float **fog_clip_distance** = 0.0f

int32_t **wind_power** = 0

uint8_t **day_night_cycle** = 0

uint8_t **is_night** = 0

uint8_t **lighting_scheme** = 0

uint8_t **fog_moon_amount** = 0

uint8_t **moon_shadows** = 0

uint8_t **fog_sun_amount** = 0

uint8_t **sun_shadows** = 0

6.13.7 nw::AttackData

struct **AttackData**

Structure for aggregating attack related data.

Public Types

using **DamageArray** = absl::InlinedVector<*DamageResult*, 8>

Public Functions

void **add**(nw::Damage type_, int amount, bool unblockable = false)

Adds damage to damage result.

DamageArray &**damages**()

Gets damage array.

const *DamageArray* &**damages**() const

Gets damage array.

Public Members

Creature ***attacker** = nullptr

ObjectBase ***target** = nullptr

Item ***weapon** = nullptr

AttackType **type** = AttackType::invalid()

AttackResult **result** = *AttackResult*::*miss_by_roll*

TargetState **target_state** = *TargetState*::*none*

bool **target_is_creature** = false

bool **is_ranged_attack** = false

bool **is_killing_blow** = false

Is the attack enough to kill target.

int **nth_attack** = 0

The nth attack in the 'round'.

int **attack_roll** = 0

int **attack_bonus** = 0

int **damage_total** = 0

int **armor_class** = 0

int **iteration_penalty** = 0

int **multiplier** = 0

int **threat_range** = 0

int **concealment** = 0

DamageResult **damage_base**

Base weapon damage.

absl::InlinedVector<nw::*Effect**, 8> **effects_to_apply**

Effects to apply to target.

absl::InlinedVector<nw::*EffectHandle*, 8> **effects_to_remove**

Effects to remove from target.

6.13.8 nw::BaseItem

struct **BaseItem**

Public Functions

bool **operator==**(const *BaseItem* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *BaseItem* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *BaseItem* **make**(int32_t id)

Makes a rule type

static inline constexpr *BaseItem* **invalid**()

Returns an invalid rule type

6.13.9 nw::BeamdogInstall

struct **BeamdogInstall**

Public Members

const char ***appid**

const char ***path**

6.13.10 nw::Bif

struct **Bif**

Bif is used only by *nw::Key*, it has no independant use.

Public Functions

Bif(*Key* *key, std::filesystem::path path)

Bif(const *Bif* &) = delete

Bif(*Bif* &&other) = default

Bif &**operator=**(const *Bif* &) = delete

Bif &**operator=**(*Bif* &&other) = default

ByteArray **demand**(size_t index) const

Friends

friend struct Key

6.13.11 nw::BodyParts

struct **BodyParts**

Public Members

uint8_t **belt** = 0

uint8_t **bicep_left** = 0

uint8_t **bicep_right** = 0

uint8_t **foot_left** = 0

uint8_t **foot_right** = 0

uint8_t **forearm_left** = 0

uint8_t **forearm_right** = 0

uint8_t **hand_left** = 0

uint8_t **hand_right** = 0

uint8_t **head** = 0

uint8_t **neck** = 0

uint8_t **pelvis** = 0

uint8_t **shin_left** = 0

uint8_t **shin_right** = 0

uint8_t **shoulder_left** = 0

uint8_t **shoulder_right** = 0

uint8_t **thigh_left** = 0

uint8_t **thigh_right** = 0

uint8_t **torso** = 0

6.13.12 nw::ByteArray

struct **ByteArray**

Public Types

using **Base** = std::vector<uint8_t>

using **iterator** = *Base*::iterator

using **const_iterator** = *Base*::const_iterator

using **size_type** = *Base*::size_type

Public Functions

ByteArray() = default

ByteArray(const uint8_t *buffer, size_t len)

ByteArray(*ByteArray*&&) = default

ByteArray(const *ByteArray*&) = default

ByteArray &**operator=**(*ByteArray*&&) = default

ByteArray &**operator=**(const *ByteArray*&) = default

inline bool **operator==**(const *ByteArray* &other) const

inline uint8_t &**operator[]**(*size_type* pos)

inline const uint8_t &**operator[]**(*size_type* pos) const

void **append**(const void *buffer, size_t len)

Appends bytes to the array.

inline void **clear**()

Clears the data in the array.

inline uint8_t ***data**() noexcept

Returns pointer to the underlying array.

inline const uint8_t ***data**() const noexcept

Returns pointer to the underlying array.

inline void **push_back**(uint8_t byte)

Appends one element to the array.

bool **read_at**(size_t offset, void *buffer, size_t size) const

Reads `size` bytes at `offset` into an arbitrary `buffer`

inline void **reserve**(*size_type* count)

Increases the capacity of the array by `count` elements.

inline void **resize**(*size_type* count)

Resizes array to contain `count` elements. If greater, than current size, null padded.

inline *size_type* **size**() const noexcept

Returns the number of bytes.

inline std::span<uint8_t> **span**()

Construct std::span.

inline std::span<const uint8_t> **span**() const

Construct std::span.

std::string_view **string_view**() const

Constructs string view of the array.

bool **write_to**(const std::filesystem::path &path) const

Write contents to file.

Public Static Functions

static *ByteArray* **from_file**(const std::filesystem::path &path)

Load a file into memory.

6.13.13 nw::Class

struct **Class**

Public Functions

bool **operator==**(const *Class* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Class* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Class* **make**(int32_t id)

Makes a rule type

static inline constexpr *Class* **invalid**()

Returns an invalid rule type

6.13.14 nw::ClassArray

struct **ClassArray**

Class Singleton component.

Public Types

using **map_type** = absl::flat_hash_map<*InternedString*, *Class*, InternedStringHash, InternedStringEq>

Public Functions

const *ClassInfo* ***get**(*Class* class_) const noexcept

bool **is_valid**(*Class* class_) const noexcept

Class **from_constant**(std::string_view constant) const

int **get_base_attack_bonus**(*Class* class_, size_t level) const

Gets class base attack from attack tables.

Saves **get_class_save_bonus**(*Class* class_, size_t level) const

Gets class save bonuses from save tables.

bool **get_is_class_skill**(*Class* class_, *Skill* skill) const

Determines if skill is a class skill.

int **get_natural_ac**(*Class* class_, size_t level) const

Gets class Natural AC gain.

const ClassRequirement *`get_requirement`(*Class* class_) const

Gets class requirements.

int `get_stat_gain`(*Class* class_, *Ability* ability, size_t level) const

Gets class ability gain.

Public Members

std::set<std::vector<int>> `attack_tables`

std::vector<int> `stat_gain_tables`

std::vector<*ClassInfo*> `entries`

map_type `constant_to_index`

6.13.15 nw::ClassEntry

struct `ClassEntry`

Public Members

Class `id` = nw::*Class*::*invalid*()

int16_t `level` = 0

SpellBook `spells`

6.13.16 nw::ClassInfo

struct `ClassInfo`

Public Functions

`ClassInfo`() = default

`ClassInfo`(const TwoDARowView &tda)

inline bool `valid`() const noexcept

Public Members

ClassRequirement **requirements**

uint32_t **name** = 0xFFFFFFFF

uint32_t **plural** = 0xFFFFFFFF

uint32_t **lower** = 0xFFFFFFFF

uint32_t **description** = 0xFFFFFFFF

Resource **icon**

int **hitdie** = 0

const std::vector<int> ***attack_bonus_table** = nullptr

Resource **feats_table**

Resource **saving_throw_table**

std::vector<*Saves*> **class_saves**

Resource **skill_table**

std::vector<int> **class_skills**

Resource **bonus_feats_table**

int **skill_point_base** = 0

Resource **spell_gain_table**

Resource **spell_known_table**

bool **player_class** = false

bool **spellcaster** = false

int **primary_ability**

uint32_t **alignment_restriction** = 0

uint32_t **alignment_restriction_type** = 0

bool **invert_restriction** = false

InternedString **constant**

Resource **prereq_table**

int **max_level** = 0

int **xp_penalty** = 0

int **arcane_spellgain_mod** = 0

int **divine_spellgain_mod** = 0

int **epic_level_limit** = -1

int **package** = 0

Resource **stat_gain_table**

std::vector<ClassStatGain> **class_stat_gain**

bool **memorizes_spells** = false

bool **spellbook_restricted** = false

bool **pick_domains** = false

bool **pick_school** = false

bool **learn_scroll** = false

bool **arcane** = false

bool **arcane_spell_failure** = false

nw::*Ability* **caster_ability** = nw::*Ability*::*invalid*()

std::string **spell_table_column**

float **caster_level_multiplier** = 1.0f

int **level_min_caster** = 0

int **level_min_associate** = 0

bool **can_cast_spontaneously** = false

6.13.17 nw::CombatInfo

struct **CombatInfo**

Public Functions

CombatInfo() = default

CombatInfo(*CombatInfo*&) = default

CombatInfo(*CombatInfo*&&) = default

CombatInfo &**operator=**(*CombatInfo*&) = delete

CombatInfo &**operator=**(*CombatInfo*&&) = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

int **ac_natural_bonus** = 0

int **attack_current** = 0

Current attack counter.

Note: Invariant: attack_current <= attacks_onhand + attacks_offhand + attacks_extra

int **attacks_onhand** = 0

int **attacks_offhand** = 0

int **attacks_extra** = 0

nw::ObjectBase ***target** = nullptr

float **target_distance_sq** = 0.0f

Distance to target squared.

TargetState **target_state** = *TargetState*::*none*

int **ac_armor_base** = 0

int **ac_shield_base** = 0

CombatMode **combat_mode** = nw::CombatMode::invalid()

int32_t **size_ab_modifier** = 0

int32_t **size_ac_modifier** = 0

std::vector<*SpecialAbility*> **special_abilities**

6.13.18 nw::Common

struct **Common**

Public Functions

bool **from_json**(const nlohmann::json &archive, *SerializationProfile* profile, *ObjectType* object_type)

nlohmann::json **to_json**(*SerializationProfile* profile, *ObjectType* object_type) const

inline bool **valid**()

Public Members

uuids::uuid **uuid**

Resref **resref**

InternedString **tag**

LocString **name**

LocalData **locals**

Location **location**

std::string **comment**

uint8_t **palette_id** = std::numeric_limits<uint8_t>::max()

6.13.19 nw::CompressionHeader

struct **CompressionHeader**

Public Members

std::array<char, 4> **magic**

uint32_t **version**

uint32_t **algorithm**

uint32_t **uncompressed_size**

6.13.20 nw::ConfigOptions

struct **ConfigOptions**

Configuration options, maybe there will be an actual config file.. someday.

Public Members

bool **include_install** = true

Load Game install files.

bool **include_nwsync** = true

Load *NWSync* files.

bool **include_user** = true

Load User files, note: if false, value overrides `include_nwsync`

6.13.21 nw::Container

struct **Container**

Base class for all containers.

Subclassed by *nw::Directory*, *nw::Erf*, *nw::Key*, *nw::NWSyncManifest*, *nw::Zip*, *nw::kernel::Resources*

Public Functions

Container()

virtual **~Container**()

virtual std::vector<*ResourceDescriptor*> **all**() const = 0

Get all resources.

virtual bool **contains**(*Resource* res) const = 0

Get if container contains resource.

virtual ResourceData **demand**(*Resource* res) const = 0

Reads resource data, empty ResourceData if no match.

virtual int **extract_by_glob**(std::string_view glob, const std::filesystem::path &output) const

Extract elements from a container by glob pattern.

virtual int **extract**(const std::regex &pattern, const std::filesystem::path &output) const = 0

Extract elements from a container by regex.

virtual const std::string &**name**() const = 0

Equivalent to basename *path()*

virtual const std::string &**path**() const = 0

Path to container, for basic containers, should be canonical.

virtual size_t **size**() const = 0

Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **stat**(const *Resource* &res) const = 0

Get some general data about a resource.

virtual bool **valid**() const noexcept = 0

Return true if loaded, false if not.

virtual void **visit**(std::function<void(const *Resource*&)> callback) const noexcept = 0

Visits all resources in a container.

const std::filesystem::path &**working_directory**() const

Get container working directory.

6.13.22 nw::Creature

struct **Creature** : public nw::ObjectBase

Subclassed by *nw::Player*

Public Functions

Creature()

inline virtual *Common* ***as_common**() override

inline virtual const *Common* ***as_common**() const override

inline virtual *Creature* ***as_creature**() override

inline virtual const *Creature* ***as_creature**() const override

virtual bool **instantiate**() override

inline virtual *InternedString* **tag**() const override

virtual *Versus* **versus_me**() const override

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Area* ***as_area**()

inline virtual const *Area* ***as_area**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Encounter* ***as_encounter**()

inline virtual const *Encounter* ***as_encounter**() const

inline virtual *Item* ***as_item**()

inline virtual const *Item* ***as_item**() const

inline virtual *Module* ***as_module**()

inline virtual const *Module* ***as_module**() const

inline virtual *Placeable* ***as_placeable**()

inline virtual const *Placeable* ***as_placeable**() const

inline virtual *Player* ***as_player**()

inline virtual const *Player* ***as_player**() const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

Appearance **appearance**

CombatInfo **combat_info**

Equips **equipment**

Inventory **inventory**

LevelStats **levels**

LevelHistory **history**

CreatureScripts **scripts**

CreatureStats **stats**

Resref **conversation**

std::string **deity**

LocString **description**

LocString **name_first**

LocString **name_last**

std::string **subrace**

float **cr** = 0.0

int32_t **cr_adjust** = 0

uint32_t **decay_time**

Race **race** = *Race*::*invalid*()

int32_t **walkrate** = 0

uint16_t **faction_id** = 0

int16_t **hp** = 0

int16_t **hp_current** = 0

int16_t **hp_max** = 0

int16_t **hp_temp** = 0

uint16_t **soundset**

int32_t **hasted** = 0

int32_t **size** = 0

uint8_t **bodybag** = 0

uint8_t **chunk_death** = 0

uint8_t **disarmable** = 0

uint8_t **gender** = 0

uint8_t **good_evil** = 50

uint8_t **interruptable** = 0

uint8_t **immortal** = 0

uint8_t **lawful_chaotic** = 50

uint8_t **lootable** = 0

uint8_t **pc** = 0

uint8_t **perception_range** = 0

bool **plot** = false

uint8_t **starting_package** = 0

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Creature* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Creature* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*creature*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::utc

6.13.23 nw::CreatureScripts

struct **CreatureScripts**

Public Functions

CreatureScripts() = default

bool **deserialize**(const *GffStruct* &archive)

bool **from_json**(const nlohmann::json &archive)

bool **serialize**(*GffBuilderStruct* &archive) const

nlohmann::json **to_json**() const

Public Members

Resref **on_attacked**

Resref **on_blocked**

Resref **on_conversation**

Resref **on_damaged**

Resref **on_death**

Resref **on_disturbed**

Resref **on_endround**

Resref **on_heartbeat**

Resref **on_perceived**

Resref **on_rested**

Resref **on_spawn**

Resref **on_spell_cast_at**

Resref **on_user_defined**

6.13.24 nw::CreatureStats

struct **CreatureStats**

Public Functions

CreatureStats() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

bool **add_feat**(*Feat* id)

Attempts to add a feat to a creature, returning true if successful.

const std::vector<*Feat*> &**feats**() const noexcept

Gets the feat array.

int **get_ability_score**(*Ability* id) const

Gets an ability score.

int **get_skill_rank**(*Skill* id) const

Gets a skill rank.

bool **has_feat**(*Feat* id) const noexcept

Determines if creature has a feat.

bool **set_ability_score**(*Ability* id, int value)

Sets an ability score, returning true if successful.

bool **set_skill_rank**(*Skill* id, int value)

Sets a skill rank, returning true if successful.

Public Members

Saves **save_bonus**

Friends

friend bool **deserialize**(*CreatureStats* &self, const *GffStruct* &archive)

friend bool **serialize**(const *CreatureStats* &self, *GffBuilderStruct* &archive)

6.13.25 nw::DamageResult

struct **DamageResult**

Public Members

nw::Damage **type** = nw::Damage::invalid()

int **amount** = 0

int **unblocked** = 0

int **immunity** = 0

int **reduction** = 0

int **reduction_remaining** = 0

int **resist** = 0

int **resist_remaining** = 0

6.13.26 nw::DamageRoll

struct **DamageRoll**

Public Members

Damage **type** = Damage::invalid()

DiceRoll **roll**

DamageCategory **flags** = DamageCategory::none

6.13.27 nw::Dialog

struct **Dialog**

Public Functions

Dialog()

explicit **Dialog**(const *GffStruct* archive)

explicit **Dialog**(const nlohmann::json &archive)

Dialog(const *Dialog*&) = delete

Dialog &**operator=**(const *Dialog*&) = delete

DialogPtr ***add**()

Adds empty *Dialog* Pointer and Node.

void **add_node_internal**(*DialogNode* *node, *DialogNodeType* type)

Adds a node to the iternal node lists

> **Warning:** This should be considered for internal use and not client code

DialogPtr ***add_ptr**(*DialogPtr* *ptr, bool is_link = false)

Adds *Dialog* Pointer, if `is_link` is false no new pointer or node is created. if `is_link` is true a new pointer will created with the node copied from input pointer.

DialogPtr ***add_string**(std::string value, nw::*LanguageID* lang = nw::*LanguageID*::*english*, bool feminine = false)

Adds *Dialog* Pointer and Node with string value set.

DialogNode ***create_node**(*DialogNodeType* type)

Creates a new *Dialog* Node.

DialogPtr ***create_ptr**()

Creates a new *Dialog* Pointer.

void **delete_node**(*DialogNode* *node)

Deletes a dialog node

> **Warning:** This should be considered for internal use and not client code

void **delete_ptr**(*DialogPtr* *ptr)

Deletes a dialog pointer

> **Warning:** `ptr` should be removed from / not added to a dialog prior to deletion

size_t **node_index**(*DialogNode* *node, *DialogNodeType* type) const

Get Node index.

void **remove_node_internal**(*DialogNode* *node, *DialogNodeType* type)

Removes a node to the iternal node lists

> **Warning:** This should be considered for internal use and not client code

void **remove_ptr**(*DialogPtr* *ptr)

Removes *Dialog* Ptr from underlying node.

inline bool **valid**() const noexcept

Checks id dialog was successfully parsed.

Public Members

std::vector<*DialogNode**> **entries**

std::vector<*DialogNode**> **replies**

Resref **script_abort**

Resref **script_end**

std::vector<*DialogPtr**> **starts**

uint32_t **delay_entry** = 0

uint32_t **delay_reply** = 0

uint32_t **word_count** = 0

bool **prevent_zoom** = false

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::dlg

6.13.28 nw::DialogNode

struct **DialogNode**

Public Functions

inline **DialogNode**()

DialogNode ***copy**() const

Copies a Node.

std::optional<std::string> **get_action_param**(const std::string &key)

Gets action parameter if it exists.

void **remove_action_param**(const std::string &key)

Removes action parameter by key.

void **remove_action_param**(size_t index)

Removes action parameter by index.

void **set_action_param**(const std::string &key, const std::string &value)

Sets action parameter, if key does not exist key and value are appended.

Public Members

Dialog ***parent** = nullptr

DialogNodeType **type**

std::string **comment**

std::string **quest**

std::string **speaker**

uint32_t **quest_entry** = std::numeric_limits<uint32_t>::max()

Resref **script_action**

Resref **sound**

LocString **text**

DialogAnimation **animation** = DialogAnimation::default_

bool **animation_loop** = false

uint32_t **delay** = std::numeric_limits<uint32_t>::max()

std::vector<*DialogPtr**> **pointers**

std::vector<std::pair<std::string, std::string>> **action_params**

6.13.29 nw::DialogPtr

struct **DialogPtr**

Public Functions

DialogPtr ***add_ptr**(*DialogPtr* *ptr, bool is_link = false)

Adds *Dialog* Pointer, if `is_link` is false no new pointer or node is created. if `is_link` is true a new pointer will created with the node copied from input pointer.

DialogPtr ***add_string**(std::string value, nw::*LanguageID* lang = nw::*LanguageID*::*english*, bool feminine = false)

Adds *Dialog* Pointer and Node with string value set.

DialogPtr ***add**()

Adds empty *Dialog* Pointer and Node.

DialogPtr ***copy**() const

Copies dialog pointer and all sub-nodes.

void **get_all_subnodes**(std::vector<*DialogNode**> &subnodes)

Gets all sub-nodes that are not links When a pointer is removed from the dialog tree all of its sub-nodes must be removed from the main node list, unless they are links.

std::optional<std::string> **get_condition_param**(const std::string &key)

Gets a condition parameter if it exists.

void **remove_condition_param**(const std::string &key)

Removes condition parameter by key.

void **remove_condition_param**(size_t index)

Removes condition parameter by index.

void **remove_ptr**(*DialogPtr* *ptr)

Removes *Dialog* Ptr from underlying node.

void **set_condition_param**(const std::string &key, const std::string &value)

Sets condition parameter, if key does not exist key and value are appended.

Public Members

Dialog ***parent** = nullptr

DialogNodeType **type** = *DialogNodeType*::*entry*

uint32_t **index** = std::numeric_limits<uint32_t>::max()

DialogNode ***node** = nullptr

Resref **script_appears**

std::vector<std::pair<std::string, std::string>> **condition_params**

bool **is_start** = false

bool **is_link** = false

std::string **comment**

6.13.30 nw::DiceRoll

struct **DiceRoll**

A dice roll.

Public Functions

inline **operator bool**()

Public Members

int **dice** = 0

Number of dice to roll.

int **sides** = 0

Number of sides on the dice.

int **bonus** = 0

Additional bonus.

6.13.31 nw::Directory

struct **Directory** : public nw::*Container*

Public Functions

Directory() = default

explicit **Directory**(const std::filesystem::path &path)

virtual **~Directory**() = default

virtual std::vector<*ResourceDescriptor*> **all**() const override

Get all resources.

virtual bool **contains**(*Resource* res) const override

Get if container contains resource.

virtual ResourceData **demand**(*Resource* res) const override

Reads resource data, empty ResourceData if no match.

virtual int **extract**(const std::regex &pattern, const std::filesystem::path &output) const override

Extract elements from a container by regex.

inline virtual const std::string &**name**() const override

Equivalent to `basename` *`path()`*

inline virtual const std::string &**path**() const override

Path to container, for basic containers, should be canonical.

virtual size_t **size**() const override

Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **stat**(const *Resource* &res) const override

Get some general data about a resource.

inline virtual bool **valid()** const noexcept override

Return true if loaded, false if not.

virtual void **visit**(std::function<void(const *Resource*&)> callback) const noexcept override

Visits all resources in a container.

virtual int **extract_by_glob**(std::string_view glob, const std::filesystem::path &output) const

Extract elements from a container by glob pattern.

const std::filesystem::path &**working_directory()** const

Get container working directory.

6.13.32 nw::Disease

struct **Disease**

Public Functions

bool **operator==**(const *Disease* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Disease* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator*()** const noexcept

Returns rule type as value

inline constexpr size_t **idx()** const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Disease* **make**(int32_t id)

Makes a rule type

static inline constexpr *Disease* **invalid()**

Returns an invalid rule type

6.13.33 nw::Door

struct **Door** : public nw::ObjectBase

Public Functions

Door()

inline virtual *Common* ***as_common()** override

inline virtual const *Common* ***as_common()** const override

inline virtual *Door* ***as_door()** override

inline virtual const *Door* ***as_door()** const override

inline virtual bool **instantiate()** override

inline virtual *InternedString* **tag()** const override

inline ObjectHandle **handle()** const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects()** const

EffectArray &**effects()**

inline virtual *Versus* **versus_me()** const

inline virtual *Area* ***as_area()**

inline virtual const *Area* ***as_area()** const

inline virtual *Creature* ***as_creature()**

inline virtual const *Creature* ***as_creature()** const

inline virtual *Encounter* ***as_encounter()**

inline virtual const *Encounter* ***as_encounter()** const

inline virtual *Item* ***as_item()**

inline virtual const *Item* ***as_item()** const

inline virtual *Module* ***as_module()**

inline virtual const *Module* ***as_module()** const

inline virtual *Placeable* ***as_placeable()**

inline virtual const *Placeable* ***as_placeable()** const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

DoorScripts **scripts**

Lock **lock**

Trap **trap**

Resref **conversation**

LocString **description**

std::string **linked_to**

Saves **saves**

uint32_t **appearance**

uint32_t **faction** = 0

uint32_t **generic_type** = 0

int16_t **hp** = 0

int16_t **hp_current** = 0

uint16_t **loadscreen** = 0

uint16_t **portrait_id**

DoorAnimationState **animation_state** = *DoorAnimationState*::*closed*

uint8_t **hardness**

bool **interruptable** = 0

uint8_t **linked_to_flags** = 0

bool **plot** = false

bool **instantiated_** = true

Public Static Functions

static bool **deserialize**(*Door* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Door* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*door*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::utd

6.13.34 nw::DoorScripts

struct **DoorScripts**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

Resref **on_click**

Resref **on_closed**

Resref **on_damaged**

Resref **on_death**

Resref **on_disarm**

Resref **on_heartbeat**

Resref **on_lock**

Resref **on_melee_attacked**

Resref **on_open**

Resref **on_open_failure**

Resref **on_spell_cast_at**

Resref **on_trap_triggered**

Resref **on_unlock**

Resref **on_user_defined**

6.13.35 nw::Effect

struct **Effect**

Public Functions

Effect()

Effect(EffectType type_)

void **clear**()

Clears the effect such that it's as if default constructed.

float **get_float**(size_t index) const noexcept

Gets a floating point value.

int **get_int**(size_t index) const noexcept

Gets an integer point value.

std::string_view **get_string**(size_t index) const noexcept

Gets a string value.

EffectHandle **handle**() noexcept

Gets the effect's handle.

EffectID **id**() const noexcept

Gets the effect's ID.

void **set_float**(size_t index, float value)

Sets a floating point value.

void **set_id**(EffectID id)

Sets effect's ID.

void **set_int**(size_t index, int value)

Sets an integer point value.

void **set_string**(size_t index, std::string value)

Sets a string value.

void **set_versus**(*Versus* vs)

Sets the versus value.

const *Versus* &**versus**() const noexcept

Gets the versus value.

Public Members

EffectType **type** = EffectType::invalid()

EffectCategory **category** = EffectCategory::magical

int **subtype** = -1

ObjectHandle **creator**

Spell **spell_id** = *Spell*::*invalid*()

float **duration** = 0.0f

uint32_t **expire_day** = 0

uint32_t **expire_time** = 0

6.13.36 nw::EffectArray

struct **EffectArray**

Public Types

using **storage** = std::vector<*EffectHandle*>

using **iterator** = *storage*::iterator

using **const_iterator** = *storage*::const_iterator

Public Functions

bool **add**(*Effect* *effect)

Adds an effect.

iterator **begin**()

const_iterator **begin**() const

iterator **end**()

const_iterator **end**() const

void **erase**(*iterator* first, *iterator* last)

Removes a range of effects.

bool **remove**(*Effect* *effect)

Removes an effect.

size_t **size**() const noexcept

Gets the number of applied effects.

6.13.37 nw::EffectHandle

struct **EffectHandle**

Public Functions

bool **operator==**(const *EffectHandle*&) const = default

auto **operator<=>**(const *EffectHandle*&) const = default

Public Members

EffectType **type** = EffectType::invalid()

int **subtype** = -1

ObjectHandle **creator**

Spell **spell_id** = *Spell*::*invalid*()

EffectCategory **category** = EffectCategory::magical

Effect ***effect** = nullptr

6.13.38 nw::Encounter

struct **Encounter** : public nw::ObjectBase

Public Functions

Encounter()

inline virtual *Common* ***as_common**() override

inline virtual const *Common* ***as_common**() const override

inline virtual *Encounter* ***as_encounter**() override

inline virtual const *Encounter* ***as_encounter**() const override

inline virtual bool **instantiate**() override

inline virtual *InternedString* **tag**() const override

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Versus* **versus_me**() const

inline virtual *Area* ***as_area**()

inline virtual const *Area* ***as_area**() const

inline virtual *Creature* ***as_creature**()

inline virtual const *Creature* ***as_creature**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Item* ***as_item**()

inline virtual const *Item* ***as_item**() const

inline virtual *Module* ***as_module**()

inline virtual const *Module* ***as_module()** const

inline virtual *Placeable* ***as_placeable()**

inline virtual const *Placeable* ***as_placeable()** const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

EncounterScripts **scripts**

std::vector<*SpawnCreature*> **creatures**

std::vector<glm::vec3> **geometry**

std::vector<*SpawnPoint*> **spawn_points**

int32_t **creatures_max** = -1

int32_t **creatures_recommended** = 0

int32_t **difficulty** = 0

int32_t **difficulty_index** = 0

uint32_t **faction** = 0

int32_t **reset_time** = 0

int32_t **respawns** = 0

int32_t **spawn_option** = 0

bool **active** = true

bool **player_only** = false

bool **reset** = true

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Encounter* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Encounter* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*encounter*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::ute

6.13.39 nw::EncounterScripts

struct **EncounterScripts**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

Resref **on_entered**

Resref **on_exhausted**

Resref **on_exit**

Resref **on_heartbeat**

Resref **on_user_defined**

6.13.40 nw::Equips

struct **Equips**

Public Functions

Equips(*Creature* *owner)

Equips(const *Equips*&) = delete

Equips(*Equips*&&) = default

Equips &**operator=**(const *Equips*&) = delete

Equips &**operator=**(*Equips*&&) = default

~Equips() = default

bool **instantiate**()

bool **from_json**(const nlohmann::json &archive, *SerializationProfile* profile)

nlohmann::json **to_json**(*SerializationProfile* profile) const

Public Members

nw::*Creature* ***owner_** = nullptr

std::array<*EquipItem*, 18> **equips**

6.13.41 nw::Erf

struct **Erf** : public nw::*Container*

Public Functions

Erf() = default

explicit **Erf**(const std::filesystem::path &path)

Erf(const *Erf* &) = delete

Erf(*Erf* &&other) = default

virtual **~Erf**() = default

bool **add**(*Resource* res, const *ByteArray* &bytes)

Adds resources from array of bytes.

bool **add**(const std::filesystem::path &path)

Adds resources from path.

size_t **erase**(const *Resource* &res)

Removes resource.

bool **merge**(const *Container* *container)

Merges the contents of another container.

bool **reload**()

Reloads *Erf*

> **Note:** *Erf::working_directory()* will not change

bool **save**() const

Saves Erf to *Erf::path()*

> **Note:** It's probably best to call *Erf::reload* after save.

bool **save_as**(const std::filesystem::path &path) const

Saves Erf to different path

> **Note:** Current *Erf* unmodified, to load *Erf* at new path a new *Erf* must be constructed.

virtual std::vector<*ResourceDescriptor*> **all**() const override

Get all resources.

virtual bool **contains**(*Resource* res) const override

Get if container contains resource.

virtual ResourceData **demand**(*Resource* res) const override

Reads resource data, empty ResourceData if no match.

virtual int **extract**(const std::regex &pattern, const std::filesystem::path &output) const override

Extract elements from a container by regex.

inline virtual const std::string &**name**() const override

Equivalent to `basename` *path()*

inline virtual const std::string &**path**() const override

Path to container, for basic containers, should be canonical.

virtual size_t **size**() const override

Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **stat**(const *Resource* &res) const override

Get some general data about a resource.

inline virtual bool **valid**() const noexcept override

Return true if loaded, false if not.

virtual void **visit**(std::function<void(const *Resource*&)> callback) const noexcept override

Visits all resources in a container.

Erf &**operator=**(const *Erf*&) = delete

Erf &**operator=**(*Erf*&&) = default

virtual int **extract_by_glob**(std::string_view glob, const std::filesystem::path &output) const

Extract elements from a container by glob pattern.

const std::filesystem::path &**working_directory**() const

Get container working directory.

Public Members

ErfType **type** = *ErfType*::*erf*

Erf type.

ErfVersion **version** = *ErfVersion*::*v1_0*

Version.

LocString **description**

Description.

6.13.42 nw::Faction

struct **Faction**

Public Functions

explicit **Faction**(const *Gff* &archive)

explicit **Faction**(const nlohmann::json &archive)

GffBuilder **serialize**() const

nlohmann::json **to_json**() const

Public Members

std::vector<*FactionInfo*> **factions**

std::vector<*Reputation*> **reputations**

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::fac

6.13.43 nw::FactionInfo

struct **FactionInfo**

Public Members

std::string **name**

uint32_t **parent** = std::numeric_limits<uint32_t>::max()

uint16_t **global** = 0

6.13.44 nw::Feat

struct **Feat**

Public Functions

bool **operator==**(const *Feat* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Feat* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Feat* **make**(int32_t id)

Makes a rule type

static inline constexpr *Feat* **invalid**()

Returns an invalid rule type

6.13.45 nw::FeatInfo

struct **FeatInfo**

Feat definition.

Public Functions

FeatInfo() = default

FeatInfo(const TwoDARowView &tda)

inline bool **valid**() const noexcept

Public Members

uint32_t **name** = 0xFFFFFFFF

uint32_t **description** = 0xFFFFFFFF

Resource **icon**

bool **all_can_use** = false

int **category** = -1

int **max_cr** = 0

int **spell** = -1

Feat **successor** = *Feat*::*invalid*()

float **cr_value** = 0.0f

int **uses** = 0

int **master** = 0

bool **target_self** = false

InternedString **constant**

int **tools_categories** = 0

bool **hostile** = false

bool **epic** = false

bool **requires_action** = false

Requirement **requirements**

6.13.46 nw::GameProfile

struct **GameProfile**

Abstract base class for game profiles.

Subclassed by *nwn1::Profile*

Public Functions

virtual **~GameProfile()** = default

virtual bool **load_rules()** const = 0

Loads game specific rules.

virtual bool **load_resources()** = 0

6.13.47 nw::Gff

struct **Gff**

Public Functions

Gff() = default

explicit **Gff**(const std::filesystem::path &file, nw::*LanguageID* lang = nw::*LanguageID*::*english*)

explicit **Gff**(ResourceData data, nw::*LanguageID* lang = nw::*LanguageID*::*english*)

GffStruct **toplevel**() const

Get the toplevel struct.

inline std::string_view **type**() const

Get *Gff* type.

bool **valid**() const

Get if *Gff* file successfully parsed.

inline std::string_view **version**() const

Get the *Gff* Version.

Public Members

GffHeader ***head_** = nullptr

GffLabel ***labels_** = nullptr

GffStructEntry ***structs_** = nullptr

GffFieldEntry ***fields_** = nullptr

uint32_t ***field_indices_** = nullptr

uint32_t ***list_indices_** = nullptr

6.13.48 nw::GffBuilder

struct **GffBuilder**

Public Functions

explicit **GffBuilder**(std::string_view type, std::string_view version = "V3.2")

size_t **add_label**(std::string_view name)

void **build**()

ByteArray **to_byte_array**() const

bool **write_to**(const std::filesystem::path &path) const

Public Members

GffBuilderStruct **top**

GffHeader **header**

ByteArray **data**

std::vector<*GffLabel*> **labels**

std::vector<uint32_t> **field_indices**

std::vector<uint32_t> **list_indices**

std::vector<GffFieldEntry> **field_entries**

std::vector<GffStructEntry> **struct_entries**

6.13.49 nw::GffBuilderField

struct **GffBuilderField**

Public Functions

explicit **GffBuilderField**(*GffBuilder* *parent_)

Public Members

GffBuilder ***parent** = nullptr

SerializationType::*type* **type**

uint32_t **index** = 0

uint32_t **label_index** = 0

uint32_t **data_or_offset** = 0

std::variant<*GffBuilderStruct*, *GffBuilderList*> **structures**

6.13.50 nw::GffBuilderList

struct **GffBuilderList**

Public Functions

GffBuilderList() = default

explicit **GffBuilderList**(*GffBuilder* *parent_)

GffBuilderStruct &**push_back**(uint32_t id)

inline size_t **size**() const noexcept

Public Members

GffBuilder ***parent** = nullptr

std::vector<*GffBuilderStruct*> **structs**

6.13.51 nw::GffBuilderStruct

struct **GffBuilderStruct**

Public Functions

GffBuilderStruct() = default

explicit **GffBuilderStruct**(*GffBuilder* *parent_)

template<typename **T**>
GffBuilderStruct &**add_field**(std::string_view name, const *T* &value)

GffBuilderList &**add_list**(std::string_view name)

GffBuilderStruct &**add_struct**(std::string_view name, uint32_t id_)

Public Members

GffBuilder ***parent** = nullptr

uint32_t **index** = 0

uint32_t **id** = 0

std::vector<*GffBuilderField*> **field_entries**

6.13.52 nw::GffField

struct **GffField**

Public Functions

template<typename **T**>
std::optional<*T*> **get**() const

Get the field's value.

template<typename **T**>
bool **get_to**(*T* &value) const

Get the field's value.

std::string_view **name**() const

Get label.

size_t **size**() const

If field is a list, returns size of list, else 0.

SerializationType::*type* **type**() const

Get field type.

inline bool **valid**() const noexcept

Get if field is valid.

GffStruct **operator[]**(size_t index) const

If field is a list, return struct at index, else invalid struct.

6.13.53 nw::GffLabel

struct **GffLabel**

Public Types

using **Storage** = std::array<char, *max_size*>

using **value_type** = typename *Storage*::value_type

using **size_type** = typename *Storage*::size_type

Public Functions

GffLabel()

GffLabel(const *GffLabel*&) = default

GffLabel(*Storage* data) noexcept

GffLabel(const char *string) noexcept

GffLabel(std::string_view string) noexcept

GffLabel &**operator=**(const *GffLabel*&) = default

bool **empty**() const noexcept

Checks if the underlying array has no non-null characters.

size_type **length**() const noexcept

Returns the number of char elements in the array, excluding nulls.

std::string **string**() const

Creates `std::string` of underlying array.

std::string_view **view**() const noexcept

Creates `std::string_view` of underlying array without null padding.

Public Static Attributes

static constexpr size_t **max_size** = 16

6.13.54 nw::GffStruct

struct **GffStruct**

Public Functions

bool **has_field**(std::string_view label) const

Check if a struct has a field.

inline uint32_t **id**() const

Get struct id.

template<typename **T**>
std::optional<*T*> **get**(std::string_view label, bool warn_missing = true) const

Gets a value from a field in the struct.

template<typename **T**>
bool **get_to**(std::string_view label, *T* &out, bool warn_missing = true) const

Gets a value from a field in the struct.

inline size_t **size**() const

Number of fields in the struct.

inline bool **valid**() const

Check if *Gff* has been parsed without error.

GffField **operator[]**(std::string_view label) const

Get field by label.

GffField **operator[]**(size_t index) const

Get field by index.

6.13.55 nw::Image

struct **Image**

Image Resource.

Read/Write Support:

- jpg, png, dds, tga (thanks to stb_image and SOIL)
- Bioware dds (thanks to NWNExplorer)

Todo:

plt

Note: Even though this supports writing images, this is **catagorically** not a tool for converting/compressing textures.

Public Functions

explicit **Image**(const std::filesystem::path &filename)

explicit **Image**(ResourceData data)

Image(*Image* &&other)

Image(const *Image* &other) = delete

Image &**operator=**(*Image* &&other)

Image &**operator=**(const *Image* &other) = delete

~Image()

uint32_t **channels**() const noexcept

Get BBP.

uint8_t ***data**()

Get raw data.

uint32_t **height**() const noexcept

Get height.

bool **is_bio_dds**() const noexcept

Returns true if image was loaded from a bioware dds file.

bool **valid**() const

Determine if successfully loaded.

uint32_t **width**() const noexcept

Get width.

bool **write_to**(const std::filesystem::path &filename) const

Write *Image* to file.

6.13.56 nw::Ini

struct **Ini**

Ini file format parser.

Lookup is by "<section>/<key>"

Note: This is read only currently.

Public Functions

Ini() = default

explicit **Ini**(const std::filesystem::path &filename)

explicit **Ini**(*ByteArray* bytes)

template<typename **T**>
std::optional<*T*> **get**(std::string key) const

Gets a value.

Template Parameters
T – int32_t, float, or std::string

Parameters
key –

Returns
std::optional<T>

bool **get_to**(std::string key, std::string &out) const

Gets string value.

bool **get_to**(std::string key, int &out) const

Gets int value.

bool **get_to**(std::string key, float &out) const

Gets float value.

bool **valid**() const noexcept

Deterimes if *Ini* file was successfully parsed.

6.13.57 nw::InstallInfo

struct **InstallInfo**

Public Members

std::filesystem::path **install**

std::filesystem::path **user**

GameVersion **version** = *GameVersion*::*invalid*

6.13.58 nw::InternedString

struct **InternedString**

Public Functions

InternedString() = default

inline explicit **InternedString**(const std::string *str)

bool **operator==**(const *InternedString* &rhs) const noexcept = default

auto **operator<=>**(const *InternedString* &rhs) const noexcept = default

inline std::string_view **view**() const noexcept

inline **operator bool**() const noexcept

6.13.59 nw::Inventory

struct **Inventory**

Public Functions

Inventory() = default

inline explicit **Inventory**(ObjectBase *owner_)

Inventory(const *Inventory*&) = delete

Inventory(*Inventory*&&) = default

Inventory &**operator=**(const *Inventory*&) = delete

Inventory &**operator=**(*Inventory*&&) = default

~**Inventory**() = default

bool **instantiate**()

bool **from_json**(const nlohmann::json &archive, *SerializationProfile* profile)

nlohmann::json **to_json**(*SerializationProfile* profile) const

Public Members

ObjectBase *owner

std::vector<*InventoryItem*> items

6.13.60 nw::InventoryItem

struct InventoryItem

Public Members

bool infinite = false

uint16_t pos_x

uint16_t pos_y

std::variant<*Resref* , *Item**> item

6.13.61 nw::Item

struct Item : public nw::ObjectBase

Public Functions

Item()

inline virtual *Common* *as_common() override

inline virtual const *Common* *as_common() const override

inline virtual *Item* *as_item() override

inline virtual const *Item* *as_item() const override

virtual bool instantiate() override

inline virtual *InternedString* tag() const override

inline ObjectHandle handle() const noexcept

inline void set_handle(ObjectHandle handle)

const *EffectArray* &effects() const

EffectArray &**effects**()

inline virtual *Versus* **versus_me**() const

inline virtual *Area* ***as_area**()

inline virtual const *Area* ***as_area**() const

inline virtual *Creature* ***as_creature**()

inline virtual const *Creature* ***as_creature**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Encounter* ***as_encounter**()

inline virtual const *Encounter* ***as_encounter**() const

inline virtual *Module* ***as_module**()

inline virtual const *Module* ***as_module**() const

inline virtual *Placeable* ***as_placeable**()

inline virtual const *Placeable* ***as_placeable**() const

inline virtual *Player* ***as_player**()

inline virtual const *Player* ***as_player**() const

inline virtual *Sound* ***as_sound**()

inline virtual const *Sound* ***as_sound**() const

inline virtual *Store* ***as_store**()

inline virtual const *Store* ***as_store**() const

inline virtual *Trigger* ***as_trigger**()

inline virtual const *Trigger* ***as_trigger**() const

inline virtual *Waypoint* ***as_waypoint**()

inline virtual const *Waypoint* ***as_waypoint**() const

Public Members

Common **common**

Inventory **inventory**

LocString **description**

LocString **description_id**

std::vector<*ItemProperty*> **properties**

int **armor_id** = -1

uint32_t **cost** = 0

uint32_t **additional_cost** = 0

nw::*BaseItem* **baseitem**

uint16_t **stacksize** = 1

uint8_t **charges** = 0

bool **cursed** = false

bool **identified** = false

bool **plot** = false

bool **stolen** = false

ItemModelType **model_type** = *ItemModelType*::*simple*

std::array<uint8_t, 6> **model_colors**

std::array<uint8_t, 19> **model_parts**

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Item* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Item* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **`json_archive_version`** = 1

static constexpr *ObjectType* **`object_type`** = *ObjectType*::*item*

static constexpr *ResourceType*::*type* **`restype`** = *ResourceType*::uti

6.13.62 nw::ItemColors

struct **`ItemColors`**

Public Types

enum **`type`**

Values:

enumerator **`cloth1`**

enumerator **`cloth2`**

enumerator **`leather1`**

enumerator **`leather2`**

enumerator **`metal1`**

enumerator **`metal2`**

6.13.63 nw::ItemModelParts

struct **`ItemModelParts`**

Public Types

enum **`type`**

Values:

enumerator **`model1`**

enumerator **model2**

enumerator **model3**

enumerator **armor_belt**

enumerator **armor_lbicep**

enumerator **armor_lfarm**

enumerator **armor_lfoot**

enumerator **armor_lhand**

enumerator **armor_lshin**

enumerator **armor_lshoul**

enumerator **armor_lthigh**

enumerator **armor_neck**

enumerator **armor_pelvis**

enumerator **armor_rbicep**

enumerator **armor_rfarm**

enumerator **armor_rfoot**

enumerator **armor_rhand**

enumerator **armor_robe**

enumerator **armor_rshin**

enumerator **armor_rshoul**

enumerator **armor_rthigh**

enumerator **armor_torso**

6.13.64 nw::ItemProperty

struct **ItemProperty**

Public Members

uint16_t **type** = std::numeric_limits<uint16_t>::max()

uint16_t **subtype** = std::numeric_limits<uint16_t>::max()

uint8_t **cost_table** = std::numeric_limits<uint8_t>::max()

uint16_t **cost_value** = std::numeric_limits<uint16_t>::max()

uint8_t **param_table** = std::numeric_limits<uint8_t>::max()

uint8_t **param_value** = std::numeric_limits<uint8_t>::max()

6.13.65 nw::Journal

struct **Journal**

Public Functions

explicit **Journal**(const *GffStruct* gff)

Public Members

std::vector<*JournalCategory*> **categories**

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::jrl

6.13.66 nw::JournalCategory

struct **JournalCategory**

Public Members

std::string **comment**

std::vector<*JournalEntry*> **entries**

LocString **name**

std::string **tag**

uint32_t **priority**

uint32_t **xp**

uint16_t **picture**

6.13.67 nw::JournalEntry

struct **JournalEntry**

Public Members

LocString **text**

uint32_t **id**

uint16_t **end**

6.13.68 nw::Key

struct **Key** : public nw::*Container*

Public Functions

explicit **Key**(std::filesystem::path path)

Key(const *Key*&) = delete

Key(*Key*&&) = default

virtual **~Key**() = default

inline bool **is_loaded**() const noexcept

Returns if *Key* file was successfully loaded.

virtual std::vector<*ResourceDescriptor*> **all**() const override

Get all resources.

virtual bool **contains**(*Resource* res) const override

Get if container contains resource.

virtual ResourceData **demand**(*Resource* res) const override

Reads resource data, empty ResourceData if no match.

virtual int **extract**(const std::regex &pattern, const std::filesystem::path &output) const override

Extract elements from a container by regex.

inline virtual const std::string &**name**() const override

Equivalent to `basename` *`path()`*

inline virtual const std::string &**path**() const override

Path to container, for basic containers, should be canonical.

virtual size_t **size**() const override

Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **stat**(const *Resource* &res) const override

Get some general data about a resource.

inline virtual bool **valid**() const noexcept override

Return true if loaded, false if not.

virtual void **visit**(std::function<void(const *Resource*&)> callback) const noexcept override

Visits all resources in a container.

Key &**operator=**(const *Key*&) = delete

Key &**operator=**(*Key*&&) = default

virtual int **extract_by_glob**(std::string_view glob, const std::filesystem::path &output) const

Extract elements from a container by glob pattern.

const std::filesystem::path &**working_directory**() const

Get container working directory.

6.13.69 nw::Language

struct **Language**

Constants and related properties for supported languages.

Note: Might not be identical to what the game uses… yet. Short codes taken from https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes. Encodings are probably right.

Public Static Functions

static std::string_view **encoding**(*LanguageID* lang)

Gets the encoding for a particular language.

static *LanguageID* **from_string**(std::string_view lang)

Converts string (short or long form) to ID.

static bool **has_feminine**(*LanguageID* lang)

Determines if language has feminine translations.

static std::pair<*LanguageID*, bool> **to_base_id**(uint32_t lang)

Convert runtime language identifier to base language and bool indicating masc/fem.

static uint32_t **to_runtime_id**(*LanguageID* lang, bool feminine = false)

Convert language ID to runtime identifier.

static std::string_view **to_string**(*LanguageID* lang, bool long_name = false)

Converts language to string form.

6.13.70 nw::Language::Properties

struct **Properties**

Public Members

LanguageID **id**

std::string_view **lang_short**

std::string_view **lang_long**

std::string_view **encoding**

bool **has_feminine**

6.13.71 nw::LevelHistory

struct **LevelHistory**

Encapsulates a players level up history.

Public Members

std::vector<LevelUp> **entries**

6.13.72 nw::LevelHistoryEntry

Warning: doxygenstruct: Cannot find class "nw::LevelHistoryEntry" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.73 nw::LevelStats

struct **LevelStats**

Public Functions

LevelStats() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

int **level**() const noexcept

Determines total level.

int **level_by_class**(*Class* id) const noexcept

Determines level by class.

size_t **position**(*Class* id) const noexcept

Returns the position of the class, or `npos`.

Public Members

std::array<*ClassEntry*, *max_classes*> **entries**

Public Static Attributes

static constexpr size_t **max_classes** = 8

static constexpr size_t **npos** = std::numeric_limits<size_t>::max()

6.13.74 nw::LocString

struct **LocString**

Public Types

using **LocStringPair** = std::pair<uint32_t, std::string>

using **Storage** = std::vector<*LocStringPair*>

using **size_type** = *Storage*::size_type

using **iterator** = *Storage*::iterator

using **const_iterator** = *Storage*::const_iterator

Public Functions

explicit **LocString**(uint32_t strref = std::numeric_limits<uint32_t>::max())

LocString(const *LocString*&) = default

LocString(*LocString*&&) = default

bool **add**(*LanguageID* language, std::string_view str, bool feminine = false)

Add a localized string.

std::string **get**(*LanguageID* language, bool feminine = false) const

Gets a localized string.

bool **contains**(*LanguageID* language, bool feminine = false) const

Determines if a localized string is set.

void **remove**(*LanguageID* language, bool feminine = false)

Removes a localized string.

size_type **size**() const

Gets number of localized strings.

uint32_t **strref**() const

Gets string reference.

iterator **begin**()
Iterators.

iterator **end**()

const_iterator **begin**() const

const_iterator **end**() const

LocString &**operator=**(const *LocString*&) = default
Operators.

LocString &**operator=**(*LocString*&&) = default

bool **operator==**(const *LocString* &other) const

6.13.75 nw::LocalData

struct **LocalData**

Public Functions

LocalData() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**(*SerializationProfile* profile) const

void **delete_float**(std::string_view var)

void **delete_int**(std::string_view var)

void **delete_object**(std::string_view var)

void **delete_string**(std::string_view var)

void **delete_location**(std::string_view var)

float **get_float**(std::string_view var) const

int32_t **get_int**(std::string_view var) const

ObjectID **get_object**(std::string_view var) const

std::string **get_string**(std::string_view var) const

Location **get_location**(std::string_view var) const

void **set_float**(std::string_view var, float value)

void **set_int**(std::string_view var, int32_t value)

void **set_object**(std::string_view var, *ObjectID* value)

void **set_string**(std::string_view var, std::string_view value)

void **set_location**(std::string_view var, *Location* value)

inline size_t **size**() const noexcept

Friends

friend bool **deserialize**(*LocalData* &self, const *GffStruct* &archive)

friend bool **serialize**(const *LocalData* &self, *GffBuilderStruct* &archive, *SerializationProfile* profile)

6.13.76 nw::LocalVar

struct **LocalVar**

Public Members

float **float_**

int32_t **integer**

ObjectID **object**

std::string **string**

Location **loc**

std::bitset<8> **flags**

6.13.77 nw::LocalVarType

struct **LocalVarType**

Public Static Attributes

static constexpr uint32_t **integer** = 1

static constexpr uint32_t **float_** = 2

static constexpr uint32_t **string** = 3

static constexpr uint32_t **object** = 4

static constexpr uint32_t **location** = 5

6.13.78 nw::Location

struct **Location**

Public Functions

Location()

inline **operator bool**()

bool **operator==**(const *Location*&) const = default

Public Members

ObjectID **area**

glm::vec3 **position**

glm::vec3 **orientation**

6.13.79 nw::Lock

struct **Lock**

Component for lockable objects.

Public Functions

Lock() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

std::string **key_name**

bool **key_required** = false

bool **lockable** = false

bool **locked** = false

uint8_t **lock_dc** = 0

uint8_t **unlock_dc** = 0

bool **remove_key** = false

6.13.80 nw::MasterFeat

struct **MasterFeat**

Public Functions

bool **operator==**(const *MasterFeat* &rhs) const = default
Defaulted equality operator

auto **operator<=>**(const *MasterFeat* &rhs) const = default
Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept
Returns rule type as value

inline constexpr size_t **idx**() const noexcept
Returns rule type as index

Public Static Functions

static inline constexpr *MasterFeat* **make**(int32_t id)
Makes a rule type

static inline constexpr *MasterFeat* **invalid**()
Returns an invalid rule type

6.13.81 nw::MasterFeatRegistry

struct **MasterFeatRegistry**

Public Functions

template<typename **T**>
void **add**(*T* type, *MasterFeat* mfeat, *Feat* feat)

void **clear**() noexcept

inline const std::vector<MasterFeatEntry> &**entries**() const noexcept

const *ModifierVariant* &**get_bonus**(*MasterFeat* mfeat) const

template<typename **T**>

void **remove**(*T* type, *MasterFeat* mfeat)

void **set_bonus**(*MasterFeat* mfeat, *ModifierVariant* bonus)

6.13.82 nw::Modifier

struct **Modifier**

Public Members

ModifierType **type** = *ModifierType*::*invalid*()

ModifierVariant **input**

InternedString **tagged**

ModifierSource **source** = ModifierSource::unknown

Requirement **requirement** = *Requirement*{}

Versus **versus** = {}

int **subtype** = -1

6.13.83 nw::ModifierRegistry

struct **ModifierRegistry**

Public Types

using **Storage** = std::vector<*Modifier*>

using **iterator** = *Storage*::iterator

using **const_iterator** = *Storage*::const_iterator

Public Functions

void **add**(*Modifier* mod)

Adds a modifier to the system.

iterator **begin**()

const_iterator **begin**() const

const_iterator **cbegin**() const

void **clear**()

Clears all modifiers.

iterator **end**()

const_iterator **end**() const

const_iterator **cend**() const

int **remove**(std::string_view tag)

Removes modifiers by tag.

Parameters

tag – if `string_view` ends with '*' then matches any tag that starts with `tag`

Returns

int number of modifiers affected

int **replace**(std::string_view tag, *ModifierVariant* value)

Replace modifier value.

Parameters

- **tag** – if `string_view` ends with '*' then matches any tag that starts with `tag`
- **value** – new value

Returns

int number of modifiers affected

int **replace**(std::string_view tag, const *Requirement* &req)

Replace modifier requirement.

Parameters

- **tag** – if `string_view` ends with '*' then matches any tag that starts with `tag`
- **req** – new requirement

Returns

int number of modifiers affected

size_t **size**() const

Gets the number of modifiers.

6.13.84 nw::ModifierType

struct **ModifierType**

Public Functions

bool **operator==**(const *ModifierType* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *ModifierType* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *ModifierType* **make**(int32_t id)

Makes a rule type

static inline constexpr *ModifierType* **invalid**()

Returns an invalid rule type

6.13.85 nw::Module

struct **Module** : public nw::ObjectBase

Public Types

using **AreaVariant** = std::variant<std::vector<*Resref*>, std::vector<*Area**>>

Public Functions

inline virtual *Module* ***as_module**() override

inline virtual const *Module* ***as_module**() const override

virtual bool **instantiate**() override

size_t **area_count**() const noexcept

const *Area* ***get_area**(size_t index) const

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Versus* **versus_me**() const

virtual *InternedString* **tag**() const

inline virtual *Area* ***as_area**()

inline virtual const *Area* ***as_area**() const

inline virtual *Common* ***as_common**()

inline virtual const *Common* ***as_common**() const

inline virtual *Creature* ***as_creature**()

inline virtual const *Creature* ***as_creature**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Encounter* ***as_encounter**()

inline virtual const *Encounter* ***as_encounter**() const

inline virtual *Item* ***as_item**()

inline virtual const *Item* ***as_item**() const

inline virtual *Placeable* ***as_placeable**()

inline virtual const *Placeable* ***as_placeable**() const

inline virtual *Player* ***as_player**()

inline virtual const *Player* ***as_player**() const

inline virtual *Sound* ***as_sound**()

inline virtual const *Sound* ***as_sound**() const

inline virtual *Store* ***as_store**()

inline virtual const *Store* ***as_store**() const

inline virtual *Trigger* ***as_trigger**()

inline virtual const *Trigger* ***as_trigger**() const

inline virtual *Waypoint* ***as_waypoint**()

inline virtual const *Waypoint* ***as_waypoint**() const

Public Members

LocalData **locals**

ModuleScripts **scripts**

AreaVariant **areas**

LocString **description**

Resref **entry_area**

glm::vec3 **entry_orientation**

glm::vec3 **entry_position**

std::vector<std::string> **haks**

ByteArray **id**

std::string **min_game_version**

LocString **name**

Resref **start_movie**

std::string **tag**

std::string **tlk**

uuids::uuid **uuid**

int32_t **creator** = 0

uint32_t **start_year**

uint32_t **version** = 3

uint16_t **expansion_pack** = 0

uint8_t **dawn_hour** = 0

uint8_t **dusk_hour** = 0

bool **is_save_game** = false

uint8_t **minutes_per_hour** = 0

uint8_t **start_day** = 0

uint8_t **start_hour** = 0

uint8_t **start_month** = 0

uint8_t **xpscale** = 0

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Module* *ent, const nlohmann::json &archive)

static bool **serialize**(const *Module* *ent, nlohmann::json &archive)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*module*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::ifo

6.13.86 nw::ModuleScripts

struct **ModuleScripts**

Public Functions

ModuleScripts() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

Resref **on_client_enter**

Resref **on_client_leave**

Resref **on_cutsnabort**

Resref **on_heartbeat**

Resref **on_item_acquire**

Resref **on_item_activate**

Resref **on_item_unaquire**

Resref **on_load**

Resref **on_player_chat**

Resref **on_player_death**

Resref **on_player_dying**

Resref **on_player_equip**

Resref **on_player_level_up**

Resref **on_player_rest**

Resref **on_player_uneqiup**

Resref **on_spawnbtndn**

Resref **on_start**

Resref **on_user_defined**

6.13.87 nw::NWSync

struct **NWSync**

Public Functions

NWSync()

explicit **NWSync**(const std::filesystem::path &path)

NWSync(const *NWSync*&) = delete

NWSync(*NWSync*&&) = default

~**NWSync**() = default

NWSyncManifest ***get**(std::string_view manifest)

Get a particular manifest as a container.

bool **is_loaded**() const noexcept

Get if *NWSync* was successfully loaded.

std::vector<std::string> **manifests**()

Get list of all manifests.

size_t **shard_count**() const noexcept

Get the number of shards.

inline sqlite3 ***meta**()

Get a pointer to the nwsyncmeta database.

inline std::vector<*sqlite3_ptr*> &**shards**()

List of all shards as active Sqlite3 connections.

NWSync &**operator=**(const *NWSync*&) = delete

NWSync &**operator=**(*NWSync*&&) = default

6.13.88 nw::NWSyncManifest

struct **NWSyncManifest** : public nw::*Container*

Abstracts over manifests, treating them as a *nw::Container*.

Public Functions

NWSyncManifest() = default

NWSyncManifest(std::string manifest, *NWSync* *parent)

virtual std::vector<*ResourceDescriptor*> **all**() const override

Get all resources.

virtual bool **contains**(*Resource* res) const override

Get if container contains resource.

virtual ResourceData **demand**(*Resource* res) const override

Reads resource data, empty ResourceData if no match.

virtual int **extract**(const std::regex &pattern, const std::filesystem::path &output) const override

Extract elements from a container by regex.

inline virtual const std::string &**name**() const override

Equivalent to `basename` *path()*

inline virtual const std::string &**path**() const override

Path to container, for basic containers, should be canonical.

inline virtual size_t **size**() const override

Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **stat**(const *Resource* &res) const override

Get some general data about a resource.

inline virtual bool **valid**() const noexcept override

Return true if loaded, false if not.

virtual void **visit**(std::function<void(const *Resource*&)> callback) const noexcept override

Visits all resources in a container.

virtual int **extract_by_glob**(std::string_view glob, const std::filesystem::path &output) const

Extract elements from a container by glob pattern.

const std::filesystem::path &**working_directory**() const

Get container working directory.

6.13.89 nw::Null

struct **Null**

Empty helper struct for *Variant*.

6.13.90 nw::Palette

struct **Palette**

Public Functions

explicit **Palette**(const *Gff* &gff)

~Palette() = default

inline uint8_t **max_id**() const noexcept

inline void **set_max_id**(uint8_t id) noexcept

inline bool **valid**() const noexcept

nlohmann::json **to_json**(nw::*ResourceType*::*type* restype) const

Public Members

PaletteTreeNode **root**

ResourceType::*type* **resource_type**

Resref **tileset**

bool **is_skeleton** = false

Public Static Attributes

static constexpr int **json_archive_version** = 1

6.13.91 nw::PaletteTreeNode

struct **PaletteTreeNode**

Public Functions

PaletteTreeNode() = default

Public Members

PaletteNodeType **type**

uint8_t **id** = std::numeric_limits<uint8_t>::max()

uint8_t **display** = 0

std::string **name**

uint32_t **strref** = std::numeric_limits<uint32_t>::max()

Resref **resref**

float **cr** = 0.0

std::string **faction**

std::vector<*PaletteTreeNode*> **children**

6.13.92 nw::Placeable

struct **Placeable** : public nw::ObjectBase

Public Functions

Placeable()

inline virtual *Common* ***as_common()** override

inline virtual const *Common* ***as_common()** const override

inline virtual *Placeable* ***as_placeable()** override

inline virtual const *Placeable* ***as_placeable()** const override

virtual bool **instantiate()** override

inline virtual *InternedString* **tag()** const override

inline ObjectHandle **handle()** const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects()** const

EffectArray &**effects()**

inline virtual *Versus* **versus_me()** const

inline virtual *Area* ***as_area()**

inline virtual const *Area* ***as_area()** const

inline virtual *Creature* ***as_creature()**

inline virtual const *Creature* ***as_creature()** const

inline virtual *Door* ***as_door()**

inline virtual const *Door* ***as_door()** const

inline virtual *Encounter* ***as_encounter()**

inline virtual const *Encounter* ***as_encounter()** const

inline virtual *Item* ***as_item()**

inline virtual const *Item* ***as_item()** const

inline virtual *Module* ***as_module()**

inline virtual const *Module* ***as_module()** const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

PlaceableScripts **scripts**

Inventory **inventory**

Lock **lock**

Trap **trap**

Resref **conversation**

LocString **description**

Saves **saves**

uint32_t **appearance**

uint32_t **faction** = 0

int16_t **hp** = 0

int16_t **hp_current** = 0

uint16_t **portrait_id**

PlaceableAnimationState **animation_state**

uint8_t **bodybag** = 0

uint8_t **hardness**

bool **has_inventory** = false

bool **interruptable** = 0

bool **plot** = 0

bool **static_** = false

bool **useable** = false

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Placeable* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Placeable* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*placeable*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::utp

6.13.93 nw::PlaceableScripts

struct **PlaceableScripts**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

Resref **on_click**

Resref **on_closed**

Resref **on_damaged**

Resref **on_death**

Resref **on_disarm**

Resref **on_heartbeat**

Resref **on_inventory_disturbed**

Resref **on_lock**

Resref **on_melee_attacked**

Resref **on_open**

Resref **on_spell_cast_at**

Resref **on_trap_triggered**

Resref **on_unlock**

Resref **on_used**

Resref **on_user_defined**

6.13.94 nw::Player

struct **Player** : public nw::*Creature*

Public Functions

inline virtual *Player* ***as_player()** override

inline virtual const *Player* ***as_player()** const override

inline virtual *InternedString* **tag()** const override

inline virtual *Common* ***as_common()** override

inline virtual const *Common* ***as_common()** const override

inline virtual *Creature* ***as_creature()** override

inline virtual const *Creature* ***as_creature()** const override

virtual bool **instantiate()** override

virtual *Versus* **versus_me()** const override

inline ObjectHandle **handle()** const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects()** const

EffectArray &**effects()**

inline virtual *Area* ***as_area()**

inline virtual const *Area* ***as_area()** const

inline virtual *Door* ***as_door()**

inline virtual const *Door* ***as_door()** const

inline virtual *Encounter* ***as_encounter()**

inline virtual const *Encounter* ***as_encounter()** const

inline virtual *Item* ***as_item()**

inline virtual const *Item* ***as_item()** const

inline virtual *Module* ***as_module()**

inline virtual const *Module* ***as_module()** const

inline virtual *Placeable* ***as_placeable()**

inline virtual const *Placeable* ***as_placeable()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

Appearance **appearance**

CombatInfo **combat_info**

Equips **equipment**

Inventory **inventory**

LevelStats **levels**

LevelHistory **history**

CreatureScripts **scripts**

CreatureStats **stats**

Resref **conversation**

std::string **deity**

LocString **description**

LocString **name_first**

LocString **name_last**

std::string **subrace**

float **cr** = 0.0

int32_t **cr_adjust** = 0

uint32_t **decay_time**

Race **race** = *Race*::*invalid*()

int32_t **walkrate** = 0

uint16_t **faction_id** = 0

int16_t **hp** = 0

int16_t **hp_current** = 0

int16_t **hp_max** = 0

int16_t **hp_temp** = 0

uint16_t **soundset**

int32_t **hasted** = 0

int32_t **size** = 0

uint8_t **bodybag** = 0

uint8_t **chunk_death** = 0

uint8_t **disarmable** = 0

uint8_t **gender** = 0

uint8_t **good_evil** = 50

uint8_t **interruptable** = 0

uint8_t **immortal** = 0

uint8_t **lawful_chaotic** = 50

uint8_t **lootable** = 0

uint8_t **pc** = 0

uint8_t **perception_range** = 0

bool **plot** = false

uint8_t **starting_package** = 0

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Player* *obj, const nlohmann::json &archive)

static bool **deserialize**(*Creature* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Creature* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*player*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::bic

6.13.95 nw::Plt

struct **Plt**

Implementation of Bioware's PLT file format.

Public Functions

Plt(std::filesystem::path file)

Plt(ResourceData data)

uint32_t **height**() const

Gets height.

const *PltPixel* ***pixels**() const

Gets pixel array.

bool **valid**() const

Determines if PLT was successfully parsed.

uint32_t **width**() const

Gets width.

6.13.96 nw::PltColors

struct **PltColors**

Plt Color Array

Note: This would be the colors that a player would select

Public Members

std::array<uint8_t, plt_layer_size> **data** = { }

6.13.97 nw::PltPixel

struct **PltPixel**

Plt Pixel.

Public Members

uint8_t **color**

PltLayer **layer**

6.13.98 nw::Qualifier

struct **Qualifier**

Public Members

Selector **selector**

absl::InlinedVector<*RuleValue*, 4> **params**

6.13.99 nw::Race

struct **Race**

Public Functions

bool **operator==**(const *Race* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Race* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Race* **make**(int32_t id)

Makes a rule type

static inline constexpr *Race* **invalid**()

Returns an invalid rule type

6.13.100 nw::RaceInfo

struct **RaceInfo**

Race definition.

Public Functions

RaceInfo() = default

RaceInfo(const TwoDARowView &tda)

inline bool **valid**() const noexcept

Public Members

uint32_t **name** = 0xFFFFFFFF

uint32_t **name_conversation** = 0xFFFFFFFF

uint32_t **name_conversation_lower** = 0xFFFFFFFF

uint32_t **name_plural** = 0xFFFFFFFF

uint32_t **description** = 0xFFFFFFFF

Resource **icon**

int **appearance** = 0

std::array<int, 6> **ability_modifiers**

int **favored_class** = 0

Resource **feats_table**

uint32_t **biography** = 0xFFFFFFFF

bool **player_race** = false

InternedString **constant**

int **age** = 1

int **toolset_class** = 0

float **cr_modifier** = 1.0f

int **feats_extra_1st_level** = 0

int **skillpoints_extra_per_level** = 0

int **skillpoints_1st_level_multiplier** = 0

int **ability_point_buy_number** = 0

int **feats_normal_level** = 0

int **feats_normal_amount** = 0

int **skillpoints_ability** = 0

6.13.101 nw::Reputation

struct **Reputation**

Public Members

uint32_t **faction_1**

uint32_t **faction_2**

uint32_t **reputation**

6.13.102 nw::Requirement

struct **Requirement**

Public Functions

explicit **Requirement**(bool conjunction_ = true)

explicit **Requirement**(std::initializer_list<*Qualifier*> quals, bool conjunction_ = true)

void **add**(*Qualifier* qualifier)

size_t **size**() const noexcept

Public Members

absl::InlinedVector<*Qualifier*, 8> **qualifiers**

bool **conjunction** = true

6.13.103 nw::Resource

struct **Resource**

A *nw::Resource* consists of a *nw::Resref* and a *nw::ResourceType*. Since NWN1/EE doesn't have any notion of hierarchical organization (paths, etc), it represents a fully-qualified resource identifier.

Public Functions

Resource() noexcept

Resource(const *Resref* &resref_, *ResourceType*::*type* type_) noexcept

Resource(std::string_view resref_, *ResourceType*::*type* type_) noexcept

Resource(const *Resource*&) = default

Resource(*Resource*&&) = default

std::string **filename**() const

Gets a Resrefs file name with extension.

bool **valid**() const noexcept

A resource is valid if resref is not empty and resref type is not invalid.

Resource &**operator=**(const *Resource*&) = default

Resource &**operator=**(*Resource*&&) = default

Public Members

Resref **resref**

ResourceType::*type* **type**

Public Static Functions

static *Resource* **from_filename**(std::string_view)

static *Resource* **from_path**(const std::filesystem::path &path)

6.13.104 nw::ResourceDescriptor

struct **ResourceDescriptor**

Public Functions

inline **operator bool**()

Public Members

Resource **name**

size_t **size** = 0

int64_t **mtime** = 0

const *Container* ***parent** = nullptr

6.13.105 nw::ResourceType

struct **ResourceType**

Resource type constants and helper functions.

Public Types

enum **type**

Enumeration of *Resource* types.

Values:

enumerator **invalid**

enumerator **container**

enumerator **gff_archive**

enumerator **movie**

enumerator **player**

enumerator **sound**

enumerator **texture**

enumerator **json**

enumerator **bmp**

enumerator **mve**

enumerator **tga**

enumerator **wav**

enumerator **plt**

enumerator **ini**

enumerator **bmu**

enumerator **mpg**

enumerator **txt**

enumerator **plh**

enumerator **tex**

enumerator **mdl**

enumerator **thg**

enumerator **fnt**

enumerator **lua**

enumerator **slt**

enumerator **nss**

enumerator **ncs**

enumerator **mod**

enumerator **are**

enumerator **set**

enumerator **ifo**

enumerator **bic**

enumerator **wok**

enumerator **twoda**

enumerator **tlk**

enumerator **txi**

enumerator **git**

enumerator **bti**

enumerator **uti**

enumerator **btc**

enumerator **utc**

enumerator **dlg**

enumerator **itp**

enumerator **btt**

enumerator **utt**

enumerator **dds**

enumerator **bts**

enumerator **uts**

enumerator **ltr**

enumerator **gff**

enumerator **fac**

enumerator **bte**

enumerator **ute**

enumerator **btd**

enumerator **utd**

enumerator **btp**

enumerator **utp**

enumerator **dft**

enumerator **gic**

enumerator **gui**

enumerator **css**

enumerator **ccs**

enumerator **btm**

enumerator **utm**

enumerator **dwk**

enumerator **pwk**

enumerator **btg**

enumerator **utg**

enumerator **jrl**

enumerator **sav**

enumerator **utw**

enumerator **fourpc**

enumerator **ssf**

enumerator **hak**

enumerator **nwm**

enumerator **bik**

enumerator **ndb**

enumerator **ptm**

enumerator **ptt**

enumerator **bak**

enumerator **dat**

enumerator **shd**

enumerator **xbc**

enumerator **wbm**

enumerator **mtr**

enumerator **ktx**

enumerator **ttf**

enumerator **sql**

enumerator **tml**

enumerator **sq3**

enumerator **lod**

enumerator **gif**

enumerator **png**

enumerator **jpg**

enumerator **caf**

enumerator **ids**

enumerator **erf**

enumerator **bif**

enumerator **key**

Public Static Functions

static inline *type* **from_extension**(std::string_view ext)

Converts extension to *ResourceType::type*.

static inline std::string **to_string**(*ResourceType*::*type* value)

Convert *ResourceType::type* to extension.

Note: The only compilers and standard libraries that are targeted have small string optimization, so there is no great overhead to just returning a `std::string`

Returns
extension or empty string on failure

static inline constexpr bool **check_category**(*ResourceType*::*type* category, *ResourceType*::*type* type)

6.13.106 nw::Resref

struct **Resref**

`nw::Resref` names a resource.

In NWN1/EE they are 16 character arrays, in NWN2 32 character arrays. These character arrays are case-insenstive.

Later evolutions of resrefs, in Dragon Age, were 32 utf16 character arrays; then ultimately seem to have been replaced by a combination of FNV hashes.

Currently only the NWN1/EE variety is supported.

Public Types

using **Storage** = std::array<char, *max_size*>

using **value_type** = typename *Storage*::value_type

using **size_type** = typename *Storage*::size_type

Public Functions

Resref()

Resref(const *Resref* &) = default

template<size_t **N**>
Resref(std::array<char, *N*> &string) noexcept

Resref(const char *string) noexcept

Resref(std::string_view string) noexcept

Resref &**operator=**(const *Resref* &) = default

const *Storage* &**data**() const noexcept

Get underlying storage.

bool **empty**() const noexcept

Checks if the underlying array has no non-null characters.

size_type **length**() const noexcept

Returns the number of char elements in the array, excluding nulls.

std::string **string**() const

Creates `std::string` of underlying array.

std::string_view **view**() const noexcept

Creates `std::string_view` of underlying array without null padding.

Public Static Attributes

static constexpr size_t **max_size** = 32

6.13.107 nw::RuleFlag

template<typename **T**, size_t **N** = 64>

struct **RuleFlag** : private std::bitset<64>

Public Types

using **Base** = std::bitset<*N*>

Public Functions

constexpr **RuleFlag**() = default

inline constexpr **RuleFlag**(unsigned long long val) noexcept

inline **RuleFlag**(*T* val) noexcept

inline explicit **RuleFlag**(std::string_view str)

inline bool **operator[]**(*T* pos) const

inline *RuleFlag* &**flip**(*T* pos)

inline *RuleFlag* &**reset**(*T* pos)

inline *RuleFlag* &**set**(*T* pos, bool value = true)

inline bool **test**(*T* pos) const

6.13.108 nw::RuleTypeArray

template<typename **RuleType**, typename **RuleTypeInfo**>

struct **RuleTypeArray**

Base template for rule type arrays.

Template Parameters

- **RuleType** –
- **RuleTypeInfo** –

Public Types

using **map_type** = absl::flat_hash_map<*InternedString*, *RuleType*, InternedStringHash, InternedStringEq>

Public Functions

inline const *RuleTypeInfo* ***get**(*RuleType* type) const noexcept

inline bool **is_valid**(*RuleType* type) const noexcept

inline *RuleType* **from_constant**(std::string_view constant) const

Public Members

std::vector<*RuleTypeInfo*> **entries**

map_type **constant_to_index**

6.13.109 nw::Save

struct **Save**

Public Functions

bool **operator==**(const *Save* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Save* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Members

int32_t **val** = -1

Public Static Functions

static inline constexpr *Save* **make**(int32_t id)

Makes a rule type

static inline constexpr *Save* **invalid**()

Returns an invalid rule type

6.13.110 nw::Saves

struct **Saves**

Public Members

int16_t **fort** = 0

int16_t **reflex** = 0

int16_t **will** = 0

6.13.111 nw::Selector

struct **Selector**

Public Members

SelectorType **type**

RuleValue **subtype** = { }

6.13.112 nw::SerializationType

struct **SerializationType**

Gff types, renamed for clarity.

Public Types

enum **type**

Values:

enumerator **invalid**

enumerator **uint8**

enumerator **int8**

enumerator **uint16**

enumerator **int16**

enumerator **uint32**

enumerator **int32**

enumerator **uint64**

enumerator **int64**

enumerator **float_**

enumerator **double_**

enumerator **string**

enumerator **resref**

enumerator **locstring**

enumerator **void_**

enumerator **struct_**

enumerator **list**

Public Static Functions

template<typename **T**>
static constexpr *SerializationType*::*type* **id**()

Convert type to *SerializationType*.

static constexpr std::string_view **to_string**(*SerializationType*::*type* type)

6.13.113 nw::Situation

struct **Situation**

Public Functions

bool **operator==**(const *Situation* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Situation* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Members

int32_t **val** = -1

Public Static Functions

static inline constexpr *Situation* **make**(int32_t id)

Makes a rule type

static inline constexpr *Situation* **invalid**()

Returns an invalid rule type

6.13.114 nw::Skill

struct **Skill**

Public Functions

bool **operator==**(const *Skill* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Skill* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Skill* **make**(int32_t id)

Makes a rule type

static inline constexpr *Skill* **invalid**()

Returns an invalid rule type

6.13.115 nw::SkillInfo

struct **SkillInfo**

Skill definition.

Public Functions

SkillInfo() = default

SkillInfo(const TwoDARowView &tda)

inline bool **valid**() const noexcept

Public Members

uint32_t **name** = 0xFFFFFFFF

uint32_t **description** = 0xFFFFFFFF

Resource **icon**

bool **untrained** = false

Ability **ability** = { }

bool **armor_check_penalty** = false

bool **all_can_use** = false

InternedString **constant**

bool **hostile** = false

6.13.116 nw::Sound

struct **Sound** : public nw::ObjectBase

Public Functions

Sound()

inline virtual *Common* ***as_common**() override

inline virtual const *Common* ***as_common**() const override

inline virtual *Sound* ***as_sound**() override

inline virtual const *Sound* ***as_sound**() const override

inline virtual bool **instantiate**() override

inline virtual *InternedString* **tag**() const override

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Versus* **versus_me**() const

inline virtual *Area* ***as_area**()

inline virtual const *Area* ***as_area**() const

inline virtual *Creature* ***as_creature**()

inline virtual const *Creature* ***as_creature**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Encounter* ***as_encounter**()

inline virtual const *Encounter* ***as_encounter**() const

inline virtual *Item* ***as_item**()

inline virtual const *Item* ***as_item**() const

inline virtual *Module* ***as_module**()

inline virtual const *Module* ***as_module**() const

inline virtual *Placeable* ***as_placeable**()

inline virtual const *Placeable* ***as_placeable**() const

inline virtual *Player* ***as_player**()

inline virtual const *Player* ***as_player**() const

inline virtual *Store* ***as_store**()

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

std::vector<*Resref*> **sounds**

float **distance_min** = 0.0f

float **distance_max** = 0.0f

float **elevation** = 0.0f

uint32_t **generated_type** = 0

uint32_t **hours** = 0

uint32_t **interval** = 0

uint32_t **interval_variation** = 0

float **pitch_variation** = 0.0f

float **random_x** = 0.0f

float **random_y** = 0.0f

bool **active** = 0

bool **continuous** = 0

bool **looping** = 0

bool **positional** = 0

uint8_t **priority** = 0

bool **random** = 0

bool **random_position** = 0

uint8_t **times** = 3

uint8_t **volume** = 100

uint8_t **volume_variation** = 0

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Sound* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static void **serialize**(const *Sound* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*sound*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::uts

6.13.117 nw::SpawnCreature

struct **SpawnCreature**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

int32_t **appearance**

float **cr**

Resref **resref**

bool **single_spawn**

6.13.118 nw::SpawnPoint

struct **SpawnPoint**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

float **orientation**

glm::vec3 **position**

6.13.119 nw::SpecialAbility

struct **SpecialAbility**

Public Members

uint16_t **spell**

uint8_t **level**

SpellFlags **flags** = *SpellFlags*::*none*

6.13.120 nw::Spell

struct **Spell**

Public Functions

bool **operator==**(const *Spell* &rhs) const = default

Defaulted equality operator

auto **operator<=>**(const *Spell* &rhs) const = default

Defaulted spaceship operator

inline constexpr int32_t **operator***() const noexcept

Returns rule type as value

inline constexpr size_t **idx**() const noexcept

Returns rule type as index

Public Static Functions

static inline constexpr *Spell* **make**(int32_t id)

Makes a rule type

static inline constexpr *Spell* **invalid**()

Returns an invalid rule type

6.13.121 nw::SpellBook

struct **SpellBook**

Public Functions

SpellBook()

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

bool **add_known_spell**(size_t level, *SpellEntry* entry)

Adds a known spell at level.

bool **add_memorized_spell**(size_t level, *SpellEntry* entry)

Adds a memorized spell at level.

size_t **get_known_spell_count**(size_t level) const

Gets the number of known at a given level.

size_t **get_memorized_spell_count**(size_t level) const

Gets the number of memorized at a given level.

SpellEntry **get_known_spell**(size_t level, size_t index) const

Gets a known spell entry.

SpellEntry **get_memorized_spell**(size_t level, size_t index) const

Gets a memorized spell entry.

void **remove_known_spell**(size_t level, *SpellEntry* entry)

Removes a known spell entry.

void **remove_memorized_spell**(size_t level, *SpellEntry* entry)

Removes a memorized spell entry.

Public Members

std::vector<std::vector<*SpellEntry*>> **known_**

std::vector<std::vector<*SpellEntry*>> **memorized_**

6.13.122 nw::SpellEntry

struct **SpellEntry**

Public Functions

bool **operator==**(const *SpellEntry*&) const = default

auto **operator<=>**(const *SpellEntry*&) const = default

Public Members

Spell **spell** = *Spell*::*invalid*()

SpellMetaMagic **meta** = *SpellMetaMagic*::*none*

SpellFlags **flags** = *SpellFlags*::*none*

6.13.123 nw::SpellInfo

struct **SpellInfo**

Public Functions

SpellInfo() = default

SpellInfo(const TwoDARowView &tda)

inline bool **valid**() const noexcept

Public Members

uint32_t **name** = 0xFFFFFFFF

Resource **icon**

nw::SpellSchool **school** = nw::SpellSchool::invalid()

SpellMetaMagic **metamagic** = *SpellMetaMagic*::*none*

int **innate_level** = 0

6.13.124 nw::Store

struct **Store** : public nw::ObjectBase

Public Functions

Store()

inline virtual *Common* ***as_common**() override

inline virtual const *Common* ***as_common**() const override

inline virtual *Store* ***as_store**() override

inline virtual const *Store* ***as_store**() const override

virtual bool **instantiate**() override

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Versus* **versus_me**() const

virtual *InternedString* **tag**() const

inline virtual *Area* ***as_area()**

inline virtual const *Area* ***as_area()** const

inline virtual *Creature* ***as_creature()**

inline virtual const *Creature* ***as_creature()** const

inline virtual *Door* ***as_door()**

inline virtual const *Door* ***as_door()** const

inline virtual *Encounter* ***as_encounter()**

inline virtual const *Encounter* ***as_encounter()** const

inline virtual *Item* ***as_item()**

inline virtual const *Item* ***as_item()** const

inline virtual *Module* ***as_module()**

inline virtual const *Module* ***as_module()** const

inline virtual *Placeable* ***as_placeable()**

inline virtual const *Placeable* ***as_placeable()** const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

StoreScripts **scripts**

StoreInventory **inventory**

int32_t **blackmarket_markdown** = 0

int32_t **identify_price** = 100

int32_t **markdown** = 0

int32_t **markup** = 0

int32_t **max_price** = -1

int32_t **gold** = -1

bool **blackmarket**

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Store* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Store* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*store*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::utm

6.13.125 nw::StoreInventory

struct **StoreInventory**

Store Inventory component.

Public Functions

StoreInventory() = default

StoreInventory(ObjectBase *owner)

void **set_owner**(ObjectBase *owner)

Sets inventory owner.

Public Members

Inventory **armor**

Inventory **miscellaneous**

Inventory **potions**

Inventory **rings**

Inventory **weapons**

std::vector<int32_t> **will_not_buy**

std::vector<int32_t> **will_only_buy**

6.13.126 nw::StoreScripts

struct **StoreScripts**

Public Members

Resref **on_closed**

Resref **on_opened**

6.13.127 nw::Tile

struct **Tile**

Public Functions

Tile() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

int32_t **id** = 0

int32_t **height** = 0

int32_t **orientation** = 0

uint8_t **animloop1** = 0

uint8_t **animloop2** = 0

uint8_t **animloop3** = 0

uint8_t **mainlight1** = 0

uint8_t **mainlight2** = 0

uint8_t **srclight1** = 0

uint8_t **srclight2** = 0

6.13.128 nw::Tlk

struct **Tlk**

Public Functions

explicit **Tlk**(*LanguageID* language = *LanguageID*::*english*)

explicit **Tlk**(std::filesystem::path filename)

Tlk(const *Tlk*&) = delete

Tlk(*Tlk*&&) = default

std::string **get**(uint32_t strref) const

Get a localized string.

LanguageID **language_id**() const noexcept

Get language ID.

bool **modified**() const noexcept

Is *Tlk* modfied.

void **save**()

Write TLK to file.

void **save_as**(const std::filesystem::path &path)
Write TLK to file.

void **set**(uint32_t strref, std::string_view string)
Set a localized string.

size_t **size**() const noexcept
Get the number of tlk entries.

Note: This is equivalent to the highest string reference, not the number of actual strings

bool **valid**() const noexcept
Get if successfully parsed.

inline std::string **operator[]**(uint32_t strref) const
Get a localized string.

Tlk &**operator=**(const *Tlk*&) = delete

Tlk &**operator=**(*Tlk*&&) = default

Public Static Attributes

static constexpr uint32_t **custom_flag** = 0x01000000

6.13.129 nw::TlkElement

struct **TlkElement**

Public Functions

inline **TlkElement**()

Public Members

uint32_t **flags**

std::array<char, 16> **sound**

uint32_t **unused**[2]

uint32_t **offset**

uint32_t **size**

float **snd_duration**

6.13.130 nw::TlkFlags

struct **TlkFlags**

Tlk Flags.

Public Static Attributes

static constexpr uint32_t **empty** = 0x0

static constexpr uint32_t **text** = 0x1

static constexpr uint32_t **sound** = 0x2

static constexpr uint32_t **sound_length** = 0x4

6.13.131 nw::TlkHeader

struct **TlkHeader**

Public Members

std::array<char, 4> **type** = {'T', 'L', 'K', ' '}

std::array<char, 4> **version** = {'V', '3', '.', '0'}

uint32_t **language_id** = 0

uint32_t **str_count** = 0

uint32_t **str_offset** = 0

6.13.132 nw::Tokenizer

struct **Tokenizer**

Public Functions

Tokenizer()

Tokenizer(std::string_view buffer, std::string_view comment, bool skip_newline = true)

std::string_view **current**() const

bool **is_newline**(std::string_view tk) const

size_t **line**() const

std::string_view **next**()

void **put_back**(std::string_view token)

void **set_buffer**(std::string_view buffer)

6.13.133 nw::Trap

struct **Trap**

Public Functions

Trap() = default

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

bool **is_trapped** = false

uint8_t **type** = 0

bool **detectable** = false

uint8_t **detect_dc** = 0

bool **disarmable** = false

uint8_t **disarm_dc** = 0

bool **one_shot** = false

6.13.134 nw::Trigger

struct **Trigger** : public nw::ObjectBase

Public Functions

Trigger()

inline virtual *Common* ***as_common**() override

inline virtual const *Common* ***as_common**() const override

inline virtual *Trigger* ***as_trigger**() override

inline virtual const *Trigger* ***as_trigger**() const override

inline virtual bool **instantiate**() override

inline virtual *InternedString* **tag**() const override

virtual *Versus* **versus_me**() const override

inline ObjectHandle **handle**() const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects**() const

EffectArray &**effects**()

inline virtual *Area* ***as_area**()

inline virtual const *Area* ***as_area**() const

inline virtual *Creature* ***as_creature**()

inline virtual const *Creature* ***as_creature**() const

inline virtual *Door* ***as_door**()

inline virtual const *Door* ***as_door**() const

inline virtual *Encounter* ***as_encounter**()

inline virtual const *Encounter* ***as_encounter**() const

inline virtual *Item* ***as_item**()

inline virtual const *Item* ***as_item**() const

inline virtual *Module* ***as_module**()

inline virtual const *Module* ***as_module**() const

inline virtual *Placeable* ***as_placeable**()

inline virtual const *Placeable* ***as_placeable**() const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Waypoint* ***as_waypoint()**

inline virtual const *Waypoint* ***as_waypoint()** const

Public Members

Common **common**

Trap **trap**

TriggerScripts **scripts**

std::vector<glm::vec3> **geometry**

std::string **linked_to**

uint32_t **faction** = 0

float **highlight_height** = 0.0f

int32_t **type** = 0

uint16_t **loadscreen** = 0

uint16_t **portrait** = 0

uint8_t **cursor** = 0

uint8_t **linked_to_flags** = 0

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Trigger* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

static bool **serialize**(const *Trigger* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Public Static Attributes

static constexpr int **json_archive_version** = 1

static constexpr *ObjectType* **object_type** = *ObjectType*::*trigger*

static constexpr *ResourceType*::*type* **restype** = *ResourceType*::utt

6.13.135 nw::TriggerScripts

struct **TriggerScripts**

Public Functions

bool **from_json**(const nlohmann::json &archive)

nlohmann::json **to_json**() const

Public Members

Resref **on_click**

Resref **on_disarm**

Resref **on_enter**

Resref **on_exit**

Resref **on_heartbeat**

Resref **on_trap_triggered**

Resref **on_user_defined**

6.13.136 nw::TwoDA

struct **TwoDA**

Public Functions

TwoDA() = default

TwoDA(const *TwoDA*&) = delete

TwoDA(*TwoDA*&&) = default

TwoDA &**operator=**(const *TwoDA*&) = delete

TwoDA &**operator=**(*TwoDA*&&) = default

explicit **TwoDA**(const std::filesystem::path &filename)

Constructs *TwoDA* object from a file.

explicit **TwoDA**(ResourceData data)

Constructs *TwoDA* object from an array of bytes.

size_t **column_index**(std::string_view column) const

Finds the index of a column, or -1.

size_t **columns**() const noexcept

Get the number of columns.

template<typename **T**>
std::optional<*T*> **get**(size_t row, size_t col) const

Gets an element.

template<typename **T**>
std::optional<*T*> **get**(size_t row, std::string_view col) const

Gets an element.

template<typename **T**>
bool **get_to**(size_t row, size_t col, *T* &out) const

Gets an element.

template<typename **T**>
bool **get_to**(size_t row, std::string_view col, *T* &out) const

Gets an element.

void **pad**(size_t count)

Pads the 2da with `count` rows.

TwoDARowView **row**(size_t row) const noexcept

Gets entire row as.

size_t **rows**() const noexcept

Number of rows.

template<typename **T**>

void **set**(size_t row, size_t col, const *T* &value)

Sets an element.

template<typename **T**>
void **set**(size_t row, std::string_view col, const *T* &value)

Sets an element.

bool **is_valid**() const noexcept

Is the 2da parsed without error.

Public Static Attributes

static constexpr size_t **npos** = std::numeric_limits<size_t>::max()

6.13.137 nw::TwoDACache

Warning: doxygenstruct: Cannot find class "nw::TwoDACache" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.138 nw::Variant

template<typename ...**Ts**>

struct **Variant**

Wrapper around std::variant.

Public Functions

Variant() = default

Variant(const *Variant*&) = default

Variant(*Variant*&&) = default

Variant &**operator=**(const *Variant*&) = default

Variant &**operator=**(*Variant*&&) = default

template<typename **T**>
inline **Variant**(*T* value)

template<typename **T**>
inline bool **is**() const noexcept

Checks variant value is T

template<typename **T**>
inline *T* &**as**()

Gets variant value as T

template<typename **T**>

inline const *T* &**as**() const

Gets variant value as T

template<typename **T**>
inline std::optional<*T*> **get**() const

Checks variant value is `std::optional<T>`

Note: This does entail a copy

inline bool **operator<**(const *Variant* &rhs) const noexcept

inline bool **operator==**(const *Variant* &rhs) const noexcept

inline bool **empty**() const noexcept

6.13.139 nw::Versus

struct **Versus**

Public Functions

bool **operator==**(const *Versus* &rhs) const = default

auto **operator<=>**(const *Versus* &rhs) const = default

inline **operator bool**() const noexcept

inline bool **match**(const *Versus* &rhs) const noexcept

Public Members

Race **race** = *Race*::*invalid*()

AlignmentFlags **align_flags** = *AlignmentFlags*::*none*

bool **trap** = false

6.13.140 nw::Waypoint

struct **Waypoint** : public nw::ObjectBase

Public Functions

Waypoint()

inline virtual *Common* ***as_common()** override

inline virtual const *Common* ***as_common()** const override

inline virtual *Waypoint* ***as_waypoint()** override

inline virtual const *Waypoint* ***as_waypoint()** const override

inline virtual bool **instantiate()** override

inline virtual *InternedString* **tag()** const override

inline ObjectHandle **handle()** const noexcept

inline void **set_handle**(ObjectHandle handle)

const *EffectArray* &**effects()** const

EffectArray &**effects()**

inline virtual *Versus* **versus_me()** const

inline virtual *Area* ***as_area()**

inline virtual const *Area* ***as_area()** const

inline virtual *Creature* ***as_creature()**

inline virtual const *Creature* ***as_creature()** const

inline virtual *Door* ***as_door()**

inline virtual const *Door* ***as_door()** const

inline virtual *Encounter* ***as_encounter()**

inline virtual const *Encounter* ***as_encounter()** const

inline virtual *Item* ***as_item()**

inline virtual const *Item* ***as_item()** const

inline virtual *Module* ***as_module()**

inline virtual const *Module* ***as_module()** const

inline virtual *Placeable* ***as_placeable()**

inline virtual const *Placeable* ***as_placeable()** const

inline virtual *Player* ***as_player()**

inline virtual const *Player* ***as_player()** const

inline virtual *Sound* ***as_sound()**

inline virtual const *Sound* ***as_sound()** const

inline virtual *Store* ***as_store()**

inline virtual const *Store* ***as_store()** const

inline virtual *Trigger* ***as_trigger()**

inline virtual const *Trigger* ***as_trigger()** const

Public Members

Common **common**

LocString **description**

Description of waypoint.

std::string **linked_to**

Tag of entity waypoint is linked to.

LocString **map_note**

Map not for player minimap.

uint8_t **appearance**

Appearance.

bool **has_map_note** = false

If true waypoint has map note.

bool **map_note_enabled** = false

If true show map note.

bool **instantiated_** = false

Public Static Functions

static bool **deserialize**(*Waypoint* *obj, const nlohmann::json &archive, *SerializationProfile* profile)

Deserializes entity from JSON.

static void **serialize**(const *Waypoint* *obj, nlohmann::json &archive, *SerializationProfile* profile)

Deserializes entity to JSON.

Public Static Attributes

static constexpr int **`json_archive_version`** = 1

static constexpr *ObjectType* **`object_type`** = *ObjectType*::*waypoint*

static constexpr *ResourceType*::*type* **`restype`** = *ResourceType*::utw

6.13.141 nw::Zip

struct **`Zip`** : public nw::*Container*

Public Functions

`Zip`(const std::filesystem::path &path)

`~Zip`()

virtual std::vector<*ResourceDescriptor*> **`all`**() const override
Get all resources.

virtual bool **`contains`**(*Resource* res) const override
Get if container contains resource.

virtual ResourceData **`demand`**(*Resource* res) const override
Reads resource data, empty ResourceData if no match.

virtual int **`extract`**(const std::regex &pattern, const std::filesystem::path &output) const override
Extract elements from a container by regex.

inline virtual const std::string &**`name`**() const override
Equivalent to `basename` *`path()`*

inline virtual const std::string &**`path`**() const override
Path to container, for basic containers, should be canonical.

virtual size_t **`size`**() const override
Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **`stat`**(const *Resource* &res) const override
Get some general data about a resource.

inline virtual bool **`valid`**() const noexcept override
Return true if loaded, false if not.

virtual void **`visit`**(std::function<void(const *Resource*&)> callback) const noexcept override
Visits all resources in a container.

virtual int **`extract_by_glob`**(std::string_view glob, const std::filesystem::path &output) const
Extract elements from a container by glob pattern.

const std::filesystem::path &**`working_directory`**() const
Get container working directory.

6.13.142 nw::ZipElement

struct **ZipElement**

Public Members

Resource **resref**

size_t **size**

6.13.143 nw::ZlibHeader

struct **ZlibHeader**

Public Members

uint32_t **version**

6.13.144 nw::ZstdHeader

struct **ZstdHeader**

Public Members

uint32_t **version**

uint32_t **dictionary**

6.13.145 nw::kernel::Config

struct **Config**

Public Functions

explicit **Config**() = default

void **initialize**(*ConfigOptions* options = {})

Initializes configuration system.

const std::filesystem::path &**install_path**() const noexcept

Game installation path.

const *ConfigOptions* &**options**() const noexcept

Gets installation info.

void **set_paths**(const std::filesystem::path install, const std::filesystem::path user)

Sets game paths.

> **Note:** If paths are unset, the kernel will attempt to find them.

void **set_version**(*GameVersion* version)

Sets game version.

const std::filesystem::path &**user_path**() const noexcept

Path to user directory.

GameVersion **version**() const noexcept

Gets games version.

6.13.146 nw::kernel::EffectSystem

struct **EffectSystem** : public nw::kernel::*Service*

Public Functions

virtual **~EffectSystem**() = default

bool **add**(EffectType type, EffectFunc apply, EffectFunc remove)

Adds an effect type to the registry.

bool **add**(ItemPropertyType type, ItemPropFunc generator)

Adds an item property type to the registry.

bool **apply**(ObjectBase *obj, *Effect* *effect)

Applies an effect to an object.

virtual void **clear**() override

Clears effect registry and all effects.

Effect ***create**(EffectType type)

Creates an effect.

void **destroy**(*Effect* *effect)

Destroys an effect.

std::pair<int, int> **effect_limits_ability**() const noexcept

Gets ability effect minimum and maximum.

std::pair<int, int> **effect_limits_armor_class**() const noexcept

Gets armor class effect minimum and maximum.

std::pair<int, int> **effect_limits_attack**() const noexcept

Gets attack effect minimum and maximum.

std::pair<int, int> **effect_limits_skill**() const noexcept

Gets skill effect minimum and maximum.

Effect ***generate**(const *ItemProperty* &property, *EquipIndex* index, *BaseItem* baseitem) const

Generates an effect from an item property.

virtual void **initialize**() override

Initialize effect system.

const *TwoDA* ***ip_cost_table**(size_t table) const

Gets an item property cost table.

const ItemPropertyDefinition ***ip_definition**(ItemPropertyType type) const

Gets an item property definition.

const *TwoDA* ***ip_param_table**(size_t table) const

Gets an item property param table.

bool **remove**(ObjectBase *obj, *Effect* *effect)

Removes an effect to an object.

void **set_effect_limits_ability**(int min, int max) noexcept

Sets ability effect minimum and maximum.

void **set_effect_limits_armor_class**(int min, int max) noexcept

Sets armor class effect minimum and maximum.

void **set_effect_limits_attack**(int min, int max) noexcept

Sets attack effect minimum and maximum.

void **set_effect_limits_skill**(int min, int max) noexcept

Sets skill effect minimum and maximum.

EffectSystemStats **stats**() const noexcept

Gets stats regarding the effect system.

6.13.147 nw::kernel::EventSystem

struct **EventSystem** : public nw::kernel::*Service*

Public Types

template<typename **T**>

using **storage** = std::priority_queue<*T*, std::vector<*T*>, std::greater<*T*>>

Public Functions

void **add**(EventType type, ObjectBase *object, void *data = nullptr)

int **process**()

inline virtual void **initialize**()
Initializes a service.

inline virtual void **clear**()
Clears a service.

Public Members

storage<EventHandle> **queue_**

6.13.148 nw::kernel::ObjectSystem

struct **ObjectSystem** : public nw::kernel::*Service*
The object system creates, serializes, and deserializes entities.

Public Functions

ObjectSystem() = default

ObjectSystem(const *ObjectSystem*&) = delete

ObjectSystem(*ObjectSystem*&&) = default

ObjectSystem &**operator=**(*ObjectSystem*&) = delete

ObjectSystem &**operator=**(*ObjectSystem*&&) = default

inline virtual **~ObjectSystem**()

virtual void **clear**() override
Destroys all objects.

inline virtual void **initialize**() override
Initializes a service.

void **destroy**(ObjectHandle obj)
Destroys a single object.

template<typename **T**>

T ***get**(ObjectHandle obj)

Gets an object.

ObjectBase ***get_object_base**(ObjectHandle obj) const

Gets an object.

ObjectBase ***get_by_tag**(std::string_view tag, int nth = 0) const

Gets object by tag.

ObjectBase ***alloc**(*ObjectType* object_type)

template<typename **T**>
T ***load**(const std::filesystem::path &archive, *SerializationProfile* profile = *SerializationProfile*::*blueprint*)

Loads an object from file system.

template<typename **T**>
T ***load**(std::string_view resref)

Loads an object from resource system.

template<typename **T**>
T ***load**(const *GffStruct* &archive)

Loads an object from gff isntance.

template<typename **T**>
T ***load**(const nlohmann::json &archive)

Loads an object from json isntance.

Player ***load_player**(std::string_view cdkey, std::string_view resref)

Loads an object from resource system.

template<typename **T**>
T ***make**()

Creates a new object.

Area ***make_area**(*Resref* area)

Creates an area object.

Module ***make_module**()

Creates a module object

> **Warning:** : `nw::kernel::resman().load_module(...)` **must** be called before this.

bool **valid**(ObjectHandle obj) const

Determines of object handle is valid.

6.13.149 nw::kernel::Resources

struct **Resources** : public nw::*Container*, public nw::kernel::*Service*

Public Types

using **SearchVector** = std::vector<LocatorPayload>

Public Functions

Resources(const *Resources* *parent = nullptr)

virtual **~Resources**() = default

virtual void **initialize**() override

Initializes resources management system.

inline virtual void **clear**() override

Clears a service.

bool **add_base_container**(const std::filesystem::path &path, const std::string &name, *ResourceType*::*type* restype = *ResourceType*::invalid)

Add a base container

Note: This anything that is BELOW the module in priority

bool **add_custom_container**(*Container* *container, bool take_ownership = true, *ResourceType*::*type* restype = *ResourceType*::invalid)

Add already created container

Note: These containers are above all others in priority

bool **add_override_container**(const std::filesystem::path &path, const std::string &name, *ResourceType*::*type* restype = *ResourceType*::invalid)

Add already created container.

Add override container

Note: This anything that is ABOVE the module in priority

void **clear_containers**()

Clears any custom loaded containers.

bool **load_module**(std::filesystem::path path, std::string_view manifest = {})

Loads container resources for a module.

void **load_module_haks**(const std::vector<std::string> &haks)

Loads module haks.

void **unload_module**()

Unloads module.

ResourceData **demand_server_vault**(std::string_view cdkey, std::string_view resref)

Demands a player character file.

ResourceData **demand_any**(*Resref* resref, std::initializer_list<*ResourceType*::*type*> restypes) const

Attempts to locate first matching resource type by container priority.

ResourceData **demand_in_order**(*Resref* resref, std::initializer_list<*ResourceType*::*type*> restypes) const

Attempts to locate first matching resource by resource type priority.

void **load_palette_textures**()

Image ***palette_texture**(*PltLayer* layer)

inline virtual std::vector<*ResourceDescriptor*> **all**() const override

Get all resources.

virtual bool **contains**(*Resource* res) const override

Get if container contains resource.

virtual ResourceData **demand**(*Resource* res) const override

Reads resource data, empty ResourceData if no match.

virtual int **extract**(const std::regex &pattern, const std::filesystem::path &output) const override

Extract elements from a container by regex.

inline virtual const std::string &**name**() const override

Equivalent to `basename` *`path()`*

inline virtual const std::string &**path**() const override

Path to container, for basic containers, should be canonical.

virtual size_t **size**() const override

Determines the size, if applicable, of the container.

virtual *ResourceDescriptor* **stat**(const *Resource* &res) const override

Get some general data about a resource.

inline virtual bool **valid**() const noexcept override

Return true if loaded, false if not.

virtual void **visit**(std::function<void(const *Resource*&)> callback) const noexcept override

Visits all resources in a container.

virtual int **extract_by_glob**(std::string_view glob, const std::filesystem::path &output) const

Extract elements from a container by glob pattern.

const std::filesystem::path &**working_directory**() const

Get container working directory.

6.13.150 nw::kernel::Rules

struct **Rules** : public nw::kernel::*Service*

Public Types

using **qualifier_type** = std::function<bool(const *Qualifier*&, const ObjectBase*)>

using **selector_type** = std::function<*RuleValue*(const *Selector*&, const ObjectBase*)>

Public Functions

virtual **~Rules**()

virtual void **initialize**() override

Initializes rules system.

virtual void **clear**() override

Clears rules system of all rules and cached 2da files.

bool **match**(const *Qualifier* &qual, const ObjectBase *obj) const

Match.

bool **meets_requirement**(const *Requirement* &req, const ObjectBase *obj) const

Meets requirements.

RuleValue **select**(const *Selector*&, const ObjectBase*) const

Select.

void **set_qualifier**(*qualifier_type* match)

Set rules qualifier.

void **set_selector**(*selector_type* selector)

Set rules selector.

Public Members

BaseItemArray **baseitems**

ClassArray **classes**

FeatArray **feats**

RaceArray **races**

SpellArray **spells**

SpellSchoolArray **spellschools**

SkillArray **skills**

MasterFeatRegistry **master_feats**

ModifierRegistry **modifiers**

6.13.151 nw::kernel::ScriptSystem

Warning: doxygenstruct: Cannot find class "nw::kernel::ScriptSystem" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.152 nw::kernel::Service

struct **Service**

Subclassed by *nw::kernel::EffectSystem*, *nw::kernel::EventSystem*, *nw::kernel::ObjectSystem*, *nw::kernel::Resources*, *nw::kernel::Rules*, *nw::kernel::Strings*, nw::kernel::TwoDACache

6.13.153 nw::kernel::Services

struct **Services**

Public Functions

Services()

void **start**()

Initializes kernel services.

void **shutdown**()

Shutsdown kernel services.

GameProfile ***profile**() const

Gets current game profile.

template<typename **T**>
T ***add**()

Adds a service.

template<typename **T**>
const *T* ***get**() const

Gets a service.

template<typename **T**>
T ***get_mut**()

Gets a service as non-const.

Public Members

std::unique_ptr<*Strings*> **strings**

std::unique_ptr<*Resources*> **resources**

std::unique_ptr<TwoDACache> **twoda_cache**

std::unique_ptr<*Rules*> **rules**

std::unique_ptr<*EffectSystem*> **effects**

std::unique_ptr<*ObjectSystem*> **objects**

std::unique_ptr<*EventSystem*> **events**

6.13.154 nw::kernel::Strings

struct **Strings** : public nw::kernel::*Service*

Public Functions

Strings() = default

virtual **~Strings**() = default

virtual void **initialize**() override

Initializes strings system.

inline virtual void **clear**() override

Initializes strings system.

std::string **get**(const *LocString* &locstring, bool feminine = false) const

Gets string by *LocString*

> **Note:** if *Tlk* strref, use that; if not look in localized strings

std::string **get**(uint32_t strref, bool feminine = false) const

Gets string by *Tlk* strref.

InternedString **get_interned**(std::string_view str) const

Gets interned string

> **Note:** Return will not be valid if there is no interned string

InternedString **intern**(std::string_view str)

Interns a string

> **Note:** Multiple calls to `intern` with the same string will and must return the same exact underlying string, such that equality can be determined by a comparison of pointers.

InternedString **intern**(uint32_t strref)

Interns a string by strref

> **Note:** Multiple calls to `intern` with the same string will and must return the same exact underlying string, such that equality can be determined by a comparison of pointers.

void **load_custom_tlk**(const std::filesystem::path &path)

Loads a modules custom *Tlk* and feminine version if available.

void **load_dialog_tlk**(const std::filesystem::path &path)

Loads a dialog *Tlk* and feminine version if available.

LanguageID **global_language**() const noexcept

Gets the language ID that is considered 'default'

> **Note:** This determines the character encoding of strings as they are stored in game resources, TLK, GFF, etc. In EE the only encoding that isn't CP1252 is Polish, so generally safe to not worry too much.

void **set_global_language**(*LanguageID* language) noexcept

Sets the language ID that is considered 'default'.

void **unload_custom_tlk**()

Unloads a modules custom *Tlk* and feminine version if available.

6.13.155 nw::model::AABBEntry

struct **AABBEntry**

Public Members

glm::vec3 **bmin**

glm::vec3 **bmax**

int32_t **leaf_face**

uint32_t **plane**

6.13.156 nw::model::AABBNode

struct **AABBNode** : public nw::model::*TrimeshNode*

Public Functions

AABBNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::vector<*AABBEntry*> **entries**

glm::vec3 **ambient**

bool **beaming**

glm::vec3 **bmin**

glm::vec3 **bmax**

std::string **bitmap**

glm::vec3 **center**

glm::vec3 **diffuse**

std::string **materialname**

bool **render** = {true}

std::string **renderhint**

bool **rotatetexture** = {false}

bool **shadow** = {false}

float **shininess**

glm::vec3 **specular**

std::array<std::string, 3> **textures**

uint32_t **tilefade** = {0}

int **transparencyhint** = {0}

bool **showdispl** = {false}

uint32_t **displtype** = {1}

uint32_t **lightmapped** = {0}

std::vector<std::string> **multimaterial**

std::vector<glm::vec3> **colors**

std::vector<Vertex> **vertices**

std::vector<uint16_t> **indices**

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.157 nw::model::Animation

struct **Animation** : public nw::model::*Geometry*

Public Functions

Animation(std::string name_)

virtual **~Animation**() = default

Node ***find**(const std::regex &re)

const *Node* ***find**(const std::regex &re) const

Public Members

float **length** = {1.0f}

float **transition_time** = {0.25f}

std::string **anim_root**

std::vector<*AnimationEvent*> **events**

std::string **name**

GeometryType **type**

std::vector<std::unique_ptr<*Node*>> **nodes**

6.13.158 nw::model::AnimationEvent

struct **AnimationEvent**

Public Members

float **time** = {0.0f}

std::string **name**

6.13.159 nw::model::AnimeshNode

struct **AnimeshNode** : public nw::model::*TrimeshNode*

Public Functions

AnimeshNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::vector<glm::vec3> **animtverts**

std::vector<glm::vec3> **animverts**

float **sampleperiod**

float **cliph** = {0.0f}

float **clipw** = {0.0f}

float **clipv** = {0.0f}

float **clipu** = {0.0f}

glm::vec3 **ambient**

bool **beaming**

glm::vec3 **bmin**

glm::vec3 **bmax**

std::string **bitmap**

glm::vec3 **center**

glm::vec3 **diffuse**

std::string **materialname**

bool **render** = {true}

std::string **renderhint**

bool **rotatetexture** = {false}

bool **shadow** = {false}

float **shininess**

glm::vec3 **specular**

std::array<std::string, 3> **textures**

uint32_t **tilefade** = {0}

int **transparencyhint** = {0}

bool **showdispl** = {false}

uint32_t **displtype** = {1}

uint32_t **lightmapped** = {0}

std::vector<std::string> **multimaterial**

std::vector<glm::vec3> **colors**

std::vector<Vertex> **vertices**

std::vector<uint16_t> **indices**

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.160 nw::model::CameraNode

struct **CameraNode** : public nw::model::*Node*

Public Functions

CameraNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.161 nw::model::ControllerKey

struct **ControllerKey**

Public Functions

inline **ControllerKey**(*InternedString* name_, uint32_t type_, int rows_, int key_offset_, int time_offset_, int data_offset_, int columns_, bool is_key_)

Public Members

InternedString **name**

uint32_t **type**

int **rows** = {0}

int **key_offset** = {0}

int **time_offset** = {0}

int **data_offset** = {0}

int **columns** = {0}

bool **is_key** = {false}

6.13.162 nw::model::ControllerType

struct **ControllerType**

Public Static Functions

static std::pair<uint32_t, uint32_t> **lookup**(std::string_view cont)

Public Static Attributes

static constexpr uint32_t **Position** = 8

static constexpr uint32_t **Orientation** = 20

static constexpr uint32_t **Scale** = 36

static constexpr uint32_t **Wirecolor** = 20004

static constexpr uint32_t **Color** = 76

static constexpr uint32_t **Radius** = 88

static constexpr uint32_t **ShadowRadius** = 96

static constexpr uint32_t **VerticalDisplacement** = 100

static constexpr uint32_t **Multiplier** = 140

static constexpr uint32_t **AlphaEnd** = 80

static constexpr uint32_t **AlphaStart** = 84

static constexpr uint32_t **BirthRate** = 88

static constexpr uint32_t **Bounce_Co** = 92

static constexpr uint32_t **ColorEnd** = 96

static constexpr uint32_t **ColorStart** = 108

static constexpr uint32_t **CombineTime** = 120

static constexpr uint32_t **Drag** = 124

static constexpr uint32_t **FPS** = 128

static constexpr uint32_t **FrameEnd** = 132

static constexpr uint32_t **FrameStart** = 136

static constexpr uint32_t **Grav** = 140

static constexpr uint32_t **LifeExp** = 144

static constexpr uint32_t **Mass** = 148

static constexpr uint32_t **P2P_Bezier2** = 152

static constexpr uint32_t **P2P_Bezier3** = 156

static constexpr uint32_t **ParticleRot** = 160

static constexpr uint32_t **RandVel** = 164

static constexpr uint32_t **SizeStart** = 168

static constexpr uint32_t **SizeEnd** = 172

static constexpr uint32_t **SizeStart_Y** = 176

static constexpr uint32_t **SizeEnd_Y** = 180

static constexpr uint32_t **Spread** = 184

static constexpr uint32_t **Threshold** = 188

static constexpr uint32_t **Velocity** = 192

static constexpr uint32_t **XSize** = 196

static constexpr uint32_t **YSize** = 200

static constexpr uint32_t **BlurLength** = 204

static constexpr uint32_t **LightningDelay** = 208

static constexpr uint32_t **LightningRadius** = 212

static constexpr uint32_t **LightningScale** = 216

static constexpr uint32_t **LightningSubDiv** = 220

static constexpr uint32_t **Detonate** = 228

static constexpr uint32_t **AlphaMid** = 464

static constexpr uint32_t **ColorMid** = 468

static constexpr uint32_t **PercentStart** = 480

static constexpr uint32_t **PercentMid** = 481

static constexpr uint32_t **PercentEnd** = 482

static constexpr uint32_t **SizeMid** = 484

static constexpr uint32_t **SizeMid_Y** = 488

static constexpr uint32_t **lock_axes** = 500

static constexpr uint32_t **spawn_type** = 501

static constexpr uint32_t **random** = 502

static constexpr uint32_t **inherit** = 503

static constexpr uint32_t **inherit_local** = 503

static constexpr uint32_t **SelfIllumColor** = 100

static constexpr uint32_t **Alpha** = 128

static const std::unordered_map<std::string_view, std::pair<uint32_t, uint32_t>> **map**

6.13.163 nw::model::DanglymeshNode

struct **DanglymeshNode** : public nw::model::*TrimeshNode*

Public Functions

DanglymeshNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::vector<float> **constraints**

float **displacement**

float **period**

float **tightness**

glm::vec3 **ambient**

bool **beaming**

glm::vec3 **bmin**

glm::vec3 **bmax**

std::string **bitmap**

glm::vec3 **center**

glm::vec3 **diffuse**

std::string **materialname**

bool **render** = {true}

std::string **renderhint**

bool **rotatetexture** = {false}

bool **shadow** = {false}

float **shininess**

glm::vec3 **specular**

std::array<std::string, 3> **textures**

uint32_t **tilefade** = {0}

int **transparencyhint** = {0}

bool **showdispl** = {false}

uint32_t **displtype** = {1}

uint32_t **lightmapped** = {0}

std::vector<std::string> **multimaterial**

std::vector<glm::vec3> **colors**

std::vector<Vertex> **vertices**

std::vector<uint16_t> **indices**

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.164 nw::model::DummyNode

struct **DummyNode** : public nw::model::*Node*

Public Functions

DummyNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.165 nw::model::EmitterFlag

struct **EmitterFlag**

Public Static Attributes

static constexpr uint32_t **P2P** = 0x0001

static constexpr uint32_t **P2PSel** = 0x0002

static constexpr uint32_t **AffectedByWind** = 0x0004

static constexpr uint32_t **IsTinted** = 0x0008

static constexpr uint32_t **Bounce** = 0x0010

static constexpr uint32_t **Random** = 0x0020

static constexpr uint32_t **Inherit** = 0x0040

static constexpr uint32_t **InheritVel** = 0x0080

static constexpr uint32_t **InheritLocal** = 0x0100

static constexpr uint32_t **Splat** = 0x0200

static constexpr uint32_t **InheritPart** = 0x0400

6.13.166 nw::model::EmitterNode

struct **EmitterNode** : public nw::model::*Node*

Public Functions

EmitterNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

float **blastlength** = {0.0f}

float **blastradius** = {0.0f}

std::string **blend**

std::string **chunkname**

float **deadspace** = {0.0f}

uint32_t **loop** = {0}

std::string **render**

uint32_t **renderorder** = {0}

int32_t **spawntype** = {0}

std::string **texture**

uint32_t **twosidedtex** = {0}

std::string **update**

uint32_t **xgrid** = {0}

uint32_t **ygrid** = {0}

uint32_t **flags** = {0}

uint32_t **render_sel** = {0}

uint32_t **blend_sel** = {0}

uint32_t **update_sel** = {0}

uint32_t **spawntype_sel** = {0}

float **opacity** = {0.0f}

std::string **p2p_type**

uint32_t **tilefade** = {0}

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.167 nw::model::Face

struct **Face**

Public Members

std::array<uint32_t, 3> **vert_idx**

int32_t **shader_group_idx**

std::array<uint32_t, 3> **tvert_idx**

uint32_t **material_idx**

6.13.168 nw::model::Geometry

struct **Geometry**

Subclassed by *nw::model::Animation*, *nw::model::Model*

Public Functions

Geometry(GeometryType type_ = GeometryType::geometry)

Geometry(*Geometry*&) = delete

virtual **~Geometry**() = default

Geometry &**operator=**(*Geometry*&) = delete

Node ***find**(const std::regex &re)

const *Node* ***find**(const std::regex &re) const

Public Members

std::string **name**

GeometryType **type**

std::vector<std::unique_ptr<*Node*>> **nodes**

6.13.169 nw::model::GeometryFlag

Warning: doxygenstruct: Cannot find class "nw::model::GeometryFlag" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.170 nw::model::GeometryType

Warning: doxygenstruct: Cannot find class "nw::model::GeometryType" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.171 nw::model::LightNode

struct **LightNode** : public nw::model::*Node*

Public Functions

LightNode(std::string name_)

virtual **~LightNode**() = default

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

int32_t **lensflares** = {0}

float **flareradius** = {0.0f}

float **multiplier** = {0.0f}

glm::vec3 **color**

std::vector<float> **flaresizes**

std::vector<float> **flarepositions**

std::vector<glm::vec3> **flarecolorshifts**

std::vector<std::string> **textures**

uint32_t **lightpriority** = {5}

int32_t **ambientonly** = {0}

bool **dynamic** = {true}

uint32_t **affectdynamic** = {1}

uint32_t **shadow** = {1}

uint32_t **generateflare** = {0}

uint32_t **fadinglight** = {1}

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.172 nw::model::Mdl

class **Mdl**

Implements Bioware MDL file format

> **Warning:** This is still incomplete

Public Functions

Mdl(const std::filesystem::path &filename)

Mdl(ResourceData data)

std::unique_ptr<*Node*> **make_node**(uint32_t type, std::string_view name)

bool **valid**() const

Public Members

Model **model**

6.13.173 nw::model::TextParser

class **TextParser**

Public Functions

TextParser(std::string_view buffer, *Mdl* *mdl)

bool **parse**()

6.13.174 nw::model::Model

struct **Model** : public nw::model::*Geometry*

Public Functions

Model()

Model(*Model*&) = delete

virtual **~Model**() = default

Model &**operator=**(*Model*&) = delete

Animation ***find_animation**(std::string_view name)

const *Animation* ***find_animation**(std::string_view name) const

Node ***find**(const std::regex &re)

const *Node* ***find**(const std::regex &re) const

Public Members

ModelClass **classification**

bool **ignorefog**

std::vector<std::unique_ptr<*Animation*>> **animations**

std::unique_ptr<*Mdl*> **supermodel**

glm::vec3 **bmin**

glm::vec3 **bmax**

float **radius**

float **animationscale**

std::string **supermodel_name**

std::string **file_dependency**

std::string **name**

GeometryType **type**

std::vector<std::unique_ptr<*Node*>> **nodes**

6.13.175 nw::model::ModelClass

Warning: doxygenstruct: Cannot find class "nw::model::ModelClass" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.176 nw::model::Node

struct **Node**

Subclassed by *nw::model::CameraNode*, *nw::model::DummyNode*, *nw::model::EmitterNode*, *nw::model::LightNode*, *nw::model::PatchNode*, *nw::model::ReferenceNode*, *nw::model::TrimeshNode*

Public Functions

Node(std::string name_, uint32_t type_)

virtual **~Node**() = default

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.177 nw::model::NodeFlags

struct **NodeFlags**

Public Static Attributes

static constexpr uint32_t **header** = 0x00000001

static constexpr uint32_t **light** = 0x00000002

static constexpr uint32_t **emitter** = 0x00000004

static constexpr uint32_t **camera** = 0x00000008

static constexpr uint32_t **reference** = 0x00000010

static constexpr uint32_t **mesh** = 0x00000020

static constexpr uint32_t **skin** = 0x00000040

static constexpr uint32_t **anim** = 0x00000080

static constexpr uint32_t **dangly** = 0x00000100

static constexpr uint32_t **aabb** = 0x00000200

static constexpr uint32_t **patch** = 0x00000400

6.13.178 nw::model::NodeType

struct **NodeType**

Public Static Functions

static inline uint32_t **from_string**(std::string_view str)

static inline constexpr std::string_view **to_string**(uint32_t value)

Public Static Attributes

static constexpr uint32_t **camera** = (*NodeFlags*::*header* | *NodeFlags*::*camera*)

static constexpr uint32_t **dummy** = *NodeFlags*::*header*

static constexpr uint32_t **emitter** = *NodeFlags*::*header* | *NodeFlags*::*emitter*

static constexpr uint32_t **light** = *NodeFlags*::*header* | *NodeFlags*::*light*

static constexpr uint32_t **reference** = *NodeFlags*::*header* | *NodeFlags*::*reference*

static constexpr uint32_t **patch** = *NodeFlags*::*header* | *NodeFlags*::*patch*

static constexpr uint32_t **trimesh** = *NodeFlags*::*header* | *NodeFlags*::*mesh*

static constexpr uint32_t **danglymesh** = *trimesh* | *NodeFlags*::*dangly*

static constexpr uint32_t **skin** = *trimesh* | *NodeFlags*::*skin*

static constexpr uint32_t **animmesh** = *trimesh* | *NodeFlags*::*anim*

static constexpr uint32_t **aabb** = *trimesh* | *NodeFlags*::*aabb*

6.13.179 nw::model::PatchNode

struct **PatchNode** : public nw::model::*Node*

Public Functions

PatchNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.180 nw::model::ReferenceNode

struct **ReferenceNode** : public nw::model::*Node*

Public Functions

ReferenceNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::string **refmodel**

bool **reattachable**

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.181 nw::model::SkinNode

struct **SkinNode** : public nw::model::*TrimeshNode*

Public Functions

SkinNode(std::string name_)

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

std::vector<SkinVertex> **vertices**

std::array<int16_t, 64> **bone_nodes**

glm::vec3 **ambient**

bool **beaming**

glm::vec3 **bmin**

glm::vec3 **bmax**

std::string **bitmap**

glm::vec3 **center**

glm::vec3 **diffuse**

std::string **materialname**

bool **render** = {true}

std::string **renderhint**

bool **rotatetexture** = {false}

bool **shadow** = {false}

float **shininess**

glm::vec3 **specular**

std::array<std::string, 3> **textures**

uint32_t **tilefade** = {0}

int **transparencyhint** = {0}

bool **showdispl** = {false}

uint32_t **displtype** = {1}

uint32_t **lightmapped** = {0}

std::vector<std::string> **multimaterial**

std::vector<glm::vec3> **colors**

std::vector<uint16_t> **indices**

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.182 nw::model::SkinWeight

Warning: doxygenstruct: Cannot find class "nw::model::SkinWeight" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.183 nw::model::TriangleMode

Warning: doxygenstruct: Cannot find class "nw::model::TriangleMode" in doxygen xml output for project "rollNW" from directory: build/xml/

6.13.184 nw::model::TrimeshNode

struct **TrimeshNode** : public nw::model::*Node*

Subclassed by *nw::model::AABBNode*, *nw::model::AnimeshNode*, *nw::model::DanglymeshNode*, *nw::model::SkinNode*

Public Functions

TrimeshNode(std::string name_, uint32_t type_ = *NodeType*::*trimesh*)

virtual **~TrimeshNode**() = default

void **add_controller_data**(std::string_view name_, uint32_t type_, std::vector<float> times_, std::vector<float> data_, int rows_, int columns_ = 1)

Adds a controller to a model node.

ControllerValue **get_controller**(uint32_t type_, bool key = false) const

Gets a controller to a model node.

Public Members

glm::vec3 **ambient**

bool **beaming**

glm::vec3 **bmin**

glm::vec3 **bmax**

std::string **bitmap**

glm::vec3 **center**

glm::vec3 **diffuse**

std::string **materialname**

bool **render** = {true}

std::string **renderhint**

bool **rotatetexture** = {false}

bool **shadow** = {false}

float **shininess**

glm::vec3 **specular**

std::array<std::string, 3> **textures**

uint32_t **tilefade** = {0}

int **transparencyhint** = {0}

bool **showdispl** = {false}

uint32_t **displtype** = {1}

uint32_t **lightmapped** = {0}

std::vector<std::string> **multimaterial**

std::vector<glm::vec3> **colors**

std::vector<Vertex> **vertices**

std::vector<uint16_t> **indices**

std::string **name**

const uint32_t **type**

bool **inheritcolor** = false

Node ***parent** = nullptr

std::vector<*Node**> **children**

std::vector<*ControllerKey*> **controller_keys**

std::vector<float> **controller_data**

6.13.185 nw::script::AssignExpression

struct **AssignExpression** : public nw::script::*Expression*

Public Functions

inline **AssignExpression**(*Expression* *lhs_, *NssToken* token, *Expression* *rhs_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***lhs** = nullptr

NssToken **op**

Expression ***rhs** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.186 nw::script::Ast

struct **Ast**

Public Functions

Ast() = default

Ast(const *Ast*&) = delete

Ast(*Ast*&&) = default

Ast &**operator=**(const *Ast*&) = delete

Ast &**operator=**(*Ast*&&) = default

template<typename **T**, typename ...**Args**>
inline *T* ***create_node**(*Args*&&... args)

inline void **accept**(*BaseVisitor* *visitor)

std::string_view **find_comment**(size_t line) const noexcept

Finds first comment that the source range of which ends on `line` or `line` - 1.

Public Members

std::vector<*Statement**> **decls**

std::vector<*Include*> **includes**

std::unordered_map<std::string, std::string> **defines**

std::vector<*Comment*> **comments**

std::vector<size_t> **line_map**

std::vector<std::unique_ptr<*AstNode*>> **nodes_**

6.13.187 nw::script::AstLocator

struct **AstLocator** : public nw::script::*BaseVisitor*

Public Functions

inline **AstLocator**(*Nss* *parent, std::string symbol, size_t line, size_t character)

inline *Symbol* **locate_in_dependencies**(const std::string &needle, bool is_type = false)

inline virtual void **visit**(*Ast* *script)

inline virtual void **visit**(*FunctionDecl* *decl)

inline virtual void **visit**(*FunctionDefinition* *decl)

inline virtual void **visit**(*StructDecl* *decl)

inline virtual void **visit**(*VarDecl* *decl)

inline virtual void **visit**(*AssignExpression* *expr)

inline virtual void **visit**(*BinaryExpression* *expr)

inline virtual void **visit**(*CallExpression* *expr)

inline virtual void **visit**(ComparisonExpression *expr)

inline virtual void **visit**(*ConditionalExpression* *expr)

inline virtual void **visit**(*DotExpression* *expr)

inline virtual void **visit**(EmptyExpression *expr)

inline virtual void **visit**(*GroupingExpression* *expr)

inline virtual void **visit**(*LiteralExpression* *expr)

inline virtual void **visit**(*LiteralVectorExpression* *expr)

inline virtual void **visit**(*LogicalExpression* *expr)

inline virtual void **visit**(*PostfixExpression* *expr)

inline virtual void **visit**(*UnaryExpression* *expr)

inline virtual void **visit**(*VariableExpression* *expr)

inline virtual void **visit**(*BlockStatement* *stmt)

inline virtual void **visit**(DeclList *stmt)

inline virtual void **visit**(*DoStatement* *stmt)

inline virtual void **visit**(EmptyStatement *stmt)

inline virtual void **visit**(*ExprStatement* *stmt)

inline virtual void **visit**(*IfStatement* *stmt)

inline virtual void **visit**(*ForStatement* *stmt)

inline virtual void **visit**(*JumpStatement* *stmt)

inline virtual void **visit**(*LabelStatement* *stmt)

inline virtual void **visit**(*SwitchStatement* *stmt)

inline virtual void **visit**(*WhileStatement* *stmt)

Public Members

const *Nss* ***parent_** = nullptr

std::string **symbol_**

SourcePosition **pos_**

bool **in_func_decl_** = false

bool **in_struct_decl_** = false

bool **found_** = false

Symbol **result_**

const *Declaration* ***last_seen_decl** = nullptr

const *DotExpression* ***dot** = nullptr

const *CallExpression* ***call** = nullptr

size_t **active_param** = 0

6.13.188 nw::script::AstNode

struct **AstNode**

Subclassed by *nw::script::Expression*, *nw::script::Statement*

Public Functions

virtual **~AstNode**() = default

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.189 nw::script::AstPrinter

struct **AstPrinter** : public nw::script::*BaseVisitor*

Public Functions

~AstPrinter() = default

inline virtual void **visit**(*Ast* *script) override

inline virtual void **visit**(*FunctionDecl* *decl) override

inline virtual void **visit**(*FunctionDefinition* *decl) override

inline virtual void **visit**(*StructDecl* *decl) override

inline virtual void **visit**(*VarDecl* *decl) override

inline virtual void **visit**(*AssignExpression* *expr) override

inline virtual void **visit**(*BinaryExpression* *expr) override

inline virtual void **visit**(*CallExpression* *expr) override

inline virtual void **visit**(ComparisonExpression *expr) override

inline virtual void **visit**(*ConditionalExpression* *expr) override

inline virtual void **visit**(*DotExpression* *expr) override

inline virtual void **visit**(EmptyExpression *expr) override

inline virtual void **visit**(*GroupingExpression* *expr) override

inline virtual void **visit**(*LiteralExpression* *expr) override

inline virtual void **visit**(*LiteralVectorExpression* *expr) override

inline virtual void **visit**(*LogicalExpression* *expr) override

inline virtual void **visit**(*PostfixExpression* *expr) override

inline virtual void **visit**(*UnaryExpression* *expr) override

inline virtual void **visit**(*VariableExpression* *expr) override

inline virtual void **visit**(*BlockStatement* *stmt) override

inline virtual void **visit**(DeclList *stmt) override

inline virtual void **visit**(*DoStatement* *stmt) override

inline virtual void **visit**(EmptyStatement*) override

inline virtual void **visit**(*ExprStatement* *stmt) override

inline virtual void **visit**(*IfStatement* *stmt) override

inline virtual void **visit**(*ForStatement* *stmt) override

inline virtual void **visit**(*JumpStatement* *stmt) override

inline virtual void **visit**(*LabelStatement* *stmt) override

inline virtual void **visit**(*SwitchStatement* *stmt) override

inline virtual void **visit**(*WhileStatement* *stmt) override

Public Members

std::stringstream **ss**

int **depth** = 0

6.13.190 nw::script::AstResolver

struct **AstResolver** : public nw::script::*BaseVisitor*

Public Types

using **ScopeMap** = std::unordered_map<std::string, ScopeDecl>

using **ScopeStack** = std::vector<*ScopeMap*>

using **EnvStack** = std::vector<immer::map<std::string, Export>>

Public Functions

inline **AstResolver**(*Nss* *parent, *Context* *ctx, bool command_script = false)

virtual **~AstResolver**() = default

inline void **begin_scope**(bool global = false)

inline void **declare**(*NssToken* token, *Declaration* *decl, bool is_type = false)

inline void **define**(*NssToken* token, bool is_type = false)

inline void **end_scope**(bool global = false)

inline immer::map<std::string, Export> **symbol_table**() const

inline const *Declaration* ***resolve**(std::string_view token, *SourceRange* range, bool is_type)

inline virtual void **visit**(*Ast* *script) override

inline void **match_function_decls**(const *FunctionDecl* *decl, const *FunctionDecl* *def)

inline bool **all_control_flow_paths_return**(const *AstNode* *node)

inline virtual void **visit**(*FunctionDecl* *decl) override

inline virtual void **visit**(*FunctionDefinition* *decl) override

inline virtual void **visit**(*StructDecl* *decl) override

inline virtual void **visit**(*VarDecl* *decl) override

inline virtual void **visit**(*AssignExpression* *expr) override

inline virtual void **visit**(*BinaryExpression* *expr) override

inline virtual void **visit**(*CallExpression* *expr) override

inline virtual void **visit**(ComparisonExpression *expr) override

inline virtual void **visit**(*ConditionalExpression* *expr) override

inline virtual void **visit**(*DotExpression* *expr) override

inline virtual void **visit**(EmptyExpression *expr) override

inline virtual void **visit**(*GroupingExpression* *expr) override

inline virtual void **visit**(*LiteralExpression* *expr) override

inline virtual void **visit**(*LiteralVectorExpression* *expr) override

inline virtual void **visit**(*LogicalExpression* *expr) override

inline virtual void **visit**(*PostfixExpression* *expr) override

inline virtual void **visit**(*UnaryExpression* *expr) override

inline virtual void **visit**(*VariableExpression* *expr) override

inline virtual void **visit**(*BlockStatement* *stmt) override

inline virtual void **visit**(DeclList *stmt) override

inline virtual void **visit**(*DoStatement* *stmt) override

inline virtual void **visit**(EmptyStatement *stmt) override

inline virtual void **visit**(*ExprStatement* *stmt) override

inline virtual void **visit**(*IfStatement* *stmt) override

inline virtual void **visit**(*ForStatement* *stmt) override

inline virtual void **visit**(*JumpStatement* *stmt) override

inline virtual void **visit**(*LabelStatement* *stmt) override

inline virtual void **visit**(*SwitchStatement* *stmt) override

inline virtual void **visit**(*WhileStatement* *stmt) override

Public Members

Nss ***parent_** = nullptr

Context ***ctx_** = nullptr

ScopeStack **scope_stack_**

EnvStack **env_stack_**

int **loop_stack_** = 0

int **switch_stack_** = 0

FunctionDefinition ***func_def_stack_** = nullptr

bool **is_command_script_** = false

6.13.191 nw::script::BaseVisitor

struct **BaseVisitor**

Subclassed by nw::script::AstConstEvaluator, nw::script::AstHinter, *nw::script::AstLocator*, *nw::script::AstPrinter*, *nw::script::AstResolver*

Public Functions

virtual **~BaseVisitor**() = default

virtual void **visit**(*Ast* *script) = 0

virtual void **visit**(*FunctionDecl* *decl) = 0

virtual void **visit**(*FunctionDefinition* *decl) = 0

virtual void **visit**(*StructDecl* *decl) = 0

virtual void **visit**(*VarDecl* *decl) = 0

virtual void **visit**(*AssignExpression* *expr) = 0

virtual void **visit**(*BinaryExpression* *expr) = 0

virtual void **visit**(*CallExpression* *expr) = 0

virtual void **visit**(ComparisonExpression *expr) = 0

virtual void **visit**(*ConditionalExpression* *expr) = 0

virtual void **visit**(*DotExpression* *expr) = 0

virtual void **visit**(EmptyExpression *expr) = 0

virtual void **visit**(*GroupingExpression* *expr) = 0

virtual void **visit**(*LiteralExpression* *expr) = 0

virtual void **visit**(*LiteralVectorExpression* *expr) = 0

virtual void **visit**(*LogicalExpression* *expr) = 0

virtual void **visit**(*PostfixExpression* *expr) = 0

virtual void **visit**(*UnaryExpression* *expr) = 0

virtual void **visit**(*VariableExpression* *expr) = 0

virtual void **visit**(*BlockStatement* *stmt) = 0

virtual void **visit**(DeclList *stmt) = 0

virtual void **visit**(*DoStatement* *stmt) = 0

virtual void **visit**(EmptyStatement *stmt) = 0

virtual void **visit**(*ExprStatement* *stmt) = 0

virtual void **visit**(*IfStatement* *stmt) = 0

virtual void **visit**(*ForStatement* *stmt) = 0

virtual void **visit**(*JumpStatement* *stmt) = 0

virtual void **visit**(*LabelStatement* *stmt) = 0

virtual void **visit**(*SwitchStatement* *stmt) = 0

virtual void **visit**(*WhileStatement* *stmt) = 0

6.13.192 nw::script::BinaryExpression

struct **BinaryExpression** : public nw::script::*Expression*

Public Functions

inline **BinaryExpression**(*Expression* *lhs_, *NssToken* token, *Expression* *rhs_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***lhs** = nullptr

NssToken **op**

Expression ***rhs** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.193 nw::script::BlockStatement

struct **BlockStatement** : public nw::script::*Statement*

Public Functions

BlockStatement() = default

BlockStatement(*BlockStatement*&) = delete

BlockStatement &**operator=**(const *BlockStatement*&) = delete

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

std::vector<*Statement**> **nodes**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.194 nw::script::CallExpression

struct **CallExpression** : public nw::script::*Expression*

Public Functions

inline explicit **CallExpression**(*Expression* *expr_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***expr** = nullptr

std::vector<*Expression**> **args**

SourceRange **arg_range**

std::vector<*SourceRange*> **comma_ranges**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.195 nw::script::Comment

struct **Comment**

Abstracts a comment.

Public Functions

inline void **append**(std::string_view comment, *SourceLocation* range)

Public Members

SourceLocation **range_**

std::string **comment_**

6.13.196 nw::script::ConditionalExpression

struct **ConditionalExpression** : public nw::script::*Expression*

Public Functions

inline **ConditionalExpression**(*Expression* *expr_, *Expression* *true_branch_, *Expression* *false_branch_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***test** = nullptr

Expression ***true_branch** = nullptr

Expression ***false_branch** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.197 nw::script::Context

struct **Context**

Public Functions

Context(std::vector<std::string> include_paths = { }, std::string command_script = "nwscript")

virtual **~Context**() = default

void **add_include_path**(const std::filesystem::path &path)

Adds include path to internal resman.

Nss ***get**(*Resref* resref, bool command_script = false)

Gets a script from internal resman.

inline const *Nss* ***command_script**() const noexcept

Gets command script.

virtual void **register_default_types**()

virtual void **register_engine_types**()

size_t **type_id**(std::string_view type_name, bool define = false)

size_t **type_id**(*Type* type_name, bool define = false)

std::string_view **type_name**(size_t type_id)

virtual size_t **type_check_binary_op**(*NssToken* op, size_t lhs, size_t rhs)

virtual bool **is_type_convertible**(size_t lhs, size_t rhs)

virtual void **lexical_diagnostic**(*Nss* *script, std::string_view msg, bool is_warning, *SourceRange* range)

virtual void **parse_diagnostic**(*Nss* *script, std::string_view msg, bool is_warning, *SourceRange* range)

virtual void **semantic_diagnostic**(*Nss* *script, std::string_view msg, bool is_warning, *SourceRange* range)

Public Members

std::vector<std::string> **include_paths_**

absl::flat_hash_map<*Resource*, std::unique_ptr<*Nss*>> **dependencies_**

std::vector<IncludeStackEntry> **include_stack_**

std::vector<IncludeStackEntry> **preprocessed_**

kernel::*Resources* **resman_**

std::string **command_script_name_**

Nss ***command_script_** = nullptr

absl::flat_hash_map<std::string, size_t> **type_map_**

std::vector<std::string> **type_array_**

std::vector<*StructDecl**> **struct_stack_**

6.13.198 nw::script::Declaration

struct **Declaration** : public nw::script::*Statement*

Subclassed by nw::script::DeclList, *nw::script::FunctionDecl*, *nw::script::FunctionDefinition*, *nw::script::StructDecl*, *nw::script::VarDecl*

Public Functions

virtual std::string **identifier**() const = 0

virtual *SourceRange* **range**() const noexcept

virtual *SourceRange* **selection_range**() const noexcept

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Type **type**

SourceRange **range_selection_**

std::string_view **view**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.199 nw::script::Diagnostic

struct **Diagnostic**

Public Members

DiagnosticType **type**

DiagnosticSeverity **severity**

std::string **script**

std::string **message**

SourceRange **location**

6.13.200 nw::script::DoStatement

struct **DoStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Statement ***block** = nullptr

Expression ***expr** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.201 nw::script::DotExpression

struct **DotExpression** : public nw::script::*Expression*

Public Functions

inline **DotExpression**(*Expression* *lhs_, *NssToken* token, *Expression* *rhs_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***lhs** = nullptr

NssToken **dot**

Expression ***rhs** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.202 nw::script::ExprStatement

struct **ExprStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***expr** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.203 nw::script::Expression

struct **Expression** : public nw::script::*AstNode*

Subclassed by *nw::script::AssignExpression*, *nw::script::BinaryExpression*, *nw::script::CallExpression*, nw::script::ComparisonExpression, *nw::script::ConditionalExpression*, *nw::script::DotExpression*, nw::script::EmptyExpression, *nw::script::GroupingExpression*, *nw::script::LiteralExpression*, *nw::script::LiteralVectorExpression*, *nw::script::LogicalExpression*, *nw::script::PostfixExpression*, *nw::script::UnaryExpression*, *nw::script::VariableExpression*

Public Functions

virtual **~Expression**() = default

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.204 nw::script::ForStatement

struct **ForStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

AstNode ***init** = nullptr

Expression ***check** = nullptr

Expression ***inc** = nullptr

Statement ***block** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.205 nw::script::FunctionDecl

struct **FunctionDecl** : public nw::script::*Declaration*

Public Functions

FunctionDecl() = default

FunctionDecl(*FunctionDecl*&) = delete

FunctionDecl &**operator=**(const *FunctionDecl*&) = delete

inline virtual std::string **identifier**() const override

virtual *SourceRange* **range**() const noexcept

virtual *SourceRange* **selection_range**() const noexcept

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **identifier_**

std::vector<*VarDecl**> **params**

Type **type**

SourceRange **range_selection_**

std::string_view **view**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.206 nw::script::FunctionDefinition

struct **FunctionDefinition** : public nw::script::*Declaration*

Public Functions

inline virtual std::string **identifier**() const override

virtual *SourceRange* **range**() const noexcept

virtual *SourceRange* **selection_range**() const noexcept

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

FunctionDecl ***decl_inline** = nullptr

BlockStatement ***block** = nullptr

const *FunctionDecl* ***decl_external** = nullptr

Type **type**

SourceRange **range_selection_**

std::string_view **view**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.207 nw::script::GroupingExpression

struct **GroupingExpression** : public nw::script::*Expression*

Public Functions

inline explicit **GroupingExpression**(*Expression* *expr_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***expr** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.208 nw::script::IfStatement

struct **IfStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***expr** = nullptr

Statement ***if_branch** = nullptr

Statement ***else_branch** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.209 nw::script::Include

struct **Include**

Abstracts an script include.

Public Members

std::string **resref**

Resref of included script.

SourceRange **location**

Source location in script.

Nss ***script** = nullptr

Loaded script.

int **used** = 0

Number of times include is used in script file.

6.13.210 nw::script::InlayHint

struct **InlayHint**

Public Members

std::string **message**

SourcePosition **position**

6.13.211 nw::script::JumpStatement

struct **JumpStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **op**

Expression ***expr** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.212 nw::script::LabelStatement

struct **LabelStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **type**

Expression ***expr** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.213 nw::script::LiteralExpression

struct **LiteralExpression** : public nw::script::*Expression*

Public Functions

inline explicit **LiteralExpression**(*NssToken* token)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **literal**

Variant<int32_t, float, std::string, *Location*, *ObjectID*> **data**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.214 nw::script::LiteralVectorExpression

struct **LiteralVectorExpression** : public nw::script::*Expression*

Public Functions

inline explicit **LiteralVectorExpression**(*NssToken* x_, *NssToken* y_, *NssToken* z_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **x**

NssToken **y**

NssToken **z**

glm::vec3 **data**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.215 nw::script::LogicalExpression

struct **LogicalExpression** : public nw::script::*Expression*

Public Functions

inline **LogicalExpression**(*Expression* *lhs_, *NssToken* token, *Expression* *rhs_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***lhs** = nullptr

NssToken **op**

Expression ***rhs** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.216 nw::script::Nss

struct **Nss**

Public Functions

explicit **Nss**(const std::filesystem::path &filename, *Context* *ctx, bool command_script = false)

explicit **Nss**(std::string_view script, *Context* *ctx, bool command_script = false)

explicit **Nss**(ResourceData data, *Context* *ctx, bool command_script = false)

void **add_diagnostic**(*Diagnostic* diagnostic)

Add diagnostic to script.

Ast &**ast**()

Gets parsed AST.

const *Ast* &**ast**() const

Gets parsed ast.

void **complete**(const std::string &needle, CompletionContext &out, bool no_filter = false) const

Generates a list of potential completions (excluding dependencies)

void **complete_at**(const std::string &needle, size_t line, size_t character, CompletionContext &out, bool no_filter = false)

Get all completions (including dependencies)

void **complete_dot**(const std::string &needle, size_t line, size_t character, std::vector<*Symbol*> &out, bool no_filter = false)

Get all completions (including dependencies)

Context ***ctx**() const

Script context.

Symbol **declaration_to_symbol**(const *Declaration* *decl) const

Converts declaration to symbol

Note: *Declaration* must be in script

---

std::vector<std::string> **dependencies**() const

Returns all transitive dependencies in 'preprocessed' order, i.e. *dependencies()*[n] was include before *dependencies()*[n+1]

const std::vector<*Diagnostic*> &**diagnostics**() const noexcept

Gets script diagnostics.

inline size_t **errors**() const noexcept

Returns how many errors were found during parsing.

inline immer::map<std::string, Export> **exports**() const noexcept

Table of symbols exported from script.

inline size_t **export_count**() const noexcept

Count of symbols exported from script.

inline void **increment_errors**() noexcept

Increments error count.

inline void **increment_warnings**() noexcept

Increments warning count.

std::vector<*InlayHint*> **inlay_hints**(*SourceRange* range)

inline bool **is_command_script**() const noexcept

Is script a command script.

Symbol **locate_export**(const std::string &symbol, bool is_type, bool search_dependencies = false) const

Locate export, i.e. a top level symbols.

Symbol **locate_symbol**(const std::string &symbol, size_t line, size_t character)

Locate symbol in source file.

std::string_view **name**() const noexcept

Script name.

void **parse**()

Parses script file.

void **process_includes**(*Nss* *parent = nullptr)

Process includes recursively.

void **resolve**()

Resolves and type checks the *Ast*.

void **set_name**(const std::string &new_name)

Sets a scripts name.

SignatureHelp **signature_help**(size_t line, size_t character)

std::string_view **text**() const noexcept

Gets text of script.

std::string_view **view_from_range**(*SourceRange* range) const noexcept

Gets a view of source file in specified range.

inline size_t **warnings**() const noexcept

Returns how many warnings were found during parsing.

6.13.217 nw::script::NssLexer

struct **NssLexer**

Public Functions

explicit **NssLexer**(std::string_view buffer, *Context* *ctx, *Nss* *parent = nullptr)

NssToken **next**()

const *NssToken* &**current**() const

const char ***data**() const

Public Members

std::vector<size_t> **line_map**

6.13.218 nw::script::NssParser

struct **NssParser**

Public Functions

explicit **NssParser**(std::string_view view, *Context* *ctx, *Nss* *parent = nullptr)

NssToken **advance**()

Advances the token stream.

bool **check**(std::initializer_list<*NssTokenType*> types) const

Checks if next token matches a particular type.

Note: **Does not** advance the token stream

Parameters

types – An initializer list of token types

Returns

True if there is a match

bool **check_is_type**() const

Checks if next token matches a particular type.

Note: **Does not** advance the token stream

Returns

True if there is a match

NssToken **consume**(*NssTokenType* type, std::string_view error)

Consumes a token.

Parameters

- **type** – *Type* of token to consume
- **error** – Error message if token type is not matched

Returns
Matched token

void **diagnostic**(std::string_view msg, *NssToken* token, bool is_warning = false)

Report diagnostic

Parameters
msg – Message to report

bool **is_end**() const

Checks if at end of token stream.

bool **match**(std::initializer_list<*NssTokenType*> types)

Checks if next token matches a particular type.

Note: Advances the token stream

Parameters
types – An initializer list of token types

Returns
True if there is a match

void **lex**()

Lexes the file.

NssToken **lookahead**(size_t index) const

Looks ahead in the token stream

Parameters
index – Index to look ahead to, from current token

NssToken **peek**() const

Next token in the token stream.

NssToken **previous**()

Previous token in the token stream.

void **synchronize**(bool allow_rbrace = false)

Advances token stream after an error.

Expression ***parse_expr**()

Expression ***parse_expr_assign**()

Expression ***parse_expr_conditional**()

Expression ***parse_expr_or**()

Expression ***parse_expr_and()**

Expression ***parse_expr_bitwise()**

Expression ***parse_expr_equality()**

Expression ***parse_expr_relational()**

Expression ***parse_expr_shift()**

Expression ***parse_expr_additive()**

Expression ***parse_expr_multiplicative()**

Expression ***parse_expr_unary()**

Expression ***parse_expr_postfix()**

Expression ***parse_expr_primary()**

Expression ***parse_expr_group()**

Statement ***parse_stmt()**

BlockStatement ***parse_stmt_block()**

DoStatement ***parse_stmt_do()**

ExprStatement ***parse_stmt_expr()**

IfStatement ***parse_stmt_if()**

ForStatement ***parse_stmt_for()**

LabelStatement ***parse_stmt_label()**

JumpStatement ***parse_stmt_jump()**

SwitchStatement ***parse_stmt_switch()**

WhileStatement ***parse_stmt_while()**

Type **parse_type()**

Statement ***parse_decl()**

StructDecl ***parse_decl_struct()**

Declaration ***parse_decl_function_def()**

FunctionDecl ***parse_decl_function()**

VarDecl ***parse_decl_param()**

Ast **parse_program()**

Parses script.

Public Members

Context *ctx_ = nullptr

Nss *parent_ = nullptr

std::string_view view_

Ast ast_

std::vector<*NssToken*> tokens

size_t current_ = 0

6.13.219 nw::script::NssToken

struct NssToken

Public Functions

NssToken() = default

inline NssToken(*NssTokenType* type_, std::string_view id_)

inline NssToken(*NssTokenType* type_, std::string_view id_, *SourcePosition* start, *SourcePosition* end)

Public Members

NssTokenType type = *NssTokenType*::*INVALID*

SourceLocation loc

6.13.220 nw::script::PostfixExpression

struct PostfixExpression : public nw::script::*Expression*

Public Functions

inline **PostfixExpression**(*Expression* *lhs_, *NssToken* token)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***lhs** = nullptr

NssToken **op**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.221 nw::script::SourceLocation

struct **SourceLocation**

Public Functions

inline size_t **length**() const noexcept

Gets the length of source code covered.

inline std::string_view **view**() const noexcept

Gets a view of the source code covered.

Public Members

const char *`start` = nullptr

Pointer to start of source code.

const char *`end` = nullptr

Pointer to end of source code.

SourceRange `range`

Source range.

6.13.222 nw::script::SourcePosition

struct `SourcePosition`

Position in source code.

Public Functions

bool `operator==`(const *SourcePosition* &rhs) const = default

auto `operator<=>`(const *SourcePosition* &rhs) const = default

Public Members

size_t `line` = 0

Starting line.

size_t `column` = 0

Starting column.

6.13.223 nw::script::SourceRange

struct `SourceRange`

Range of source code.

Public Members

SourcePosition `start`

Start of range.

SourcePosition `end`

End of Range.

6.13.224 nw::script::Statement

struct **Statement** : public nw::script::*AstNode*

Subclassed by *nw::script::BlockStatement*, *nw::script::Declaration*, *nw::script::DoStatement*, nw::script::EmptyStatement, *nw::script::ExprStatement*, *nw::script::ForStatement*, *nw::script::IfStatement*, *nw::script::JumpStatement*, *nw::script::LabelStatement*, *nw::script::SwitchStatement*, *nw::script::WhileStatement*

Public Functions

virtual **~Statement**() = default

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.225 nw::script::StructDecl

struct **StructDecl** : public nw::script::*Declaration*

Public Functions

inline virtual std::string **identifier**() const override

const *VarDecl* ***locate_member_decl**(std::string_view name) const

virtual *SourceRange* **range**() const noexcept

virtual *SourceRange* **selection_range**() const noexcept

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

std::vector<*Declaration**> **decls**

Type **type**

SourceRange **range_selection_**

std::string_view **view**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.226 nw::script::SwitchStatement

struct **SwitchStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***target**

Statement ***block** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.227 nw::script::Symbol

struct **Symbol**

Info regarding a particular symbol somewhere in a source file.

Public Members

AstNode ***node** = nullptr

AstNode if symbol is used in a variable expression.

const *Declaration* ***decl** = nullptr

Original declaration.

std::string **comment**

Comment on original declaration, in case of functions decl is preferred over definition.

std::string **type**

Type of the symbol.

SymbolKind **kind**

The kind of symbol.

const *Nss* ***provider** = nullptr

What script this symbol is from, i.e. "nwscript".

std::string_view **view**

View of declaration.

6.13.228 nw::script::Type

struct **Type**

Contains type tokens.

Public Functions

inline *SourcePosition* **range_start()** const noexcept

Public Members

NssToken **type_qualifier**

`const`

NssToken **type_specifier**

`int`, `float`, `string`, etc

NssToken **struct_id**

6.13.229 nw::script::UnaryExpression

struct **UnaryExpression** : public nw::script::*Expression*

Public Functions

inline **UnaryExpression**(*NssToken* token, *Expression* *rhs_)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

> **Note:** This function does not traverse dependencies

Public Members

NssToken **op**

Expression ***rhs** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.230 nw::script::VarDecl

struct **VarDecl** : public nw::script::*Declaration*

Public Functions

inline virtual std::string **identifier**() const override

virtual *SourceRange* **range**() const noexcept

virtual *SourceRange* **selection_range**() const noexcept

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **identifier_**

Expression ***init** = nullptr

Type **type**

SourceRange **range_selection_**

std::string_view **view**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.231 nw::script::VariableExpression

struct **VariableExpression** : public nw::script::*Expression*

Public Functions

inline explicit **VariableExpression**(*NssToken* token)

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

NssToken **var**

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.232 nw::script::WhileStatement

struct **WhileStatement** : public nw::script::*Statement*

Public Functions

virtual void **accept**(*BaseVisitor* *visitor) = 0

virtual void **complete**(const std::string &needle, std::vector<const *Declaration**> &out, bool no_filter = false) const

Find completions for this *Ast* Node

Note: This function does not traverse dependencies

Public Members

Expression ***check** = nullptr

Statement ***block** = nullptr

size_t **type_id_** = invalid_type_id

bool **is_const_** = false

immer::map<std::string, Export> **env_**

SourceRange **range_**

6.13.233 nwn1::Profile

struct **Profile** : public nw::*GameProfile*

NWN1 Game *Profile*.

Public Functions

virtual **~Profile**() = default

virtual bool **load_rules**() const override

Loads rules.

- Load Selector and Matcher
- Load Components
- Load 2DAs
- Load Constants
- Post Process 2DAs

virtual bool **load_resources**() override

Loads containers into resman.

6.14 defines

6.14.1 DEFINE_ENUM_FLAGS

DEFINE_ENUM_FLAGS(T)

Defines bitwise functions for an `enum` type.

6.14.2 ROLLNW_STRINGIFY

ROLLNW_STRINGIFY(a)

6.14.3 ROLLNW_UNUSED

ROLLNW_UNUSED(thing)

Silences unused variable warnings.

6.14.4 SCOPE_EXIT

SCOPE_EXIT(f)

Creates scope exit.

6.15 enums

6.15.1 nw::AlignmentAxis

enum class `nw::`**AlignmentAxis**

Values:

enumerator **neither**

enumerator **law_chaos**

enumerator **good_evil**

enumerator **both**

6.15.2 nw::AlignmentFlags

enum class `nw::AlignmentFlags`

Values:

enumerator **none**

enumerator **neutral**

enumerator **lawful**

enumerator **chaotic**

enumerator **good**

enumerator **evil**

6.15.3 nw::AlignmentType

enum class `nw::AlignmentType`

Values:

enumerator **all**

enumerator **neutral**

enumerator **lawful**

enumerator **chaotic**

enumerator **good**

enumerator **evil**

6.15.4 nw::AreaFlags

enum class `nw::AreaFlags` : uint32_t

Values:

enumerator **none**

Unspecified.

enumerator **interior**

Exterior if unset.

enumerator **underground**

Above ground if unset.

enumerator **natural**

Urban if unset.

6.15.5 nw::AttackResult

enum class `nw::AttackResult`

Values:

enumerator **hit_by_auto_success**

enumerator **hit_by_critical**

enumerator **hit_by_roll**

enumerator **miss_by_auto_fail**

enumerator **miss_by_concealment**

enumerator **miss_by_miss_chance**

enumerator **miss_by_roll**

6.15.6 nw::DialogNodeType

enum class `nw::DialogNodeType`

Values:

enumerator **entry**

enumerator **reply**

6.15.7 nw::DoorAnimationState

enum class `nw::DoorAnimationState` : uint8_t

Values:

enumerator **closed**

enumerator **opened1**

enumerator **opened2**

6.15.8 nw::EquipIndex

enum class `nw::EquipIndex` : uint32_t

Values:

enumerator **head**

enumerator **chest**

enumerator **boots**

enumerator **arms**

enumerator **righthand**

enumerator **lefthand**

enumerator **cloak**

enumerator **leftring**

enumerator **rightring**

enumerator **neck**

enumerator **belt**

enumerator **arrows**

enumerator **bullets**

enumerator **bolts**

enumerator **creature_left**

enumerator **creature_right**

enumerator **creature_bite**

enumerator **creature_skin**

enumerator **invalid**

6.15.9 nw::EquipSlot

enum class `nw::EquipSlot`

Values:

enumerator **head**

enumerator **chest**

enumerator **boots**

enumerator **arms**

enumerator **righthand**

enumerator **lefthand**

enumerator **cloak**

enumerator **leftring**

enumerator **rightring**

enumerator **neck**

enumerator **belt**

enumerator **arrows**

enumerator **bullets**

enumerator **bolts**

enumerator **creature_left**

enumerator **creature_right**

enumerator **creature_bite**

enumerator **creature_skin**

6.15.10 nw::ErfType

enum class `nw::ErfType`

Values:

enumerator **erf**

enumerator **hak**

enumerator **mod**

enumerator **sav**

6.15.11 nw::ErfVersion

enum class `nw::ErfVersion`

Values:

enumerator **v1_0**

enumerator **v1_1**

6.15.12 nw::GameVersion

enum class `nw::GameVersion`

Values:

enumerator **invalid**

enumerator **v1_69**

enumerator **vEE**

enumerator **nwn2**

6.15.13 nw::ItemModelType

enum class `nw::ItemModelType` : uint8_t

Values:

enumerator **simple**

enumerator **layered**

enumerator **composite**

enumerator **armor**

6.15.14 nw::LanguageID

enum class `nw::LanguageID` : uint32_t

Language IDs.

Values:

enumerator **invalid**

enumerator **english**

enumerator **french**

enumerator **german**

enumerator **italian**

enumerator **spanish**

enumerator **polish**

enumerator **korean**

Unsupported in EE?

enumerator **chinese_traditional**

Unsupported in EE?

enumerator **chinese_simplified**

Unsupported in EE?

enumerator **japanese**

Unsupported in EE?

6.15.15 nw::ObjectID

enum class `nw::ObjectID` : uint32_t

Opaque type.. for now.

Values:

6.15.16 nw::ObjectType

enum class `nw::ObjectType` : uint16_t

Object types.

Values:

enumerator **invalid**

enumerator **gui**

enumerator **tile**

enumerator **module**

enumerator **area**

enumerator **creature**

enumerator **item**

enumerator **trigger**

enumerator **projectile**

enumerator **placeable**

enumerator **door**

enumerator **areaofeffect**

enumerator **waypoint**

enumerator **encounter**

enumerator **store**

enumerator **portal**

enumerator **sound**

enumerator **player**

6.15.17 nw::PaletteNodeType

enum class `nw::PaletteNodeType`

Values:

enumerator **branch**

enumerator **category**

enumerator **blueprint**

6.15.18 nw::PlaceableAnimationState

enum class `nw::PlaceableAnimationState` : uint8_t

Values:

enumerator **none**

enumerator **open**

enumerator **closed**

enumerator **destroyed**

enumerator **activated**

enumerator **deactivated**

6.15.19 nw::PltLayer

enum `nw::PltLayer`

Plt formats respective layers.

Values:

enumerator **plt_layer_skin**

enumerator **plt_layer_hair**

enumerator **plt_layer_metal1**

enumerator **plt_layer_metal2**

enumerator **plt_layer_cloth1**

enumerator **plt_layer_cloth2**

enumerator **plt_layer_leather1**

enumerator **plt_layer_leather2**

enumerator **plt_layer_tattoo1**

enumerator **plt_layer_tattoo2**

enumerator **plt_layer_size**

6.15.20 nw::SelectorType

enum class `nw::SelectorType` : uint32_t

Selector types.

Values:

enumerator **ability**

Subtype: ability_* constant.

enumerator **ac**

Subtype: ac_* constant.

enumerator **alignment**

Subtype: AlignmentAxis.

enumerator **arcane_level**

Subtype: none.

enumerator **bab**

Subtype: none.

enumerator **caster_level**

Subtype:

enumerator **class_level**

Subtype: class_* constant.

enumerator **feat**

Subtype: feat_* constant.

enumerator **hitpoints_max**

Subtype: none.

enumerator **level**

Subtype: none.

enumerator **local_var_int**

Subtype: local var name, eg. "X1_AllowArcher".

enumerator **local_var_str**

Subtype: local var name, eg. "some_var".

enumerator **race**

Subtype: none.

enumerator **skill**

Subtype: skill_* constant.

enumerator **spell_level**

Subtype:

6.15.21 nw::SerializationProfile

enum class `nw::SerializationProfile`

Game serialization profiles.

Values:

enumerator **any**

enumerator **blueprint**

enumerator **instance**

enumerator **savegame**

6.15.22 nw::SpellFlags

enum class `nw::SpellFlags` : uint8_t

Values:

enumerator **none**

enumerator **readied**

enumerator **spontaneous**

enumerator **unlimited**

6.15.23 nw::SpellMetaMagic

enum class `nw::SpellMetaMagic` : uint8_t

Values:

enumerator **none**

enumerator **empower**

enumerator **extend**

enumerator **maximize**

enumerator **quicken**

enumerator **silent**

enumerator **still**

6.15.24 nw::TargetState

enum class `nw::TargetState`

Values:

enumerator **none**

enumerator **blind**

enumerator **attacker_invis**

enumerator **unseen**

enumerator **moving**

enumerator **prone**

enumerator **stunned**

enumerator **flanked**

enumerator **flatfooted**

enumerator **asleep**

enumerator **attacker_unseen**

enumerator **invis**

6.15.25 nw::script::DiagnosticLevel

Warning: doxygenenum: Cannot find enum "nw::script::DiagnosticLevel" in doxygen xml output for project "rollNW" from directory: build/xml/

6.15.26 nw::script::DiagnosticType

enum class `nw::script::DiagnosticType`

Values:

enumerator `lexical`

enumerator `parse`

enumerator `semantic`

6.15.27 nw::script::NssTokenType

enum class `nw::script::NssTokenType`

Values:

enumerator `INVALID`

enumerator `END`

enumerator `IDENTIFIER`

enumerator `COMMENT`

enumerator `LPAREN`

enumerator `RPAREN`

enumerator `LBRACE`

enumerator `RBRACE`

enumerator `LBRACKET`

enumerator `RBRACKET`

enumerator **COMMA**

enumerator **COLON**

enumerator **QUESTION**

enumerator **SEMICOLON**

enumerator **POUND**

enumerator **DOT**

enumerator **AND**

enumerator **ANDAND**

enumerator **ANDEQ**

enumerator **DIV**

enumerator **DIVEQ**

enumerator **EQ**

enumerator **EQEQ**

enumerator **GT**

enumerator **GTEQ**

enumerator **LT**

enumerator **LTEQ**

enumerator **MINUS**

enumerator **MINUSEQ**

enumerator **MINUSMINUS**

enumerator **MOD**

enumerator **MODEQ**

enumerator **TIMES**

enumerator **TIMESEQ**

enumerator **NOT**

enumerator **NOTEQ**

enumerator **OR**

enumerator **OREQ**

enumerator **OROR**

enumerator **PLUS**

enumerator **PLUSEQ**

enumerator **PLUSPLUS**

enumerator **SL**

enumerator **SLEQ**

enumerator **SR**

enumerator **SREQ**

enumerator **TILDE**

enumerator **USR**

enumerator **USREQ**

enumerator **XOR**

enumerator **XOREQ**

enumerator **FLOAT_CONST**

enumerator **INTEGER_CONST**

enumerator **OBJECT_INVALID_CONST**

enumerator **OBJECT_SELF_CONST**

enumerator **STRING_CONST**

enumerator **STRING_RAW_CONST**

enumerator **LOCATION_INVALID**

enumerator **JSON_CONST**

enumerator **ACTION**

enumerator **BREAK**

enumerator **CASE**

enumerator **CASSOWARY**

enumerator **CONST_**

enumerator **CONTINUE**

enumerator **DEFAULT**

enumerator **DO**

enumerator **EFFECT**

enumerator **ELSE**

enumerator **EVENT**

enumerator **FLOAT**

enumerator **FOR**

enumerator **IF**

enumerator **INT**

enumerator **ITEMPROPERTY**

enumerator **JSON**

enumerator **LOCATION**

enumerator **OBJECT**

enumerator **RETURN**

enumerator **STRING**

enumerator **STRUCT**

enumerator **SQLQUERY**

enumerator **SWITCH**

enumerator **TALENT**

enumerator **VECTOR**

enumerator **VOID_**

enumerator **WHILE**

6.15.28 nw::script::SymbolKind

enum class `nw::script::`**SymbolKind**

Values:

enumerator **variable**

enumerator **function**

enumerator **type**

enumerator **param**

enumerator **field**

6.16 functions

6.16.1 nw::alignment_axis_from_flags

AlignmentAxis `nw::alignment_axis_from_flags(AlignmentFlags flags)`

Gets alignment axis from alignment flags.

6.16.2 nw::always_false

template<typename **T**>
constexpr bool `nw::always_false()`

Always returns false for use with `static_assert`

Template Parameters
T – type is disregarded

6.16.3 nw::count_feats_in_range

int `nw::count_feats_in_range(const nw::Creature *obj, nw::Feat start, nw::Feat end)`

Counts the number of known feats in the range [start, end].

6.16.4 nw::create_unique_tmp_path

fs::path `nw::create_unique_tmp_path()`

Creates randomly named folder in tmp. Analguous to POSIX `mkdtemp`.

6.16.5 nw::decode_plt_color

std::array<uint8_t, 4> `nw::decode_plt_color(const Plt &plt, const PltColors &colors, uint32_t x, uint32_t y)`

Decodes PLT and user selected colors to RBGA.

6.16.6 nw::decompress

ByteArray `nw::decompress(std::span<const uint8_t> span, const char *magic)`

Decompress a NWN:EE compressed buffer.

Note: Doesn't support Zstd dictionaries, but the game doesn't either.. yet. Supporting that will likely lead to API change.

Parameters

- **span** – Compressed data
- **magic** – Magic 4 byte sequence, i.e. "NSYC"

Returns
ByteArray Uncompressed data, empty on error.

6.16.7 nw::documents_path

fs::path `nw::documents_path()`

Gets user's documents path.

6.16.8 nw::equip_index_to_string

constexpr std::string_view `nw::equip_index_to_string(`*EquipIndex* idx`)`

6.16.9 nw::equip_slot_to_index

constexpr *EquipIndex* `nw::equip_slot_to_index(`*EquipSlot* slot`)`

6.16.10 nw::expand_path

std::filesystem::path `nw::expand_path(`const std::filesystem::path &path`)`

Expands path with ~ and environment variables.

6.16.11 nw::find_first_effect_of

template<typename **It**>
It `nw::find_first_effect_of(`*It* begin, *It* end, nw::EffectType type, int subtype = -1`)`

Finds first effect of a given type.

Template Parameters

It – A forward iterator

Parameters

- **begin** – Beginning of an range of effects
- **end** – Beginning of an range of effects
- **type** – An effect_type_* constant
- **subtype** – An effect subtype

Returns

It iterator to the first effect, or `end`

6.16.12 nw::from_base64

ByteArray `nw::from_base64(`const std::string &string`)`

Converts base64 string to an array of bytes.

6.16.13 nw::from_json

void `nw::from_json`(const nlohmann::json &json, *ByteArray* &ba)

Warning: doxygenfunction: Unable to resolve function "from_json" with arguments (const nlohmann::json&, Dialog&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile,␣
↪ObjectType object_type)
- void from_json(const nlohmann::json &j, Ability &type)
- void from_json(const nlohmann::json &j, ArmorClass &type)
- void from_json(const nlohmann::json &j, AttackType &type)
- void from_json(const nlohmann::json &j, BaseItem &type)
- void from_json(const nlohmann::json &j, Class &type)
- void from_json(const nlohmann::json &j, CombatMode &type)
- void from_json(const nlohmann::json &j, Damage &type)
- void from_json(const nlohmann::json &j, DamageModType &type)
- void from_json(const nlohmann::json &j, Disease &type)
- void from_json(const nlohmann::json &j, EffectType &type)
- void from_json(const nlohmann::json &j, Feat &type)
- void from_json(const nlohmann::json &j, ItemPropertyType &type)
- void from_json(const nlohmann::json &j, LocString &loc)
- void from_json(const nlohmann::json &j, MasterFeat &type)
- void from_json(const nlohmann::json &j, MissChanceType &type)
- void from_json(const nlohmann::json &j, ModifierType &type)
- void from_json(const nlohmann::json &j, ObjectID &id)
- void from_json(const nlohmann::json &j, ObjectType &type)
- void from_json(const nlohmann::json &j, Poison &type)
- void from_json(const nlohmann::json &j, Race &type)
- void from_json(const nlohmann::json &j, Resource &r)
- void from_json(const nlohmann::json &j, Resref &r)
- void from_json(const nlohmann::json &j, Save &type)
```

```
- void from_json(const nlohmann::json &j, SaveVersus &type)
- void from_json(const nlohmann::json &j, Situation &type)
- void from_json(const nlohmann::json &j, Skill &type)
- void from_json(const nlohmann::json &j, SpecialAttack &type)
- void from_json(const nlohmann::json &j, Spell &type)
- void from_json(const nlohmann::json &j, SpellEntry &spell)
- void from_json(const nlohmann::json &j, SpellSchool &type)
- void from_json(const nlohmann::json &json, ByteArray &ba)
- void from_json(const nlohmann::json &json, LevelUp &entry)
- void from_json(const nlohmann::json &json, Location &loc)
- void from_json(const nlohmann::json &json, Saves &saves)
```

Warning: doxygenfunction: Unable to resolve function "from_json" with arguments (const nlohmann::json&, DialogNode&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile,␣
↪ObjectType object_type)
- void from_json(const nlohmann::json &j, Ability &type)
- void from_json(const nlohmann::json &j, ArmorClass &type)
- void from_json(const nlohmann::json &j, AttackType &type)
- void from_json(const nlohmann::json &j, BaseItem &type)
- void from_json(const nlohmann::json &j, Class &type)
- void from_json(const nlohmann::json &j, CombatMode &type)
- void from_json(const nlohmann::json &j, Damage &type)
- void from_json(const nlohmann::json &j, DamageModType &type)
- void from_json(const nlohmann::json &j, Disease &type)
- void from_json(const nlohmann::json &j, EffectType &type)
- void from_json(const nlohmann::json &j, Feat &type)
- void from_json(const nlohmann::json &j, ItemPropertyType &type)
- void from_json(const nlohmann::json &j, LocString &loc)
```

```
- void from_json(const nlohmann::json &j, MasterFeat &type)
- void from_json(const nlohmann::json &j, MissChanceType &type)
- void from_json(const nlohmann::json &j, ModifierType &type)
- void from_json(const nlohmann::json &j, ObjectID &id)
- void from_json(const nlohmann::json &j, ObjectType &type)
- void from_json(const nlohmann::json &j, Poison &type)
- void from_json(const nlohmann::json &j, Race &type)
- void from_json(const nlohmann::json &j, Resource &r)
- void from_json(const nlohmann::json &j, Resref &r)
- void from_json(const nlohmann::json &j, Save &type)
- void from_json(const nlohmann::json &j, SaveVersus &type)
- void from_json(const nlohmann::json &j, Situation &type)
- void from_json(const nlohmann::json &j, Skill &type)
- void from_json(const nlohmann::json &j, SpecialAttack &type)
- void from_json(const nlohmann::json &j, Spell &type)
- void from_json(const nlohmann::json &j, SpellEntry &spell)
- void from_json(const nlohmann::json &j, SpellSchool &type)
- void from_json(const nlohmann::json &json, ByteArray &ba)
- void from_json(const nlohmann::json &json, LevelUp &entry)
- void from_json(const nlohmann::json &json, Location &loc)
- void from_json(const nlohmann::json &json, Saves &saves)
```

Warning: doxygenfunction: Unable to resolve function "from_json" with arguments (const nlohmann::json&, DialogPtr&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile)
- bool from_json(const nlohmann::json &archive, SerializationProfile profile,␣
↪ObjectType object_type)
- void from_json(const nlohmann::json &j, Ability &type)
- void from_json(const nlohmann::json &j, ArmorClass &type)
- void from_json(const nlohmann::json &j, AttackType &type)
```

```
- void from_json(const nlohmann::json &j, BaseItem &type)
- void from_json(const nlohmann::json &j, Class &type)
- void from_json(const nlohmann::json &j, CombatMode &type)
- void from_json(const nlohmann::json &j, Damage &type)
- void from_json(const nlohmann::json &j, DamageModType &type)
- void from_json(const nlohmann::json &j, Disease &type)
- void from_json(const nlohmann::json &j, EffectType &type)
- void from_json(const nlohmann::json &j, Feat &type)
- void from_json(const nlohmann::json &j, ItemPropertyType &type)
- void from_json(const nlohmann::json &j, LocString &loc)
- void from_json(const nlohmann::json &j, MasterFeat &type)
- void from_json(const nlohmann::json &j, MissChanceType &type)
- void from_json(const nlohmann::json &j, ModifierType &type)
- void from_json(const nlohmann::json &j, ObjectID &id)
- void from_json(const nlohmann::json &j, ObjectType &type)
- void from_json(const nlohmann::json &j, Poison &type)
- void from_json(const nlohmann::json &j, Race &type)
- void from_json(const nlohmann::json &j, Resource &r)
- void from_json(const nlohmann::json &j, Resref &r)
- void from_json(const nlohmann::json &j, Save &type)
- void from_json(const nlohmann::json &j, SaveVersus &type)
- void from_json(const nlohmann::json &j, Situation &type)
- void from_json(const nlohmann::json &j, Skill &type)
- void from_json(const nlohmann::json &j, SpecialAttack &type)
- void from_json(const nlohmann::json &j, Spell &type)
- void from_json(const nlohmann::json &j, SpellEntry &spell)
- void from_json(const nlohmann::json &j, SpellSchool &type)
- void from_json(const nlohmann::json &json, ByteArray &ba)
- void from_json(const nlohmann::json &json, LevelUp &entry)
- void from_json(const nlohmann::json &json, Location &loc)
- void from_json(const nlohmann::json &json, Saves &saves)
```

void **nw::from_json**(const nlohmann::json &json, *Location* &loc)

void **nw::from_json**(const nlohmann::json &j, *LocString* &loc)

void **nw::from_json**(const nlohmann::json &j, *ObjectID* &id)

nlohmann::json specialization

void **nw::from_json**(const nlohmann::json &j, *ObjectType* &type)

nlohmann::json specialization

void **nw::from_json**(const nlohmann::json &j, *Resource* &r)

nlohmann::json specialization

void **nw::from_json**(const nlohmann::json &j, *Resref* &r)

nlohmann::json specialization

void **nw::from_json**(const nlohmann::json &json, *Saves* &saves)

void **nw::from_json**(const nlohmann::json &j, *SpellEntry* &spell)

6.16.14 nw::from_utf8

std::string **nw::from_utf8**(std::string_view str, std::string_view encoding, bool ignore_errors = false)

Convert from utf8.

6.16.15 nw::from_utf8_by_global_lang

std::string **nw::from_utf8_by_global_lang**(std::string_view str, bool ignore_errors)

6.16.16 nw::from_utf8_by_langid

std::string **nw::from_utf8_by_langid**(std::string_view str, *LanguageID* id, bool ignore_errors)

6.16.17 nw::get_all_available_feats

std::vector<nw::*Feat*> **nw::get_all_available_feats**(const nw::*Creature* *obj)

Gets all feats for which requirements are met

Note: This is not yet very useful until a level up parameter is added.

6.16.18 nw::gff_to_gffjson

nlohmann::json **nw::gff_to_gffjson**(const *Gff* &gff)

Convert a *Gff* to JSON (nwn-lib/neverwinter.nim format, I think.)

6.16.19 nw::has_effect_applied

bool **nw::has_effect_applied**(nw::ObjectBase *obj, nw::EffectType type, int subtype = -1)

Determines if an effect type is applied to an object.

6.16.20 nw::has_feat_successor

std::pair<nw::*Feat*, int> **nw::has_feat_successor**(const nw::*Creature* *obj, nw::*Feat* feat)

Gets the highest known successor feat.

6.16.21 nw::highest_feat_in_range

nw::*Feat* `nw::highest_feat_in_range`(const nw::*Creature* *obj, nw::*Feat* start, nw::*Feat* end)

Gets the highest known feat in range [start, end].

6.16.22 nw::home_path

fs::path `nw::home_path`()

Gets user's home path.

6.16.23 nw::init_logger

void `nw::init_logger`(int argc, char *argv[])

Initialize logger.

6.16.24 nw::is_attack_type_hit

constexpr bool `nw::is_attack_type_hit`(*AttackResult* value)

6.16.25 nw::is_attack_type_miss

constexpr bool `nw::is_attack_type_miss`(*AttackResult* value)

6.16.26 nw::istream_read

template<typename **T**, typename **U**>
std::istream &`nw::istream_read`(std::istream &stream, *T* *data, *U* size)

Reads from a stream into an arbitrary pointer of type `T`

6.16.27 nw::itemprop_to_string

std::string `nw::itemprop_to_string`(const nw::*ItemProperty* &ip)

Converts item property to in-game style string.

6.16.28 nw::kernel::config

Config &`nw::kernel::config`()

Gets configuration options.

6.16.29 nw::kernel::load_module

Module *nw::kernel::**load_module**(const std::filesystem::path &path, std::string_view manifest = {})

Loads a module.

6.16.30 nw::kernel::max_modifier

template<typename **T**>
T nw::kernel::**max_modifier**(const ObjectBase *obj, const *ModifierType* type)

Maxes all modifiers of `type`

Template Parameters
T –

---

template<typename **T**>
T nw::kernel::**max_modifier**(const ObjectBase *obj, const *ModifierType* type, const ObjectBase *versus)

Maxes all modifiers of `type` versus an object.

Template Parameters
T –

---

template<typename **T**, typename **SubType**>
T nw::kernel::**max_modifier**(const ObjectBase *obj, const *ModifierType* type, *SubType* subtype)

Maxes all modifiers of a `type` and `subtype`

Template Parameters

- **T** –
- **U** – is some rule subtype

---

template<typename **T**, typename **SubType**>
T nw::kernel::**max_modifier**(const ObjectBase *obj, const *ModifierType* type, *SubType* subtype, const ObjectBase *versus)

Maxes all modifiers of a `type` and `subtype` versus another object.

Template Parameters

- **T** –
- **U** – is some rule subtype

6.16.31 nw::kernel::objects

inline *ObjectSystem* &nw::kernel::**objects**()

6.16.32 nw::kernel::resman

inline *Resources* &nw::kernel::**resman**()

6.16.33 nw::kernel::resolve_master_feat

template<typename **T**, typename **U**>
T nw::kernel::**resolve_master_feat**(const *Creature* *obj, *U* type, *MasterFeat* mfeat)

Resolves a master feat bonus.

Template Parameters

- **T** – Return type
- **U** – Rule type

Parameters

- **obj** – *Creature* object
- **type** – Rule value
- **mfeat** – Master feat

6.16.34 nw::kernel::resolve_master_feats

template<typename **T**, typename **U**, typename **Callback**, typename ...**Args**>
void nw::kernel::**resolve_master_feats**(const *Creature* *obj, *U* type, *Callback* cb, *Args*... mfeats)

Resolves an arbitrary number of master feats.

Template Parameters

- **T** – Return type
- **U** – Rule type
- **Callback** – Callback type should be void(T)
- **Args** – *MasterFeat*...

Parameters

- **obj** – *Creature* object
- **type** – Rule value
- **cb** – This parameter will be called with any valid master feat bonus as a parameter.
- **mfeats** – As many master feats as needed

6.16.35 nw::kernel::resolve_modifier

template<typename **Callback**>
bool `nw::kernel::`**`resolve_modifier`**(const ObjectBase *obj, const *Modifier* &mod, *Callback* cb, const ObjectBase *versus = nullptr, int32_t subtype = -1)

Calculates a modifier.

Template Parameters
Callback – *Modifier* callback function

---

template<typename **Callback**>
bool `nw::kernel::`**`resolve_modifier`**(const ObjectBase *obj, const *ModifierType* type, *Callback* cb)

Calculates all modifiers of `type`

Template Parameters
Callback – *Modifier* callback function

---

template<typename **Callback**>
bool `nw::kernel::`**`resolve_modifier`**(const ObjectBase *obj, const *ModifierType* type, const ObjectBase *versus, *Callback* cb)

Calculates all modifiers of `type` versus an object.

Template Parameters
Callback – *Modifier* callback function

---

template<typename **SubType**, typename **Callback**>
bool `nw::kernel::`**`resolve_modifier`**(const ObjectBase *obj, const *ModifierType* type, *SubType* subtype, *Callback* cb)

Calculates all modifiers of a `type` and `subtype`

Template Parameters

- **U** – is some rule subtype
- **Callback** – *Modifier* callback function

---

template<typename **SubType**, typename **Callback**>
bool `nw::kernel::`**`resolve_modifier`**(const ObjectBase *obj, const *ModifierType* type, *SubType* subtype, const ObjectBase *versus, *Callback* cb)

Calculates all modifiers of a `type` and `subtype` versus another object.

Template Parameters

- **U** – is some rule subtype
- **Callback** – *Modifier* callback function

6.16.36 nw::kernel::rules

inline *Rules* &nw::kernel::**rules**()

6.16.37 nw::kernel::serial_id_to_obj_type

inline *ObjectType* nw::kernel::**serial_id_to_obj_type**(std::string_view id)

6.16.38 nw::kernel::services

Services &nw::kernel::**services**()

Gets services.

6.16.39 nw::kernel::strings

inline *Strings* &nw::kernel::**strings**()

6.16.40 nw::kernel::sum_master_feats

template<typename **T**, typename **U**, typename ...**MasterFeats**>
T nw::kernel::**sum_master_feats**(const *Creature* *obj, *U* type, *MasterFeats*... mfeats)

Sum master feat bonus.

Template Parameters

- **T** – Return type
- **U** – Rule type
- **Args** – *MasterFeat*...

Parameters

- **obj** – *Creature* object
- **type** – Rule value
- **mfeats** – *MasterFeat*s

6.16.41 nw::kernel::sum_modifier

template<typename **T**>
T nw::kernel::**sum_modifier**(const ObjectBase *obj, const *ModifierType* type)

Sums all modifiers of type

Template Parameters

T –

Warning: doxygenfunction: Unable to resolve function "sum_modifier" with arguments (const ObjectBase*, const ModifierType, const ObjectBase*) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- template<typename T, typename SubType> T sum_modifier(const ObjectBase *obj, const␣
↪ModifierType type, SubType subtype)
- template<typename T, typename SubType> T sum_modifier(const ObjectBase *obj, const␣
↪ObjectBase *versus, const ModifierType type, SubType subtype)
- template<typename T> T sum_modifier(const ObjectBase *obj, const ModifierType type)
- template<typename T> T sum_modifier(const ObjectBase *obj, const ObjectBase *versus,
↪ const ModifierType type)
```

---

template<typename **T**, typename **SubType**>
T `nw::kernel::`**`sum_modifier`**(const ObjectBase *obj, const *ModifierType* type, *SubType* subtype)

Sums all modifiers of a `type` and `subtype`

Template Parameters

- **T** –
- **U** – is some rule subtype

Warning: doxygenfunction: Unable to resolve function "sum_modifier" with arguments (const ObjectBase*, const ModifierType, SubType, const ObjectBase*) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- template<typename T, typename SubType> T sum_modifier(const ObjectBase *obj, const␣
↪ModifierType type, SubType subtype)
- template<typename T, typename SubType> T sum_modifier(const ObjectBase *obj, const␣
↪ObjectBase *versus, const ModifierType type, SubType subtype)
- template<typename T> T sum_modifier(const ObjectBase *obj, const ModifierType type)
- template<typename T> T sum_modifier(const ObjectBase *obj, const ObjectBase *versus,
↪ const ModifierType type)
```

6.16.42 nw::kernel::unload_module

void `nw::kernel::`**`unload_module`**()

Unloads currently active module.

6.16.43 nw::knows_feat

bool `nw::`**`knows_feat`**(const nw::*Creature* *obj, nw::*Feat* feat)

Checks if an entity knows a given feat.

6.16.44 nw::max_effects_of

template<typename **T**, typename **It**, typename **Extractor** = decltype(&effect_extract_int0)>
std::pair<*T*, *It*> `nw::max_effects_of`(*It* begin, *It* end, nw::EffectType type, int subtype, *Versus* vs = {}, *Extractor* extractor = &effect_extract_int0) noexcept

Finds all applicable effects of a given type / subtype.

Applicable effects are passed to a user supplied callback.

Template Parameters

- **T** – An arbitrary type that can be held in an effect, e.g. a simple integer, a damage roll, etc.
- **It** – An iterator type
- **Extractor** – A function that extracts a value of type `T` from an *EffectHandle*

Parameters

- **begin** – Start of a range of effect handles of a type/subtype
- **end** – End range of effect handles
- **type** – An effect_type_* constant
- **subtype** – A effect subtype, if no subtype -1 should be passed
- **vs** – *Versus* struct
- **extractor** – A function that extracts the value from a particular effect.

Returns
(result, iterator)

6.16.45 nw::move_file_safely

bool `nw::move_file_safely`(const std::filesystem::path &from, const std::filesystem::path &to)

Copies and deletes a file to a new location, overwrites existing.

6.16.46 nw::needs_quote

inline bool `nw::needs_quote`(std::string_view str)

6.16.47 nw::operator==

bool `nw::operator==`(const *DiceRoll* &lhs, const *DiceRoll* &rhs)

inline bool `nw::operator==`(const *Null*&, const *Null*&)

inline bool `nw::operator==`(const *Resource* &lhs, const *Resource* &rhs)

inline bool `nw::operator==`(const *Resref* &lhs, const *Resref* &rhs)

Warning: doxygenfunction: Unable to resolve function "operator==" with arguments (const Selector&, const Selector&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- bool operator==(const Ability &rhs) const = default
- bool operator==(const ArmorClass &rhs) const = default
- bool operator==(const AttackType &rhs) const = default
- bool operator==(const BaseItem &rhs) const = default
- bool operator==(const ByteArray &other) const
- bool operator==(const Class &rhs) const = default
- bool operator==(const CombatMode &rhs) const = default
- bool operator==(const Damage &rhs) const = default
- bool operator==(const DamageModType &rhs) const = default
- bool operator==(const DiceRoll &lhs, const DiceRoll &rhs)
- bool operator==(const Disease &rhs) const = default
- bool operator==(const EffectHandle&) const = default
- bool operator==(const EffectType &rhs) const = default
- bool operator==(const EventHandle &rhs) const
- bool operator==(const Feat &rhs) const = default
- bool operator==(const InternedString &rhs) const noexcept = default
- bool operator==(const ItemPropertyType &rhs) const = default
- bool operator==(const LocString &other) const
- bool operator==(const Location&) const = default
- bool operator==(const MasterFeat &rhs) const = default
- bool operator==(const MasterFeatEntry &lhs, const MasterFeatEntry &rhs)
- bool operator==(const MissChanceType &rhs) const = default
- bool operator==(const ModifierType &rhs) const = default
- bool operator==(const Null&, const Null&)
- bool operator==(const Poison &rhs) const = default
- bool operator==(const Race &rhs) const = default
- bool operator==(const Resource &lhs, const Resource &rhs)
- bool operator==(const ResourceData &other) const = default
- bool operator==(const Resref &lhs, const Resref &rhs)
- bool operator==(const Save &rhs) const = default
- bool operator==(const SaveVersus &rhs) const = default
- bool operator==(const Situation &rhs) const = default
- bool operator==(const Skill &rhs) const = default
- bool operator==(const SourcePosition &rhs) const = default
- bool operator==(const SpecialAttack &rhs) const = default
- bool operator==(const Spell &rhs) const = default
- bool operator==(const SpellEntry&) const = default
- bool operator==(const SpellSchool &rhs) const = default
- bool operator==(const Variant &rhs) const noexcept
- bool operator==(const Versus &rhs) const = default
- bool operator==(const WeaponModifier &lhs, const WeaponModifier &rhs)
```

6.16.48 nw::operator<<

std::ostream &nw::**operator<<**(std::ostream &out, const *Resource* &res)

std::ostream &nw::**operator<<**(std::ostream &out, const *Resref* &resref)

Warning: doxygenfunction: Unable to resolve function "operator<<" with arguments (std::ostream&, const TwoDA&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- std::ostream &operator<<(std::ostream &out, const Resource &res)
- std::ostream &operator<<(std::ostream &out, const Resref &resref)
- std::ostream &operator<<(std::ostream &out, const TwoDA &tda)
- std::ostream &operator<<(std::ostream &out, const TwoDA &tda)
- std::ostream &operator<<(std::ostream &out, const nw::script::NssToken &token)
```

inline std::ostream &**operator<<**(std::ostream &out, const nw::script::*NssToken* &token)

6.16.49 nw::operator<

bool nw::**operator<**(const *DiceRoll* &lhs, const *DiceRoll* &rhs)

inline bool nw::**operator<**(const *Null*&, const *Null*&)

inline bool nw::**operator<**(const *Resource* &lhs, const *Resource* &rhs)

inline bool nw::**operator<**(const *Resref* &lhs, const *Resref* &rhs)

Warning: doxygenfunction: Unable to resolve function "operator<" with arguments (const Selector&, const Selector&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- bool operator<(const DiceRoll &lhs, const DiceRoll &rhs)
- bool operator<(const EventHandle &rhs) const
- bool operator<(const MasterFeatEntry &lhs, const MasterFeatEntry &rhs)
- bool operator<(const Modifier &lhs, const Modifier &rhs)
- bool operator<(const Null&, const Null&)
- bool operator<(const Resource &lhs, const Resource &rhs)
- bool operator<(const Resref &lhs, const Resref &rhs)
- bool operator<(const Variant &rhs) const noexcept
- bool operator<(const WeaponModifier &lhs, const WeaponModifier &rhs)
```

6.16.50 nw::ostream_write

template<typename **T**, typename **U**>
std::ostream &nw::**ostream_write**(std::ostream &stream, const *T* *data, *U* size)

Writes to a stream from nto an arbitrary pointer of type T

6.16.51 nw::probe_nwn_install

InstallInfo `nw::probe_nwn_install(`*GameVersion* version`)`

Probes for an NWN install

Parameters

only – probe for specific version

6.16.52 nw::resolve_effects_of

template<typename **T**, typename **It**, typename **CallBack**, typename **Extractor**, typename **Comp** = std::greater<*T*>>
It `nw::resolve_effects_of(`*It* begin, *It* end, nw::EffectType type, int subtype, *Versus* vs, *CallBack* cb, *Extractor* extractor, *Comp* comparator = std::greater<*T*>{}`)` noexcept

Finds all applicable effects of a given type / subtype.

Applicable effects are passed to a user supplied callback.

Template Parameters

- **T** – An arbitrary type that can be held in an effect, e.g. a simple integer, a damage roll, etc.
- **It** – An iterator type
- **CallBack** – A function with the signature void(T) supplied by the user
- **Extractor** – A function that extracts a value of type `T` from an *EffectHandle*
- **Comp** – A comparator taking two `T` values and returns `true` if the first is greater (Default std::greater<T>)

Parameters

- **begin** – Start of a range of effect handles of a type/subtype
- **end** – End range of effect handles
- **type** – An effect_type_* constant
- **subtype** – A effect subtype, if no subtype -1 should be passed
- **vs** – *Versus* struct
- **cb** – A user defined callback that will be passed an applicable effect's value.
- **extractor** – A function that extracts the value from a particular effect.
- **comparator** – A function taking two `T` values and returns `true` if the first is greater (Default std::greater<T>)

Returns

iterator to passed last processed effect

6.16.53 nw::reverse

template<typename **T**>
reversion_wrapper<*T*> `nw::reverse`(*T* &&iterable)

Creates a reverse iterator for range-for loops.

6.16.54 nw::roll_dice

int `nw::roll_dice`(*DiceRoll* roll, int multiplier)

Rolls a set of dice

Parameters

- **roll** – Dice to roll
- **multiplier** – Roll dice n times

6.16.55 nw::roll_dice_explode

int `nw::roll_dice_explode`(*DiceRoll* dice, int on, int limit)

Rolls a set exploding of dice

Parameters

- **dice** – Dice to roll
- **on** – Value to explode on, default is the sides of the dice
- **limit** – Limit of the number of explosions, default limit is 20

6.16.56 nw::string::desanitize_colors

std::string `nw::string::desanitize_colors`(std::string str)

Converts color hex to bytes <cXXXXXX> -> <c\x\x\x>. Note: MOVE in the string.

6.16.57 nw::string::endswith

bool `nw::string::endswith`(std::string_view str, std::string_view suffix)

Determines if a string ends with a given suffix.

6.16.58 nw::string::from

template<typename **T**>
std::optional<*T*> `nw::string::from`(std::string_view str) = delete

String conversions to integral and floating pointing types.

Note: Even those tho this function is deleted, see Template Parameters for specilized versions.

Template Parameters

T – `bool`, `int32_t`, `uint32_t`, `int64_t`, `uint64_t`, `float`, `double`

Parameters
str – Input string

Returns
std::optional<T>

6.16.59 nw::string::glob_to_regex

std::regex `nw::string::glob_to_regex`(std::string_view pattern, bool icase = false)

Converts a glob pattern to a regex.

Note: This only supports ?, *, and [seq]

Parameters

- **pattern** – E.g, "file?_nam*.ext"
- **icase** – If true returns a case insentive regex

Returns
std::regex

6.16.60 nw::string::icmp

bool `nw::string::icmp`(std::string_view first, std::string_view second)

Case insenstive comparison.

6.16.61 nw::string::join

std::string `nw::string::join`(const std::vector<std::string> &strings, const char *delim = " ")

Joins a vector of strings.

Parameters

- **strings** – Vector of strings.
- **delim** – Separator. Default " "

Returns
std::string

6.16.62 nw::string::ltrim_in_place

std::string *`nw::string::ltrim_in_place`(std::string *str)

Trims left in place.

6.16.63 nw::string::rtrim_in_place

std::string *`nw::string::rtrim_in_place`(std::string *str)

Trims right in place.

6.16.64 nw::string::sanitize_colors

std::string `nw::string::sanitize_colors`(std::string str)

Converts color bytes to hex <c\x\x\x> -> <cXXXXXX>. Note: MOVE in the string.

6.16.65 nw::string::split

std::vector<std::string> `nw::string::split`(const std::string &str, char delim, bool skipEmpty = true, bool trimmed = true)

Splits a string into an vector of strings.

Parameters

- **str** – String to split
- **delim** – Delimiter
- **skipEmpty** – Ignore empty strings
- **trimmed** – Trim strings after split

Returns

std::vector<std::string>

6.16.66 nw::string::startswith

bool `nw::string::startswith`(std::string_view str, std::string_view prefix)

Determines if a string starts with a given prefix.

6.16.67 nw::string::tolower

void `nw::string::tolower`(std::string *str)

Converts string to lowercase, in place.

6.16.68 nw::string::trim_in_place

std::string *`nw::string::trim_in_place`(std::string *str)

Trims string in place.

6.16.69 nw::sum_effects_of

template<typename **T**, typename **It**, typename **Extractor** = decltype(&effect_extract_int0), typename **Comp** = std::greater<*T*>>
std::pair<*T*, *It*> `nw::sum_effects_of`(*It* begin, *It* end, nw::EffectType type, int subtype, *Versus* vs = {}, *Extractor* extractor = &effect_extract_int0, *Comp* comparator = std::greater<*T*>{}) noexcept

Finds all applicable effects of a given type / subtype.

Applicable effects are passed to a user supplied callback.

Template Parameters

- **T** – An arbitrary type that can be held in an effect, e.g. a simple integer, a damage roll, etc.
- **It** – An iterator type
- **Extractor** – A function that extracts a value of type `T` from an *EffectHandle*
- **Comp** – A comparator taking two `T` values and returns `true` if the first is greater (Default std::greater<T>)

Parameters

- **begin** – Start of a range of effect handles of a type/subtype
- **end** – End range of effect handles
- **type** – An effect_type_* constant
- **subtype** – A effect subtype, if no subtype -1 should be passed
- **vs** – *Versus* struct
- **extractor** – A function that extracts the value from a particular effect.
- **comparator** – A function taking two `T` values and returns `true` if the first is greater (Default std::greater<T>)

Returns
(result, iterator)

6.16.70 nw::to_base64

std::string `nw::to_base64`(std::span<const uint8_t> bytes)

Converts span of bytes to a base64 string.

6.16.71 nw::to_bool

template<typename **T**>
constexpr bool `to_bool`(const *T* thing)

Converts enum flag to boolean.

6.16.72 nw::to_json

void **nw::to_json**(nlohmann::json &json, const *ByteArray* &ba)

Warning: doxygenfunction: Unable to resolve function "to_json" with arguments (nlohmann::json&, const Dialog&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile, ObjectType object_type) const
- nlohmann::json to_json(nw::ResourceType::type restype) const
- void to_json(nlohmann::json &j, ObjectID id)
- void to_json(nlohmann::json &j, ObjectType type)
- void to_json(nlohmann::json &j, const Ability &type)
- void to_json(nlohmann::json &j, const ArmorClass &type)
- void to_json(nlohmann::json &j, const AttackType &type)
- void to_json(nlohmann::json &j, const BaseItem &type)
- void to_json(nlohmann::json &j, const Class &type)
- void to_json(nlohmann::json &j, const CombatMode &type)
- void to_json(nlohmann::json &j, const Damage &type)
- void to_json(nlohmann::json &j, const DamageModType &type)
- void to_json(nlohmann::json &j, const Disease &type)
- void to_json(nlohmann::json &j, const EffectType &type)
- void to_json(nlohmann::json &j, const Feat &type)
- void to_json(nlohmann::json &j, const ItemPropertyType &type)
- void to_json(nlohmann::json &j, const LocString &loc)
- void to_json(nlohmann::json &j, const MasterFeat &type)
- void to_json(nlohmann::json &j, const MissChanceType &type)
- void to_json(nlohmann::json &j, const ModifierType &type)
- void to_json(nlohmann::json &j, const Poison &type)
- void to_json(nlohmann::json &j, const Race &type)
- void to_json(nlohmann::json &j, const Resource &r)
- void to_json(nlohmann::json &j, const Resref &r)
```

```
- void to_json(nlohmann::json &j, const Save &type)
- void to_json(nlohmann::json &j, const SaveVersus &type)
- void to_json(nlohmann::json &j, const Situation &type)
- void to_json(nlohmann::json &j, const Skill &type)
- void to_json(nlohmann::json &j, const SpecialAttack &type)
- void to_json(nlohmann::json &j, const Spell &type)
- void to_json(nlohmann::json &j, const SpellEntry &spell)
- void to_json(nlohmann::json &j, const SpellSchool &type)
- void to_json(nlohmann::json &json, const ByteArray &ba)
- void to_json(nlohmann::json &json, const LevelUp &entry)
- void to_json(nlohmann::json &json, const Location &loc)
- void to_json(nlohmann::json &json, const Saves &saves)
```

Warning: doxygenfunction: Unable to resolve function "to_json" with arguments (nlohmann::json&, const DialogNode&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile, ObjectType object_type) const
- nlohmann::json to_json(nw::ResourceType::type restype) const
- void to_json(nlohmann::json &j, ObjectID id)
- void to_json(nlohmann::json &j, ObjectType type)
- void to_json(nlohmann::json &j, const Ability &type)
- void to_json(nlohmann::json &j, const ArmorClass &type)
- void to_json(nlohmann::json &j, const AttackType &type)
- void to_json(nlohmann::json &j, const BaseItem &type)
- void to_json(nlohmann::json &j, const Class &type)
- void to_json(nlohmann::json &j, const CombatMode &type)
- void to_json(nlohmann::json &j, const Damage &type)
- void to_json(nlohmann::json &j, const DamageModType &type)
- void to_json(nlohmann::json &j, const Disease &type)
```

```
- void to_json(nlohmann::json &j, const EffectType &type)
- void to_json(nlohmann::json &j, const Feat &type)
- void to_json(nlohmann::json &j, const ItemPropertyType &type)
- void to_json(nlohmann::json &j, const LocString &loc)
- void to_json(nlohmann::json &j, const MasterFeat &type)
- void to_json(nlohmann::json &j, const MissChanceType &type)
- void to_json(nlohmann::json &j, const ModifierType &type)
- void to_json(nlohmann::json &j, const Poison &type)
- void to_json(nlohmann::json &j, const Race &type)
- void to_json(nlohmann::json &j, const Resource &r)
- void to_json(nlohmann::json &j, const Resref &r)
- void to_json(nlohmann::json &j, const Save &type)
- void to_json(nlohmann::json &j, const SaveVersus &type)
- void to_json(nlohmann::json &j, const Situation &type)
- void to_json(nlohmann::json &j, const Skill &type)
- void to_json(nlohmann::json &j, const SpecialAttack &type)
- void to_json(nlohmann::json &j, const Spell &type)
- void to_json(nlohmann::json &j, const SpellEntry &spell)
- void to_json(nlohmann::json &j, const SpellSchool &type)
- void to_json(nlohmann::json &json, const ByteArray &ba)
- void to_json(nlohmann::json &json, const LevelUp &entry)
- void to_json(nlohmann::json &json, const Location &loc)
- void to_json(nlohmann::json &json, const Saves &saves)
```

Warning: doxygenfunction: Unable to resolve function "to_json" with arguments (nlohmann::json&, const DialogPtr&) in doxygen xml output for project "rollNW" from directory: build/xml/. Potential matches:

```
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json() const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile) const
- nlohmann::json to_json(SerializationProfile profile, ObjectType object_type) const
- nlohmann::json to_json(nw::ResourceType::type restype) const
```

```
- void to_json(nlohmann::json &j, ObjectID id)
- void to_json(nlohmann::json &j, ObjectType type)
- void to_json(nlohmann::json &j, const Ability &type)
- void to_json(nlohmann::json &j, const ArmorClass &type)
- void to_json(nlohmann::json &j, const AttackType &type)
- void to_json(nlohmann::json &j, const BaseItem &type)
- void to_json(nlohmann::json &j, const Class &type)
- void to_json(nlohmann::json &j, const CombatMode &type)
- void to_json(nlohmann::json &j, const Damage &type)
- void to_json(nlohmann::json &j, const DamageModType &type)
- void to_json(nlohmann::json &j, const Disease &type)
- void to_json(nlohmann::json &j, const EffectType &type)
- void to_json(nlohmann::json &j, const Feat &type)
- void to_json(nlohmann::json &j, const ItemPropertyType &type)
- void to_json(nlohmann::json &j, const LocString &loc)
- void to_json(nlohmann::json &j, const MasterFeat &type)
- void to_json(nlohmann::json &j, const MissChanceType &type)
- void to_json(nlohmann::json &j, const ModifierType &type)
- void to_json(nlohmann::json &j, const Poison &type)
- void to_json(nlohmann::json &j, const Race &type)
- void to_json(nlohmann::json &j, const Resource &r)
- void to_json(nlohmann::json &j, const Resref &r)
- void to_json(nlohmann::json &j, const Save &type)
- void to_json(nlohmann::json &j, const SaveVersus &type)
- void to_json(nlohmann::json &j, const Situation &type)
- void to_json(nlohmann::json &j, const Skill &type)
- void to_json(nlohmann::json &j, const SpecialAttack &type)
- void to_json(nlohmann::json &j, const Spell &type)
- void to_json(nlohmann::json &j, const SpellEntry &spell)
- void to_json(nlohmann::json &j, const SpellSchool &type)
- void to_json(nlohmann::json &json, const ByteArray &ba)
- void to_json(nlohmann::json &json, const LevelUp &entry)
- void to_json(nlohmann::json &json, const Location &loc)
- void to_json(nlohmann::json &json, const Saves &saves)
```

void **nw::to_json**(nlohmann::json &json, const *Location* &loc)

void **nw::to_json**(nlohmann::json &j, const *LocString* &loc)

void **nw::to_json**(nlohmann::json &j, const *Resource* &r)

nlohmann::json specialization

void **nw::to_json**(nlohmann::json &j, const *Resref* &r)

nlohmann::json specialization

void **nw::to_json**(nlohmann::json &json, const *Saves* &saves)

void **nw::to_json**(nlohmann::json &j, const *SpellEntry* &spell)

void **nw::to_json**(nlohmann::json &j, *ObjectID* id)

nlohmann::json specialization

void **nw::to_json**(nlohmann::json &j, *ObjectType* type)

nlohmann::json specialization

6.16.73 nw::to_underlying

template<class **Enum**>
constexpr std::underlying_type_t<*Enum*> `nw::to_underlying`(*Enum* e) noexcept

Gets the underlying value of an enum.

Note: Replace when C++23 comes around

Template Parameters

Enum – Any enum type.

6.16.74 nw::to_utf8

std::string `nw::to_utf8`(std::string_view str, std::string_view encoding, bool ignore_errors = false)

Convert to utf8.

6.16.75 nw::to_utf8_by_global_lang

std::string `nw::to_utf8_by_global_lang`(std::string_view str, bool ignore_errors)

6.16.76 nw::to_utf8_by_langid

std::string `nw::to_utf8_by_langid`(std::string_view str, *LanguageID* id, bool ignore_errors)

6.16.77 nwn1::base_attack_bonus

int `nwn1::base_attack_bonus`(const nw::*Creature* *obj)

Calculates base attack bonus.

6.16.78 nwn1::effect_ability_modifier

nw::*Effect* *`nwn1::effect_ability_modifier`(nw::*Ability* ability, int modifier)

Creates an ability modifier effect.

6.16.79 nwn1::effect_armor_class_modifier

nw::*Effect* *`nwn1::effect_armor_class_modifier`(nw::ArmorClass type, int modifier)

Creates an armor class modifier effect.

6.16.80 nwn1::effect_attack_modifier

nw::*Effect* *nwn1::**effect_attack_modifier**(nw::AttackType attack, int modifier)

Creates an attack modifier effect.

6.16.81 nwn1::effect_concealment

nw::*Effect* *nwn1::**effect_concealment**(int value, nw::MissChanceType type = miss_chance_type_normal)

Creates concealment effect.

6.16.82 nwn1::effect_haste

nw::*Effect* *nwn1::**effect_haste**()

Creates a haste effect.

6.16.83 nwn1::effect_miss_chance

nw::*Effect* *nwn1::**effect_miss_chance**(int value, nw::MissChanceType type = miss_chance_type_normal)

Creates miss chance effect.

6.16.84 nwn1::effect_skill_modifier

nw::*Effect* *nwn1::**effect_skill_modifier**(nw::*Skill* skill, int modifier)

Creates an skill modifier effect.

6.16.85 nwn1::get_ability_modifier

int nwn1::**get_ability_modifier**(const nw::*Creature* *obj, nw::*Ability* ability, bool base = false)

Gets creatures ability modifier.

6.16.86 nwn1::get_ability_score

int nwn1::**get_ability_score**(const nw::*Creature* *obj, nw::*Ability* ability, bool base = false)

Gets creatures ability score.

6.16.87 nwn1::get_caster_level

int nwn1::**get_caster_level**(nw::*Creature* *obj, nw::*Class* class_)

Gets creature's caster level for specified class.

6.16.88 nwn1::get_dex_modifier

int `nwn1::get_dex_modifier`(const nw::*Creature* *obj)

Gets creatures dexterity modifier as modified by armor, etc.

6.16.89 nwn1::get_max_hitpoints

int `nwn1::get_max_hitpoints`(const nw::ObjectBase *obj)

Gets objects maximum hit points.

6.16.90 nwn1::get_skill_rank

int `nwn1::get_skill_rank`(const nw::*Creature* *obj, nw::*Skill* skill, nw::ObjectBase *versus = nullptr, bool base = false)

Determines creatures skill rank.

6.16.91 nwn1::get_spell_dc

int `nwn1::get_spell_dc`(nw::*Creature* *obj, nw::*Class* class_, nw::*Spell* spell)

Gets spell DC.

6.16.92 nwn1::get_weapon_by_attack_type

nw::*Item* *`nwn1::get_weapon_by_attack_type`(const nw::*Creature* *obj, nw::AttackType type)

Gets an equipped weapon by attack type.

6.16.93 nwn1::has_effect_type_applied

bool `nwn1::has_effect_type_applied`(nw::ObjectBase *obj, nw::EffectType type)

Determines if object has effect type applied.

6.16.94 nwn1::is_creature_weapon

bool `nwn1::is_creature_weapon`(const nw::*Item* *item)

Determines if item is creature weapon.

6.16.95 nwn1::is_ranged_weapon

bool `nwn1::is_ranged_weapon`(const nw::*Item* *item)

Determines if item is ranged weapon.

6.16.96 nwn1::is_shield

bool `nwn1::is_shield`(nw::*BaseItem* baseitem)

Determines if item is a shield.

6.16.97 nwn1::is_unarmed_weapon

bool `nwn1::is_unarmed_weapon`(const nw::*Item* *item)

Determines if item is unarmed weapon.

6.16.98 nwn1::meets_requirements

Warning: doxygenfunction: Cannot find function "nwn1::meets_requirements" in doxygen xml output for project "rollNW" from directory: build/xml/

6.16.99 nwn1::qual::ability

nw::*Qualifier* `nwn1::qual::ability`(nw::*Ability* id, int min, int max)

6.16.100 nwn1::qual::alignment

nw::*Qualifier* `nwn1::qual::alignment`(nw::*AlignmentAxis* axis, nw::*AlignmentFlags* flags)

6.16.101 nwn1::qual::class_level

nw::*Qualifier* `nwn1::qual::class_level`(nw::*Class* id, int min, int max)

6.16.102 nwn1::qual::feat

nw::*Qualifier* `nwn1::qual::feat`(nw::*Feat* id)

6.16.103 nwn1::qual::level

nw::*Qualifier* `nwn1::qual::level`(int min, int max)

6.16.104 nwn1::qual::race

nw::*Qualifier* `nwn1::qual::race`(nw::*Race* id)

6.16.105 nwn1::qual::skill

nw::*Qualifier* `nwn1::qual::skill`(nw::*Skill* id, int min, int max)

6.16.106 nwn1::resolve_attack

std::unique_ptr<nw::*AttackData*> `nwn1::resolve_attack`(nw::*Creature* *attacker, nw::ObjectBase *target)

Resolves an attack

Note: All transient book keeping is done at the toplevel of this function, any other that attacker and/or target are passed to, are passed as const.

6.16.107 nwn1::resolve_attack_bonus

int `nwn1::resolve_attack_bonus`(const nw::*Creature* *obj, nw::AttackType type, const nw::ObjectBase *versus = nullptr)

Resolves attack bonus.

6.16.108 nwn1::resolve_attack_damage

std::unique_ptr<nw::*AttackData*> `nwn1::resolve_attack`(nw::*Creature* *attacker, nw::ObjectBase *target)

Resolves an attack

Note: All transient book keeping is done at the toplevel of this function, any other that attacker and/or target are passed to, are passed as const.

6.16.109 nwn1::resolve_attack_roll

nw::*AttackResult* `nwn1::resolve_attack_roll`(const nw::*Creature* *obj, nw::AttackType type, const nw::ObjectBase *vs, nw::*AttackData* *data = nullptr)

Resolves an attack roll.

6.16.110 nwn1::resolve_attack_type

nw::AttackType `nwn1::resolve_attack_type`(const nw::*Creature* *obj)

Resolves attack type.

6.16.111 nwn1::resolve_concealment

std::pair<int, bool> `nwn1::resolve_concealment`(const nw::ObjectBase *obj, const nw::ObjectBase *target, bool vs_ranged)

Resolves an concealment - i.e. the highest of concealment and miss chance

Returns

Concealment amount, bool that if `true` is from attacking creature i.e miss chance, if `false` from target object i.e. concealment

6.16.112 nwn1::resolve_creature_damage

nw::*DiceRoll* `nwn1::resolve_creature_damage`(const nw::*Creature* *attacker, nw::*Item* *weapon)

Resolves creature weapon damage.

6.16.113 nwn1::resolve_critical_multiplier

int `nwn1::resolve_critical_multiplier`(const nw::*Creature* *obj, nw::AttackType type, const nw::ObjectBase *vs = nullptr)

Resolves critical multiplier.

6.16.114 nwn1::resolve_critical_threat

int `nwn1::resolve_critical_threat`(const nw::*Creature* *obj, nw::AttackType type)

Resolves critical threat range.

6.16.115 nwn1::resolve_damage_modifiers

void `nwn1::resolve_damage_modifiers`(const nw::*Creature* *obj, const nw::ObjectBase *versus, nw::*AttackData* *data)

Resolves damage modifiers - soak, resist, immunity.

6.16.116 nwn1::resolve_dual_wield_penalty

std::pair<int, int> `nwn1::resolve_dual_wield_penalty`(const nw::*Creature* *obj)

Resolves dual-wield penalty.

6.16.117 nwn1::resolve_iteration_penalty

int `nwn1::resolve_iteration_penalty`(const nw::*Creature* *attacker, nw::AttackType type)

Resolves iteration penalty.

6.16.118 nwn1::resolve_number_of_attacks

std::pair<int, int> `nwn1::resolve_number_of_attacks`(const nw::*Creature* *obj)

Resolves number of attacks.

6.16.119 nwn1::resolve_saving_throw

bool `nwn1::resolve_saving_throw`(const nw::ObjectBase *obj, nw::*Save* type, int dc, nw::SaveVersus type_vs, const nw::ObjectBase *versus)

6.16.120 nwn1::resolve_skill_check

bool `nwn1::resolve_skill_check`(const nw::*Creature* *obj, nw::*Skill* skill, int dc, nw::ObjectBase *versus)

6.16.121 nwn1::resolve_unarmed_damage

nw::*DiceRoll* `nwn1::resolve_unarmed_damage`(const nw::*Creature* *attacker)

Resolve unarmed base damage.

6.16.122 nwn1::resolve_weapon_damage

nw::*DiceRoll* `nwn1::resolve_weapon_damage`(const nw::*Creature* *attacker, nw::*BaseItem* item)

Resolve weapon base damage

Note: Includes specialization and arcane archer bonuses

6.16.123 nwn1::resolve_weapon_damage_flags

nw::*DamageFlag* `nwn1::resolve_weapon_damage_flags`(const nw::*Item* *weapon)

Resolve weapon base damage flags.

6.16.124 nwn1::saving_throw

int `nwn1::saving_throw`(const nw::ObjectBase *obj, nw::*Save* type, nw::SaveVersus type_vs, const nw::ObjectBase *versus)

6.16.125 nwn1::sel::ability

nw::*Selector* `nwn1::sel::ability`(nw::*Ability* id)

6.16.126 nwn1::sel::alignment

nw::*Selector* `nwn1::sel::alignment`(nw::*AlignmentAxis* id)

6.16.127 nwn1::sel::class_level

nw::*Selector* `nwn1::sel::class_level`(nw::*Class* id)

6.16.128 nwn1::sel::feat

nw::*Selector* `nwn1::sel::feat`(nw::*Feat* id)

6.16.129 nwn1::sel::level

nw::*Selector* `nwn1::sel::level`()

6.16.130 nwn1::sel::race

nw::*Selector* `nwn1::sel::race`()

6.16.131 nwn1::sel::skill

nw::*Selector* `nwn1::sel::skill`(nw::*Skill* id)

6.16.132 nwn1::selector

nw::*RuleValue* `nwn1::selector`(const nw::*Selector* &selector, const nw::ObjectBase *obj)

6.16.133 nwn1::weapon_is_finessable

bool `nwn1::weapon_is_finessable`(const nw::*Creature* *obj, nw::*Item* *weapon)

Determines if a weapon is finessable.

6.16.134 nwn1::weapon_iteration

int `nwn1::weapon_iteration`(const nw::*Creature* *obj, const nw::*Item* *weapon)

Calculates weapon iteration, e.g. 5 or 3 for monk weapons.

6.17 typedefs

6.17.1 nw::AbilityArray

using `nw::AbilityArray` = *RuleTypeArray*<*Ability*, *AbilityInfo*>

Ability singleton component.

6.17.2 nw::BaseItemArray

using `nw::BaseItemArray` = *RuleTypeArray*<*BaseItem*, BaseItemInfo>

BaseItem singleton component.

6.17.3 nw::DamageFlag

using `nw::DamageFlag` = *RuleFlag*<Damage, 32>

6.17.4 nw::EquipItem

using `nw::EquipItem` = std::variant<*Resref* , *Item*>

6.17.5 nw::FeatArray

using `nw::FeatArray` = *RuleTypeArray*<*Feat*, *FeatInfo*>

Feat Singleton Component.

6.17.6 nw::LocalVarTable

using `nw::LocalVarTable` = absl::flat_hash_map<std::string, *LocalVar*>

6.17.7 nw::ModifierFunction

using `nw::ModifierFunction` = std::function<*ModifierResult*(const ObjectBase*)>

6.17.8 nw::ModifierInputs

Warning: doxygentypedef: Cannot find typedef "nw::ModifierInputs" in doxygen xml output for project "rollNW" from directory: build/xml/

6.17.9 nw::ModifierOutputs

Warning: doxygentypedef: Cannot find typedef "nw::ModifierOutputs" in doxygen xml output for project "rollNW" from directory: build/xml/

6.17.10 nw::ModifierResult

using `nw::ModifierResult` = *Variant*<int, float, *DamageRoll*>

6.17.11 nw::ModifierVariant

using `nw::ModifierVariant` = *Variant*<int, float, *DamageRoll*, *ModifierFunction*, ModifierSubFunction, ModifierVsFunction, ModifierSubVsFunction>

6.17.12 nw::RaceArray

using `nw::RaceArray` = *RuleTypeArray*<*Race*, *RaceInfo*>

Race singleton component.

6.17.13 nw::RuleValue

using `nw::RuleValue` = *Variant*<int32_t, float, std::string>

6.17.14 nw::SkillArray

using `nw::SkillArray` = *RuleTypeArray*<*Skill*, *SkillInfo*>

Singleton Component for Skills.

6.17.15 nw::SpellArray

using `nw::SpellArray` = *RuleTypeArray*<*Spell*, *SpellInfo*>

Spell singleton component.

6.17.16 nw::sqlite3_ptr

using `nw::sqlite3_ptr` = std::unique_ptr<sqlite3, void (*)(sqlite3*)>

sqlite3 database pointer wrapper

6.17.17 nw::unique_container

using `nw::unique_container` = std::unique_ptr<*Container*>

6.18 rollnw

class `rollnw.Appearance`

Class containing creature's appearance

body_parts: *BodyParts*

body_parts

hair: **int**

hair

id: **int**

Index into `appearance.2da`

phenotype: **int**

phenotype

portrait_id: **int**

Index into `portraits.2da`

skin: **int**

skin

tail: **int**

tail

tattoo1: **int**

tattoo1

tattoo2: **int**

tattoo2

wings: **int**

wings

class rollnw.**Area**

Area object

comments: **str**

creator_id: **int**

creatures: **List[*Creature*]**

doors: **List[*Door*]**

encounters: **List[*Encounter*]**

flags: ***AreaFlags***

height: **int**

id: **int**

items: **List[*Item*]**

json_archive_version: **ClassVar[int]**

listen_check_mod: **int**

loadscreen: **int**

name: ***LocString***

no_rest: **int**

object_type: **ClassVar[int]**

placeables: **List[*Placeable*]**

pvp: **int**

scripts: ***AreaScripts***

shadow_opacity: **int**

skybox: **int**

sounds: **List[*Sound*]**

spot_check_mod: **int**

stores: **List[*Store*]**

tiles: List[*Tile*]

tileset: str

triggers: List[*Trigger*]

version: int

waypoints: List[*Waypoint*]

weather: *AreaWeather*

width: int

flag rollnw.**AreaFlags**(*value*)

Valid values are as follows:

interior = <AreaFlags.interior: 1>

underground = <AreaFlags.underground: 2>

natural = <AreaFlags.natural: 4>

class rollnw.**AreaScripts**

Area's scripts

on_enter: str

on_exit: str

on_heartbeat: str

on_user_defined: str

class rollnw.**AreaWeather**

Area's weather

chance_lightning: int

chance_rain: int

chance_snow: int

color_moon_ambient: int

color_moon_diffuse: int

color_moon_fog: int

color_sun_ambient: int

color_sun_diffuse: int

color_sun_fog: int

day_night_cycle: int

fog_clip_distance: int

fog_moon_amount: int

fog_sun_amount: int

is_night: int

lighting_scheme: int

moon_shadows: int

sun_shadows: int

wind_power: int

class rollnw.**AttackData**

Class aggregating attack data

Attributes:

armor_class: int

attack_bonus: int

attack_roll: int

attacker: *Creature*

concealment: int

is_ranged_attack: bool

iteration_penalty: int

multiplier: int

nth_attack: int

result: *AttackResult*

target: *ObjectBase*

target_is_creature: bool

target_state: int

threat_range: int

type: int

enum rollnw.**AttackResult**(*value*)

Attack Result Type

Member Type
int

Valid values are as follows:

hit_by_auto_success = <AttackResult.hit_by_auto_success: 1>

hit_by_critical = <AttackResult.hit_by_critical: 2>

hit_by_roll = <AttackResult.hit_by_roll: 3>

miss_by_auto_fail = <AttackResult.miss_by_auto_fail: 4>

miss_by_concealment = <AttackResult.miss_by_concealment: 5>

miss_by_miss_chance = <AttackResult.miss_by_miss_chance: 6>

miss_by_roll = <AttackResult.miss_by_roll: 7>

class rollnw.**BodyParts**

Class containing references to creature's body parts

belt: int

bicep_left: int

bicep_right: int

foot_left: int

foot_right: int

forearm_left: int

forearm_right: int

hand_left: int

hand_right: int

head: int

neck: int

pelvis: int

shin_left: int

shin_right: int

shoulder_left: int

shoulder_right: int

thigh_left: int

thigh_right: int

class rollnw.**ClassEntry**

Class level data

id: int

level: int

spells: *SpellBook*

class rollnw.**CombatInfo**

Class containing combat related data

ac_armor_base: int

ac_natural_bonus: int

ac_shield_base: int

combat_mode: int

size_ab_modifier: int

size_ac_modifier: int

target_state: int

class rollnw.**Common**

Class containing attributes common to all objects

comment: str

locals: *LocalData*

location: *Location*

name: *LocString*

palette_id: int

resref: str

tag: str

class rollnw.**Container**

Base container interface

all()

Get all resources

contains(*res:* Resource | *str*) → bool

Get if container contains resource

demand(*res:* Resource | *str*) → bytes

Reads resource data, empty ByteArray if no match.

extract(*pattern*, *output*) → int

Extract elements from a container by regex

extract_by_glob(*glob: str*, *output: str*) → int

Extract elements from a container by glob pattern

name() → str

Equivalent to *basename path()*

path() → str

Path to container, for basic containers, should be canonical

size() → int

Gets the number of resources, if applicable, of the container

stat(*res*) → *ResourceDescriptor*

Get some general data about a resource

valid() → bool

Return true if loaded, false if not.

working_directory() → str

Get container working directory

class rollnw.**Creature**

Class that represents a Creature object

appearance: *Appearance*

bodybag: **int**

chunk_death: **int**

common: *Common*

conversation: **str**

Dialog resref

cr: **float**

cr_adjust: **int**

decay_time: **int**

deity: **str**

description: *LocString*

disarmable: **int**

property equipment: *Equips*

Gets creatures equipped items

faction_id: **int**

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

gender: **int**

good_evil: **int**

property history: *LevelHistory*

Gets creatures level history

hp: **int**

hp_current: **int**

hp_max: **int**

immortal: **int**

interruptable: **int**

property inventory: *Inventory*

Gets creatures inventory

json_archive_version: ClassVar[int]

lawful_chaotic: int

levels: *LevelStats*

lootable: int

name_first: *LocString*

name_last: *LocString*

object_type: ClassVar[int]

pc: int

perception_range: float

plot: int

race: int

scripts: *CreatureScripts*

soundset: str

starting_package: int

stats: *CreatureStats*

Offensive and defensive stats.

subrace: str

walkrate: int

class rollnw.**CreatureScripts**

A class containing a creature's script set.

on_attacked: str

on_blocked: str

on_conversation: str

on_damaged: str

on_death: str

on_disturbed: str

on_endround: str

on_heartbeat: str

on_perceived: str

on_rested: str

on_spawn: str

on_spell_cast_at: str

on_user_defined: str

class rollnw.**CreatureStats**

Implementation of a creature's general attributes and stats

add_feat(*feat*) → bool

Attempts to add a feat to a creature, returning true if successful

get_ability_score(*id: int*)

Gets an ability score

get_skill_rank(*id: int*)

Gets a skill rank

has_feat(*feat*) → bool

Determines if creature has feat

set_ability_score(*id: int*, *value: int*) → bool

Sets an ability score, returning true if successful

set_skill_rank(*id: int*, *value: int*) → bool

Sets a skill rank, returning true if successful

class rollnw.**Dialog**

add() → *DialogPtr*

Adds empty Dialog Pointer and Node

add_ptr(*ptr:* DialogPtr, *is_link: bool = False*) → *DialogPtr*

Adds Dialog Pointer, if *is_link* is false no new pointer or node is created. if *is_link* is true a new pointer will created with the node copied from input pointer.

add_string(*value: str*, *lang:* LanguageID = *LanguageID.english*, *feminine: bool = False*) → *DialogPtr*

Adds Dialog Pointer and Node with string value set

delay_entry: int = 0

delay_reply: int = 0

delete_ptr(*ptr:* DialogPtr)

Deletes a dialog pointer @warning `ptr` should be removed from / not added to a dialog prior to deletion

static from_file(*path: str*) → *Dialog*

Creates a dialog from a GFF or rollnw JSON file

json_archive_version: ClassVar[int]

prevent_zoom: bool = False

remove_ptr(*ptr:* DialogPtr)

Removes Dialog Ptr from underlying node

restype: ClassVar[*ObjectType*]

save(*path: str*)

Saves a dialog to file, valid extentions are ".dlg" and ".dlg.json"

script_abort: str

script_end: str

valid() → bool

Checks id dialog was successfully parsed

word_count: int = 0

enum rollnw.**DialogAnimation**(*value*)

Member Type
int

Valid values are as follows:

default = <DialogAnimation.default: 0>

taunt = <DialogAnimation.taunt: 28>

greeting = <DialogAnimation.greeting: 29>

listen = <DialogAnimation.listen: 30>

worship = <DialogAnimation.worship: 33>

salute = <DialogAnimation.salute: 34>

bow = <DialogAnimation.bow: 35>

steal = <DialogAnimation.steal: 37>

talk_normal = <DialogAnimation.talk_normal: 38>

talk_pleading = <DialogAnimation.talk_pleading: 39>

talk_forceful = <DialogAnimation.talk_forceful: 40>

talk_laugh = <DialogAnimation.talk_laugh: 41>

victory_1 = <DialogAnimation.victory_1: 44>

victory_2 = <DialogAnimation.victory_2: 45>

victory_3 = <DialogAnimation.victory_3: 46>

look_far = <DialogAnimation.look_far: 48>

drink = <DialogAnimation.drink: 70>

read = <DialogAnimation.read: 71>

none = <DialogAnimation.none: 88>

class rollnw.**DialogNode**

animation: *DialogAnimation* **= 0**

comment: str

copy() → *DialogNode*

Copies a Node

delay: int = -1

get_action_param(*key: str*) → str | None

Gets action parameter if it exists

parent: *Dialog*

pointers: List[*DialogPtr*]

quest: str

quest_entry: int = -1

remove_action_param(*key: str*)

Removes action parameter by key

script_action: str

set_action_param(*key: str*, *value: str*)

Sets action parameter, if key does not exist key and value are appended

sound: str

speaker: str

text: *LocString*

type: *DialogNodeType*

enum rollnw.**DialogNodeType**(*value*)

Member Type
int

Valid values are as follows:

entry = <DialogNodeType.entry: 0>

reply = <DialogNodeType.reply: 1>

class rollnw.**DialogPtr**

add() → *DialogPtr*

Adds empty Dialog Pointer and Node

add_ptr(*ptr:* DialogPtr, *is_link: bool = False*) → *DialogPtr*

Adds Dialog Pointer, if *is_link* is false no new pointer or node is created. if *is_link* is true a new pointer will created with the node copied from input pointer.

add_string(*value: str*, *lang:* LanguageID = *LanguageID.english*, *feminine: bool = False*) → *DialogPtr*

Adds Dialog Pointer and Node with string value set

comment: str

copy() → *DialogPtr*

Copies dialog pointer and all sub-nodes

get_condition_param(*key: str*) → str | None

Gets condition parameter by key

is_link: **bool**

is_start: **bool**

node: ***DialogNode***

parent: ***Dialog***

remove_condition_param(*key: str*)

Removes condition parameter by key

remove_ptr(*ptr:* DialogPtr)

Removes Dialog Ptr from underlying node

script_appears: **str**

set_condition_param(*key: str*, *value: str*)

Sets condition parameter, if key does not exist key and value are appended

type: ***DialogNodeType***

class rollnw.**DiceRoll**

Dice roll

bonus: **int**

dice: **int**

sides: **int**

class rollnw.**Directory**(*path: str*)

Implementation of a directory as a rollnw.Container

Parameters

path (*str*) – Directory to load

class rollnw.**Door**

Class that represents a Door object

animation_state: ***DoorAnimationState***

appearance: **int**

conversation: **str**

Door's conversation resref

description: ***LocString***

faction: **int**

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

generic_type: int

hardness: int

hp: int

hp_current: int

interruptable: int

json_archive_version: ClassVar[int]

linked_to: str

linked_to_flags: int

loadscreen: int

lock: *Lock*

object_type: ClassVar[int]

plot: int

portrait_id: int

saves: *Saves*

scripts: *DoorScripts*

trap: *Trap*

enum rollnw.DoorAnimationState(*value*)

Door animation states

Valid values are as follows:

closed = <DoorAnimationState.closed: 1>

opened1 = <DoorAnimationState.opened1: 2>

opened2 = <DoorAnimationState.opened2: 3>

class rollnw.DoorScripts

Door's scripts

on_click: str

on_closed: str

on_damaged: str

on_death: str

on_disarm: str

on_heartbeat: str

on_lock: str

on_melee_attacked: str

on_open: str

on_open_failure: str

on_spell_cast_at: str

on_trap_triggered: str

on_unlock: str

on_user_defined: str

class rollnw.**Effect**

clear()

Clears the effect such that it's as if default constructed

get_float(*index*)

Gets a floating point value

get_int(*index*)

Gets an integer point value

get_string(*index*)

Gets a string value

handle()

Gets the effect's handle

id()

Gets the effect's ID

set_float(*index: int*, *value: float*)

Sets a floating point value

set_int(*index: int*, *value: int*)

Sets an integer point value

set_string(*index: int*, *value: str*)

Sets a string value

set_versus(*vs*)

Sets the versus value

versus()

Gets the versus value

enum rollnw.**EffectCategory**(*value*)

Effect category

Member Type

`int`

Valid values are as follows:

magical = <EffectCategory.magical: 1>

extraordinary = <EffectCategory.extraordinary: 2>

supernatural = <EffectCategory.supernatural: 3>

item = <EffectCategory.item: 4>

innate = <EffectCategory.innate: 5>

class rollnw.**EffectHandle**

Effect Handle

category: *EffectCategory*

creator: *ObjectHandle*

effect: *Effect*

spell_id: int

subtype: int

type: int

class rollnw.**EffectID**

Effect ID

index: int

version: int

class rollnw.**Encounter**

Class that represents an Encounter object

active: bool

creatures: List[*SpawnCreature*]

creatures_max: int

creatures_recommended: int

difficulty: int

difficulty_index: int

faction: int

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

geometry: List[*Vector3*]

player_only: bool

reset: bool

reset_time: int

respawns: int

scripts: *EncounterScripts*

spawn_option: int

spawn_points: List[*SpawnPoint*]

class rollnw.**EncounterScripts**

Encounter's scripts

on_entered: str

on_exhausted: str

on_exit: str

on_heartbeat: str

on_user_defined: str

enum rollnw.**EquipIndex**(*value*)

Member Type
int

Valid values are as follows:

head = <EquipIndex.head: 0>

chest = <EquipIndex.chest: 1>

boots = <EquipIndex.boots: 2>

arms = <EquipIndex.arms: 3>

righthand = <EquipIndex.righthand: 4>

lefthand = <EquipIndex.lefthand: 5>

cloak = <EquipIndex.cloak: 6>

leftring = <EquipIndex.leftring: 7>

rightring = <EquipIndex.rightring: 8>

neck = <EquipIndex.neck: 9>

belt = <EquipIndex.belt: 10>

arrows = <EquipIndex.arrows: 11>

bullets = <EquipIndex.bullets: 12>

bolts = <EquipIndex.bolts: 13>

creature_left = <EquipIndex.creature_left: 14>

creature_right = <EquipIndex.creature_right: 15>

creature_bite = <EquipIndex.creature_bite: 16>

creature_skin = <EquipIndex.creature_skin: 17>

invalid = <EquipIndex.invalid: 4294967295>

flag rollnw.**EquipSlot**(*value*)

Equipment slot flags

Valid values are as follows:

head = <EquipSlot.head: 1>

chest = <EquipSlot.chest: 2>

boots = <EquipSlot.boots: 4>

arms = <EquipSlot.arms: 8>

righthand = <EquipSlot.righthand: 16>

lefthand = <EquipSlot.lefthand: 32>

cloak = <EquipSlot.cloak: 64>

leftring = <EquipSlot.leftring: 128>

rightring = <EquipSlot.rightring: 256>

neck = <EquipSlot.neck: 512>

belt = <EquipSlot.belt: 1024>

arrows = <EquipSlot.arrows: 2048>

bullets = <EquipSlot.bullets: 4096>

bolts = <EquipSlot.bolts: 8192>

creature_left = <EquipSlot.creature_left: 16384>

creature_right = <EquipSlot.creature_right: 32768>

creature_bite = <EquipSlot.creature_bite: 65536>

creature_skin = <EquipSlot.creature_skin: 131072>

class rollnw.**Equips**

Creature's equipment

equips: List[str | *Item*]

len(equips) == 18

Type

Note

instantiate()

Instantiates equipment by loading contained items from the resource manager

class rollnw.**Erf**(*path: str*)

Implementation of Erf file format

Parameters

path (*str*) – Erf file to load

add(*path*)

Adds resources from path

erase(*resource*)

Removes resource

merge(*container*)

Merges the contents of another rollnw.Container

reload()

Reloads Erf

Notes

Erf:: working_directory() will not change

save()

Saves Erf to Erf:: path()

Notes

It's probably best to call Erf:: reload after save.

save_as(*path*)

Saves Erf to different path

Notes

Current Erf unmodified, to load Erf at new path a new Erf must be constructed.

enum rollnw.**GameVersion**(*value*)

Game versions

Valid values are as follows:

v1_69 = <GameVersion.v1_69: 1>

vEE = <GameVersion.vEE: 2>

nwn2 = <GameVersion.nwn2: 3>

class rollnw.**IVector4**

w: int

x: int

y: int

z: int

class rollnw.**Image**(*filename: str*)

Loads an image

Parameters

filename (*str*) – image file to load

channels()

Gets BPP

data()

Get raw data

height()

Get height

valid()

Determine if successfully loaded.

width()

Get width

write_to()

Write Image to file

class rollnw.**Ini**(*filename: str*)

Loads an ini

Parameters

filename (*str*) – ini file to load

get_float(*key: str*) → float | None

Gets an INI value

get_int(*key: str*) → int | None

Gets an INI value

get_str(*key: str*) → str | None

Gets an INI value

valid()

Deterimes if Ini file was successfully parsed

class rollnw.**Inventory**

An Object's inventory

instantiate()

Instantiates inventory by loading contained items from the resource manager

items: **List[*InventoryItem*]**

owner: ***ObjectBase***

class rollnw.**InventoryItem**

An inventory item

infinite: **bool**

Only applicable to stores

item: **str | *Item***

x: **int**

y: **int**

class rollnw.**Item**

Class that represents an Item object

additional_cost: int

baseitem: int

charges: int

cost: int

cursed: bool

description: *LocString*

Description

description_id: *LocString*

Description after being identified.

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

identified: bool

inventory: *Inventory*

model_colors: List[int]

model_parts: List[int]

model_type: *ItemModelType*

plot: bool

Is a plot item.

properties: List[*ItemProperty*]

stacksize: int

stolen: bool

enum rollnw.**ItemColors**(*value*)

Valid values are as follows:

cloth1 = <ItemColors.cloth1: 1>

cloth2 = <ItemColors.cloth2: 2>

leather1 = <ItemColors.leather1: 3>

leather2 = <ItemColors.leather2: 4>

metal1 = <ItemColors.metal1: 5>

metal2 = <ItemColors.metal2: 6>

enum rollnw.**ItemModelParts**(*value*)

Valid values are as follows:

model1 = <ItemModelParts.model1: 1>

model2 = <ItemModelParts.model2: 2>

model3 = <ItemModelParts.model3: 3>

armor_belt = <ItemModelParts.armor_belt: 4>

armor_lbicep = <ItemModelParts.armor_lbicep: 5>

armor_lfarm = <ItemModelParts.armor_lfarm: 6>

armor_lfoot = <ItemModelParts.armor_lfoot: 7>

armor_lhand = <ItemModelParts.armor_lhand: 8>

armor_lshin = <ItemModelParts.armor_lshin: 9>

armor_lshoul = <ItemModelParts.armor_lshoul: 10>

armor_lthigh = <ItemModelParts.armor_lthigh: 11>

armor_neck = <ItemModelParts.armor_neck: 12>

armor_pelvis = <ItemModelParts.armor_pelvis: 13>

armor_rbicep = <ItemModelParts.armor_rbicep: 14>

armor_rfarm = <ItemModelParts.armor_rfarm: 15>

armor_rfoot = <ItemModelParts.armor_rfoot: 16>

armor_rhand = <ItemModelParts.armor_rhand: 17>

armor_robe = <ItemModelParts.armor_robe: 18>

armor_rshin = <ItemModelParts.armor_rshin: 19>

armor_rshoul = <ItemModelParts.armor_rshoul: 20>

armor_rthigh = <ItemModelParts.armor_rthigh: 21>

armor_torso = <ItemModelParts.armor_torso: 22>

enum rollnw.**ItemModelType**(*value*)

Valid values are as follows:

simple = <ItemModelType.simple: 1>

layered = <ItemModelType.layered: 2>

composite = <ItemModelType.composite: 3>

armor = <ItemModelType.armor: 4>

class rollnw.**ItemProperty**

An item property

cost_table: int

cost_value: int

param_table: int

param_value: int

subtype: int

type: int

class rollnw.**Key**(*path: str*)

Implementation Key/Bif file format as a rollnw.Container

Parameters
path (*str*) – Path to key file

class rollnw.**Language**

static encoding(*language:* LanguageID) → str

Gets the encoding for a particular language

static from_string(*string: str*) → *LanguageID*

Converts string (short or long form) to ID

static has_feminine(*language:* LanguageID) → bool

Determines if language has feminine translations

static to_base_id(*id: int*) → Tuple[*LanguageID*, bool]

Convert runtime language identifier to base language and bool indicating masc/fem.

static to_runtime_id(*language:* LanguageID, *feminine: bool = False*) → int

Convert language ID to runtime identifier.

static to_string(*language:* LanguageID, *long_name: bool = False*) → str

Converts language to string form

enum rollnw.**LanguageID**(*value*)

Member Type
int

Valid values are as follows:

invalid = <LanguageID.invalid: -1>

english = <LanguageID.english: 0>

french = <LanguageID.french: 1>

german = <LanguageID.german: 2>

italian = <LanguageID.italian: 3>

spanish = <LanguageID.spanish: 4>

polish = <LanguageID.polish: 5>

korean = <LanguageID.korean: 128>

chinese_traditional = <LanguageID.chinese_traditional: 129>

chinese_simplified = <LanguageID.chinese_simplified: 130>

japanese = <LanguageID.japanese: 131>

class rollnw.**LevelHistory**

Implements a creatures levelup history

entries: List[*LevelUp*]

Entries for levels

class rollnw.**LevelStats**

Implements a creatures level related stats

entries: List[*ClassEntry*]

Entries for levels

level() → int

Gets total level

level_by_class(*class_: int*) → int

Gets level by class

class rollnw.**LevelUp**

Level up data

ability: int

Ability score that was raised, if any. -1 if none

class_: int

Class the level was taken as

epic: bool

`True` if level is an epic level

feats: List[int]

Added feats

hitpoints: int

Hitpoints gained.

known_spells: List[Tuple[int, int]]

Level, Spell pair for gained spells

skillpoints: int

Roll over skill points

skills: List[Tuple[int, int]]

Skill and the amount increased

class rollnw.**LocString**(*strref: int = -1*)

Implements a localized string

Parameters

strref (*int*) – String reference. (default -1)

add(*language:* LanguageID, *string: str*, *feminine: bool = False*)

Adds a localized string

contains(*language:* LanguageID, *feminine: bool = False*)

Checks if a localized string is contained

static from_dict(*data: dict*)

Converts python `dict` to `LocString`

get(*language:* LanguageID, *feminine: bool = False*)

Gets a localized string

remove(*language:* LanguageID, *feminine: bool = False*)

Removes a localized string

size()

Gets number of localized strings

strref()

Gets string reference

to_dict() → DefaultDict

Converts `LocString` to python `dict`

class rollnw.**LocalData**

delete_float(*varname: str*)

Deletes float variable

delete_int(*varname: str*)

Deletes int variable

delete_location(*varname: str*)

Deletes location variable

delete_object(*varname: str*)

Deletes object variable

delete_string(*varname: str*)

Deletes string variable

get_float(*varname: str*)

Gets float variable

get_int(*varname: str*) → int

Gets int variable

get_location(*varname: str*)

Gets location variable

get_object(*varname: str*)

Gets object variable

get_string(*varname: str*) → str

Gets string variable

set_float(*varname: str*, *value: float*)

Sets float variable

set_int(*varname: str*, *value: int*)

Sets int variable

set_location(*varname: str*, *value:* Location)

Sets location variable

set_object(*varname: str*, *value:* ObjectHandle)

Sets object variable

set_string(*varname: str*, *value: str*)

Sets string variable

size()

Gets number of variables

class rollnw.**Location**

Class representing an objects location

area: int

orientation: *Vector3*

position: *Vector3*

class rollnw.**Lock**

Class representing a lock on an object

key_name: str

key_required: bool

lock_dc: int

lockable: bool

locked: bool

remove_key: bool

unlock_dc: int

class rollnw.**Module**

Class that represents a Module object

area_count() → int

Gets number of areas in module

creator: int

dawn_hour: int

description: *LocString*

dusk_hour: int

entry_area: str

entry_orientation: *Vector3*

entry_position: *Vector3*

expansion_pack: int

get_area(*index: int*) → *Area* | None

Gets number of areas in module

haks: List[str]

id: ByteString

is_save_game: bool

locals: *LocalData*

min_game_version: int

minutes_per_hour: int

name: *LocString*

scripts: *ModuleScripts*

start_day: int

start_hour: int

start_month: int

start_movie: str

start_year: int

tag: str

tlk: str

property uuid: str

Gets modules UUID

version: int

xpscale: int

class rollnw.**ModuleScripts**

Module Scripts

on_client_enter: str

on_client_leave: str

on_cutsnabort: str

on_heartbeat: str

on_item_acquire: str

on_item_activate: str

on_item_unaquire: str

on_load: str

on_player_chat: str

on_player_death: str

on_player_dying: str

on_player_equip: str

on_player_level_up: str

on_player_rest: str

on_player_uneqiup: str

on_spawnbtndn: str

on_start: str

on_user_defined: str

class rollnw.**NWSync**(*path: str*)

Implementation of NWSync file format

Parameters
path (*str*) – Path to NWSync repository

get(*manifest*)

Gets a particular manifest as a container

is_loaded()

Gets if NWSync was successfully loaded

manifests()

Get list of all manifests

shard_count()

Get the number of shards

class rollnw.**NWSyncManifest**

Implementation of NWSync Manifest as a rollnw.Container

class rollnw.**ObjectBase**

handle()

Gets object handle

class rollnw.**ObjectHandle**

Object handle

id: int

index into object array

type: *ObjectType*

object type

valid()

Determines if handle is valid

version: int

object index version

enum rollnw.**ObjectType**(*value*)

Object types

Valid values are as follows:

invalid = <ObjectType.invalid: 1>

gui = <ObjectType.gui: 2>

tile = <ObjectType.tile: 3>

module = <ObjectType.module: 4>

area = <ObjectType.area: 5>

creature = <ObjectType.creature: 6>

item = <ObjectType.item: 7>

trigger = <ObjectType.trigger: 8>

projectile = <ObjectType.projectile: 9>

placeable = <ObjectType.placeable: 10>

door = <ObjectType.door: 11>

areaofeffect = <ObjectType.areaofeffect: 12>

waypoint = <ObjectType.waypoint: 13>

encounter = <ObjectType.encounter: 14>

store = <ObjectType.store: 15>

portal = <ObjectType.portal: 16>

sound = <ObjectType.sound: 17>

class rollnw.**Placeable**

Class that represents a Placeable object

animation_state: *PlaceableAnimationState*

appearance: int

bodybag: int

common: *Common*

conversation: str

description: *LocString*

faction: int

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

hardness: int

has_inventory: bool

hp: int

hp_current: int

interruptable: bool

inventory: *Inventory*

json_archive_version: ClassVar[int]

lock: *Lock*

object_type: ClassVar[int]

plot: bool

portrait_id: int

saves: *Saves*

scripts: *PlaceableScripts*

static: bool

trap: *Trap*

useable: bool

enum rollnw.**PlaceableAnimationState**(*value*)

Valid values are as follows:

none = <PlaceableAnimationState.none: 1>

open = <PlaceableAnimationState.open: 2>

closed = <PlaceableAnimationState.closed: 3>

destroyed = <PlaceableAnimationState.destroyed: 4>

activated = <PlaceableAnimationState.activated: 5>

deactivated = <PlaceableAnimationState.deactivated: 6>

class rollnw.**PlaceableScripts**

Placeable's scripts

on_click: str

on_closed: str

on_damaged: str

on_death: str

on_disarm: str

on_heartbeat: str

on_inventory_disturbed: str

on_lock: str

on_melee_attacked: str

on_open: str

on_spell_cast_at: str

on_trap_triggered: str

on_unlock: str

on_used: str

on_user_defined: str

class rollnw.**Player**

Player character

> **Warning:** This is very incomplete

class rollnw.**Plt**

Implementation of PLT file format

height()

pixels()

valid()

width()

class rollnw.**PltColors**

Plt Color Array

Notes

This would be the colors that a player would select

colors: List[int]

enum rollnw.**PltLayer**(*value*)

Plt layers

Member Type
int

Valid values are as follows:

plt_layer_skin = <PltLayer.plt_layer_skin: 0>

plt_layer_hair = <PltLayer.plt_layer_hair: 1>

plt_layer_metal1 = <PltLayer.plt_layer_metal1: 2>

plt_layer_metal2 = <PltLayer.plt_layer_metal2: 3>

plt_layer_cloth1 = <PltLayer.plt_layer_cloth1: 4>

plt_layer_cloth2 = <PltLayer.plt_layer_cloth2: 5>

plt_layer_leather1 = <PltLayer.plt_layer_leather1: 6>

plt_layer_leather2 = <PltLayer.plt_layer_leather2: 7>

plt_layer_tattoo1 = <PltLayer.plt_layer_tattoo1: 8>

plt_layer_tattoo2 = <PltLayer.plt_layer_tattoo2: 9>

class rollnw.**PltPixel**

Plt pixel

color

layer

class rollnw.**Resource**

Resource name

Parameters

- **name** (*str*) – resref or filename
- **type** (ResourceType | *None*) – (Default None)

Notes

If a resource type is not passed `name` is assumed to be a file name, e.g. 'nw_chicken.utc'

resref

Type
str

type

Type
ResourceType

filename() → str

Returns resource as 'resref.ext'

static from_filename(*filename: str*) → *Resource*

Creates resource from file name

valid() → bool

Determines if is valid resource name

class rollnw.**ResourceDescriptor**

Resource descriptor

name

size

mtime

parent

enum rollnw.**ResourceType**(*value*)

Valid values are as follows:

invalid = <ResourceType.invalid: 1>

container = <ResourceType.container: 2>

gff_archive = <ResourceType.gff_archive: 3>

movie = <ResourceType.movie: 4>

player = <ResourceType.player: 5>

sound = <ResourceType.sound: 6>

texture = <ResourceType.texture: 7>

json = <ResourceType.json: 8>

bmp = <ResourceType.bmp: 9>

mve = <ResourceType.mve: 10>

tga = <ResourceType.tga: 11>

wav = <ResourceType.wav: 12>

plt = <ResourceType.plt: 13>

ini = <ResourceType.ini: 14>

bmu = <ResourceType.bmu: 15>

mpg = <ResourceType.mpg: 16>

txt = <ResourceType.txt: 17>

plh = <ResourceType.plh: 18>

tex = <ResourceType.tex: 19>

mdl = <ResourceType.mdl: 20>

thg = <ResourceType.thg: 21>

fnt = <ResourceType.fnt: 22>

lua = <ResourceType.lua: 23>

slt = <ResourceType.slt: 24>

nss = <ResourceType.nss: 25>

ncs = <ResourceType.ncs: 26>

mod = <ResourceType.mod: 27>

```
are = <ResourceType.are:  28>
set = <ResourceType.set:  29>
ifo = <ResourceType.ifo:  30>
bic = <ResourceType.bic:  31>
wok = <ResourceType.wok:  32>
twoda = <ResourceType.twoda:  33>
tlk = <ResourceType.tlk:  34>
txi = <ResourceType.txi:  35>
git = <ResourceType.git:  36>
bti = <ResourceType.bti:  37>
uti = <ResourceType.uti:  38>
btc = <ResourceType.btc:  39>
utc = <ResourceType.utc:  40>
dlg = <ResourceType.dlg:  41>
itp = <ResourceType.itp:  42>
btt = <ResourceType.btt:  43>
utt = <ResourceType.utt:  44>
dds = <ResourceType.dds:  45>
bts = <ResourceType.bts:  46>
uts = <ResourceType.uts:  47>
ltr = <ResourceType.ltr:  48>
gff = <ResourceType.gff:  49>
fac = <ResourceType.fac:  50>
bte = <ResourceType.bte:  51>
ute = <ResourceType.ute:  52>
btd = <ResourceType.btd:  53>
utd = <ResourceType.utd:  54>
btp = <ResourceType.btp:  55>
utp = <ResourceType.utp:  56>
dft = <ResourceType.dft:  57>
gic = <ResourceType.gic:  58>
```

```
gui = <ResourceType.gui:  59>
css = <ResourceType.css:  60>
ccs = <ResourceType.ccs:  61>
btm = <ResourceType.btm:  62>
utm = <ResourceType.utm:  63>
dwk = <ResourceType.dwk:  64>
pwk = <ResourceType.pwk:  65>
btg = <ResourceType.btg:  66>
utg = <ResourceType.utg:  67>
jrl = <ResourceType.jrl:  68>
sav = <ResourceType.sav:  69>
utw = <ResourceType.utw:  70>
fourpc = <ResourceType.fourpc:  71>
ssf = <ResourceType.ssf:  72>
hak = <ResourceType.hak:  73>
nwm = <ResourceType.nwm:  74>
bik = <ResourceType.bik:  75>
ndb = <ResourceType.ndb:  76>
ptm = <ResourceType.ptm:  77>
ptt = <ResourceType.ptt:  78>
bak = <ResourceType.bak:  79>
dat = <ResourceType.dat:  80>
shd = <ResourceType.shd:  81>
xbc = <ResourceType.xbc:  82>
wbm = <ResourceType.wbm:  83>
mtr = <ResourceType.mtr:  84>
ktx = <ResourceType.ktx:  85>
ttf = <ResourceType.ttf:  86>
sql = <ResourceType.sql:  87>
tml = <ResourceType.tml:  88>
sq3 = <ResourceType.sq3:  89>
```

lod = <ResourceType.lod: 90>

gif = <ResourceType.gif: 91>

png = <ResourceType.png: 92>

jpg = <ResourceType.jpg: 93>

caf = <ResourceType.caf: 94>

ids = <ResourceType.ids: 95>

erf = <ResourceType.erf: 96>

bif = <ResourceType.bif: 97>

key = <ResourceType.key: 98>

class rollnw.**Saves**

An objects saves

fort: int

reflex: int

will: int

class rollnw.**Sound**

Class that represents a Sound object

active: bool

common: *Common*

continuous: bool

distance_max: float

distance_min: float

elevation: float

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

generated_type: int

hours: int

interval: int

interval_variation: int

json_archive_version: ClassVar[int]

looping: bool

object_type: ClassVar[int]

pitch_variation: float

positional: bool

priority: int

random: bool

random_position: bool

random_x: float

random_y: float

sounds: List[str]

times: int

volume: int

volume_variation: int

class rollnw.**SpawnCreature**

Encounter creature spawn

appearance: int

cr: int

resref: str

single_spawn: bool

class rollnw.**SpawnPoint**

A spawn point

orientation: *Vector3*

position: *Vector3*

class rollnw.**SpecialAbility**

Special Ability

flags: *SpellFlags*

level: int

spell: int

class rollnw.**SpellBook**

Implements a spell casters spellbook

add_known_spell(*level: int*, *entry:* SpellEntry)

Adds a known spell at level

add_memorized_spell(*level: int*, *entry:* SpellEntry)

Adds a memorized spell at level

get_known_spell(*level: int*, *index: int*)

Gets a known spell entry

get_known_spell_count(*level: int*)

Gets the number of known at a given level

get_memorized_spell(*level: int*, *index: int*)

Gets a memorized spell entry

get_memorized_spell_count(*level: int*)

Gets the number of memorized at a given level

remove_known_spell(*level: int*, *entry:* SpellEntry)

Removes a known spell entry

remove_memorized_spell(*level: int*, *entry:* SpellEntry)

Removes a memorized spell entry

class rollnw.**SpellEntry**

An entry in a spellbook

flags: *SpellFlags*

meta: *SpellMetaMagic*

spell: int

flag rollnw.**SpellFlags**(*value*)

Valid values are as follows:

readied = <SpellFlags.readied: 1>

spontaneous = <SpellFlags.spontaneous: 2>

unlimited = <SpellFlags.unlimited: 4>

flag rollnw.**SpellMetaMagic**(*value*)

Valid values are as follows:

empower = <SpellMetaMagic.empower: 1>

extend = <SpellMetaMagic.extend: 2>

maximize = <SpellMetaMagic.maximize: 4>

quicken = <SpellMetaMagic.quicken: 8>

silent = <SpellMetaMagic.silent: 16>

still = <SpellMetaMagic.still: 32>

class rollnw.**Store**

Class that represents a Store object

armor: *Inventory*

blackmarket: bool

blackmarket_markdown: int

static from_dict(*value: dict*) → *Store*

Constructs object from python dict.

static from_file(*path: str*) → *Store*

Constructs object from file. The file can be JSON or Gff.

gold: int

identify_price: int

json_archive_version: ClassVar[int]

markdown: int

markup: int

max_price: int

miscellaneous: *Inventory*

object_type: ClassVar[int]

potions: *Inventory*

rings: *Inventory*

scripts: *StoreScripts*

weapons: *Inventory*

class rollnw.**StoreScripts**

A Store's scripts

on_closed: str

on_opened: str

class rollnw.**Tile**

Area tile

animloop1: int

animloop2: int

animloop3: int

height: int

id: int

mainlight1: int

mainlight2: int

orientation: int

srclight1: int

srclight2: int

class rollnw.**Tlk**(*init: str* | LanguageID)

Implementation of the TLK file format

Parameters

init (*str* | LanguageID) – if passed a string, `init` will be treated as a path to a TLK file, if passed a LanguageID, default constructs with the TLKs language set to `init`.

get(*strref: int*) → str

Gets a tlk entry.

language_id()

Gets the language ID

modified()

Is Tlk modfied

save()

Writes TLK to file

save_as(*path: str*)

Writes TLK to file

set(*strref: int*, *string: str*)

Sets a localized string

size()

Gets the highest set strref

valid()

Gets if successfully parsed

class rollnw.**Trap**

Class representing a trap on an object

detect_dc: **int**

detectable: **bool**

disarm_dc: **int**

disarmable: **bool**

is_trapped: **bool**

one_shot: **bool**

type: **int**

class rollnw.**Trigger**

Class that represents a Trigger object

cursor: **int**

faction: **int**

static from_dict(*value: dict*) → *Trigger*

Constructs object from python dict.

static **from_file**(*path: str*) → *Trigger*

Constructs object from file. The file can be JSON or Gff.

geometry: List[*Vector3*]

highlight_height: float

linked_to: str

linked_to_flags: int

loadscreen: int

portrait: int

scripts: *TriggerScripts*

trap: *Trap*

type: int

class rollnw.**TriggerScripts**

A trigger's scripts

on_click: str

on_disarm: str

on_enter: str

on_exit: str

on_heartbeat: str

on_trap_triggered: str

on_user_defined: str

class rollnw.**TwoDA**(*filename: str*)

Implementation of 2da file format

Parameters

filename (*str*) – 2da file to load

get(*row: int*, *column: int | str*)

Gets a TwoDA value

Parameters

- **row** (*int*) – Row number
- **column** (*int | str*) – Column number or label

Returns

An int | float | string depending on the underlying value

set(*row: int*, *column: int | str*, *value: int | float | str*)

Sets a TwoDA value

Parameters

- **row** (*int*) – Row number

- **column** (*int | str*) – Column number or label
- **value** (*int | float | str*) – New value

class rollnw.**Vector2**

x: float

y: float

class rollnw.**Vector3**

x: float

y: float

z: float

class rollnw.**Vector4**

W: float

x: float

y: float

z: float

class rollnw.**Waypoint**

Class that represents a Waypoint object

appearance: int

description: *LocString*

static from_dict(*value: dict*)

Constructs object from python dict.

static from_file(*path: str*)

Constructs object from file. The file can be JSON or Gff.

has_map_note: bool

Has a map note

linked_to: str

Tag of linked object

map_note: *LocString*

map_note_enabled: bool

class rollnw.**Zip**(*path: str*)

Implementation of Zip file format as a container

Parameters

path (*str*) – Path to zip file

rollnw.**decode_plt_color**(*plt:* Plt, *colors:* PltColors, *x: int*, *y: int*) → List[int]

Decodes PLT and user selected colors to RBGA

`rollnw.resmatch(res: Resource, pattern: str) → bool`

Analog of fnmatch but for resource names

Parameters

- **res** (Resource) – Resource name
- **pattern** (*str*) – glob pattern

6.19 rollnw.kernel

class `rollnw.kernel.Config`

Configuration service

initialize(*options: ConfigOptions*)

Initialize config system

install_path() → str

Gets game install path

options() → *ConfigOptions*

Gets config options

set_paths(*install: str*, *user: str*)

Sets game paths

Note: Must be called before `initialize`

set_version(*version: GameVersion*)

Sets game paths

Note: Must be called before `initialize`

user_path() → str

Gets game install path

class `rollnw.kernel.ConfigOptions`

Configuration options

include_install: bool = True

If true, load base game data.

include_nwsync: bool = True

If true, load NWSync data.

include_user: bool = True

If true, load user data.

class `rollnw.kernel.EffectSystem`

add_effect(*type*, *apply*, *remove*)

Adds an effect type to the registry

add_itemprop(*type*, *generator*)

Adds an item property type to the registry

apply(*obj:* ObjectBase, *effect:* Effect) → bool

Applies an effect to an object

create(*type*) → *Effect*

Creates an effect

destroy(*effect:* Effect) → None

Destroys an effect

effect_limits_ability() → Tuple[int, int]

Gets ability effect minimum and maximum

effect_limits_armor_class() → Tuple[int, int]

Gets armor class effect minimum and maximum

effect_limits_attack() → Tuple[int, int]

Gets attack effect minimum and maximum

effect_limits_skill() → Tuple[int, int]

Gets skill effect minimum and maximum

ip_cost_table(*table: int*) → *TwoDA* | None

Gets an item property cost table

ip_definition(*type*)

Gets an item property definition

ip_param_table(*table: int*) → *TwoDA* | None

Gets an item property param table

remove(*obj:* ObjectBase, *effect:* Effect) → bool

Removes an effect to an object

set_effect_limits_ability(*min: int*, *max: int*) → None

Sets ability effect minimum and maximum

set_effect_limits_armor_class(*min: int*, *max: int*) → None

Sets armor class effect minimum and maximum

set_effect_limits_attack(*min: int*, *max: int*) → None

Sets attack effect minimum and maximum

set_effect_limits_skill(*min: int*, *max: int*) → None

Sets skill effect minimum and maximum

stats() → *EffectSystemStats*

Gets stats regarding the effect system

class rollnw.kernel.**EffectSystemStats**

Effect system stat data

free_list_size: int

pool_size: int

class rollnw.kernel.**Objects**

The object system creates, serializes, and deserializes entities

area(*resref: str*) → *Area*

creature(*resref: str*) → *Creature*

destroy(*obj:* ObjectHandle) → None

Destroys an object and removes it from object system

door(*resref: str*) → *Door*

encounter(*resref: str*) → *Encounter*

get(*handle:* ObjectHandle)

Gets an object by its handle

get_by_tag(*tag: str*, *nth: int = 0*) → *ObjectBase* | None

Gets an object with specific tag

placeable(*resref: str*) → *Placeable*

store(*resref: str*) → *Store*

trigger(*resref: str*) → *Trigger*

valid(*handle:* ObjectHandle) → bool

Checks if an object handle is still valid

waypoint(*resref: str*) → *Waypoint*

class rollnw.kernel.**Resources**(*parent:* Resources | *None*)

Resources service

class rollnw.kernel.**Rules**

Rules service

class rollnw.kernel.**Strings**

Strings service

class rollnw.kernel.**TwoDACache**

2da cache

get(*name: str* | Resource) → *TwoDA* | None

Gets a cached twoda

rollnw.kernel.**config**()

Gets config service

rollnw.kernel.**effects**()

Gets effects service

rollnw.kernel.**load_module**(*path: str*, *manifest: str*) → *Module*

Loads a module

Parameters

- **path** (*str*) – path to module, can be a directory (with module.ifo), a mod file, or a zip file
- **manifest** (*str*) – NWSynch manifest hash

rollnw.kernel.**objects**()

Gets objects service

`rollnw.kernel.resman()` → *Resources*

Gets resman service

`rollnw.kernel.rules()`

Gets rules service

`rollnw.kernel.start(options: ConfigOptions | None)`

Starts kernel services

Parameters

config (*rollnw.ConfigOptions | None*) – Optionally pass in configuration. Default behavior is to search for whatever NWN(:EE) install that it can find

`rollnw.kernel.strings()`

Gets strings service

`rollnw.kernel.unload_module()` → None

Unloads the currently loaded module

6.20 rollnw.model

class `rollnw.model.Mdl`

Implementation of ASCII Mdl file format

static from_file(*path*)

Loads mdl file from file path

model: *MdlModel*

The parsed model

valid()

Determines if file was successfully parsed

class `rollnw.model.MdlAABBEntry`

AABB Entry

bmax: *Vector3*

bmin: *Vector3*

leaf_face: int

plane: int

class `rollnw.model.MdlAABBNode`

AABB model node

entries: List[*MdlAABBEntry*]

class `rollnw.model.MdlAnimation`

Class containing model animation

anim_root: str

events: List[*MdlAnimationEvent*]

length: float

transition_time: float

class rollnw.model.**MdlAnimationEvent**

Animation Event

name: str

time: float

class rollnw.model.**MdlAnimeshNode**

Animated mesh node

animtverts: List[*Vector3*]

animverts: List[*Vector3*]

sampleperiod: float

class rollnw.model.**MdlCameraNode**

Camera node

enum rollnw.model.**MdlClassification**(*value*)

Model classes

Member Type
int

Valid values are as follows:

invalid = <MdlClassification.invalid: 1>

effect = <MdlClassification.effect: 2>

tile = <MdlClassification.tile: 3>

character = <MdlClassification.character: 4>

door = <MdlClassification.door: 5>

item = <MdlClassification.item: 6>

gui = <MdlClassification.gui: 7>

class rollnw.model.**MdlControllerKey**

Model controller

columns: int

data_offset: int

is_key: bool

key_offset: int

name: str

rows: int

time_offset: int

type: int

class rollnw.model.**MdlControllerType**

Controller types

alpha: ClassVar[int]

alpha_end: ClassVar[int]

alpha_mid: ClassVar[int]

alpha_start: ClassVar[int]

birthrate: ClassVar[int]

blur_length: ClassVar[int]

bounce_co: ClassVar[int]

color: ClassVar[int]

color_end: ClassVar[int]

color_mid: ClassVar[int]

color_start: ClassVar[int]

combine_time: ClassVar[int]

detonate: ClassVar[int]

drag: ClassVar[int]

fps: ClassVar[int]

frame_end: ClassVar[int]

frame_start: ClassVar[int]

grav: ClassVar[int]

life_exp: ClassVar[int]

lightning_delay: ClassVar[int]

lightning_radius: ClassVar[int]

lightning_scale: ClassVar[int]

lightning_subdiv: ClassVar[int]

mass: ClassVar[int]

multiplier: ClassVar[int]

orientation: ClassVar[int]

p2p_bezier2: ClassVar[int]

p2p_bezier3: ClassVar[int]

particle_rot: ClassVar[int]

percent_end: ClassVar[int]

percent_mid: ClassVar[int]

percent_start: ClassVar[int]

position: ClassVar[int]

radius: ClassVar[int]

rand_vel: ClassVar[int]

scale: ClassVar[int]

self_illum_color: ClassVar[int]

shadow_radius: ClassVar[int]

size_end: ClassVar[int]

size_end_y: ClassVar[int]

size_mid: ClassVar[int]

size_mid_y: ClassVar[int]

size_start: ClassVar[int]

size_start_y: ClassVar[int]

spread: ClassVar[int]

threshold: ClassVar[int]

velocity: ClassVar[int]

vertical_displacement: ClassVar[int]

wirecolor: ClassVar[int]

xsize: ClassVar[int]

ysize: ClassVar[int]

class rollnw.model.**MdlDanglymeshNode**

constraints: List[float]

displacement: float

period: float

tightness: float

class rollnw.model.**MdlDummyNode**

Dummy node

class rollnw.model.**MdlEmitterNode**

Emitter node

blastlength: float

blastradius: float

blend: str

blend_sel: int

chunkname: str

deadspace: float

flags: int

loop: int

opacity: float

p2p_type: str

render: str

render_sel: int

renderorder: int

spawntype: int

spawntype_sel: int

texture: str

twosidedtex: int

update: str

update_sel: int

xgrid: int

ygrid: int

class rollnw.model.**MdlFace**

Model face

material_idx: int

shader_group_idx: int

tvert_idx: List[int]

vert_idx: List[int]

class rollnw.model.**MdlGeometry**

Class containing model geometry

name: str

type: int

enum rollnw.model.**MdlGeometryFlag**(*value*)

Geometry flags

Member Type
int

Valid values are as follows:

geometry = <MdlGeometryFlag.geometry: 1>

model = <MdlGeometryFlag.model: 2>

animation = <MdlGeometryFlag.animation: 3>

binary = <MdlGeometryFlag.binary: 4>

enum rollnw.model.**MdlGeometryType**(*value*)

Geometry types

Member Type
int

Valid values are as follows:

geometry = <MdlGeometryType.geometry: 1>

model = <MdlGeometryType.model: 2>

animation = <MdlGeometryType.animation: 3>

class rollnw.model.**MdlLightNode**

Light node

affectdynamic: int

ambientonly: int

color: *Vector3*

dynamic: bool

fadinglight: int

flarecolorshifts: List[*Vector3*]

flarepositions: List[float]

flareradius: float

flaresizes: List[float]

generateflare: int

lensflares: float

lightpriority: int

multiplier: float

shadow: int

textures: **List[str]**

class rollnw.model.**MdlModel**

A parsed model

animation_count()

Gets the number of animations

animations()

Gets an iterator of animations

animationscale: **float**

bmax: *Vector3*

bmin: *Vector3*

classification: **int**

file_dependency: **str**

get_animation(*index: int*)

Gets an animation

ignorefog: **bool**

radius: **float**

supermodel: *Mdl* **| None**

supermodel_name: **str**

class rollnw.model.**MdlNode**

Base Model Node

children: **List[*MdlNode*]**

get_controller(*type: int*, *is_key: bool*) → Tuple[*MdlControllerKey*, List[float], List[float]]

Gets a controller key and times and key data

> **Note:** If not an animation, time will be empty

inheritcolor: **bool**

name: **str**

parent: *MdlNode*

type: **int**

class rollnw.model.**MdlNodeFlags**

Model node flags

aabb: **ClassVar[int]**

anim: **ClassVar[int]**

camera: **ClassVar[int]**

dangly: ClassVar[int]

emitter: ClassVar[int]

header: ClassVar[int]

light: ClassVar[int]

mesh: ClassVar[int]

patch: ClassVar[int]

reference: ClassVar[int]

skin: ClassVar[int]

class rollnw.model.**MdlNodeType**

Model node types

aabb: ClassVar[int]

animmesh: ClassVar[int]

camera: ClassVar[int]

danglymesh: ClassVar[int]

dummy: ClassVar[int]

emitter: ClassVar[int]

light: ClassVar[int]

patch: ClassVar[int]

reference: ClassVar[int]

skin: ClassVar[int]

trimesh: ClassVar[int]

class rollnw.model.**MdlPatchNode**

Patch node

class rollnw.model.**MdlReferenceNode**

Reference node

reattachable: bool

refmodel: str

class rollnw.model.**MdlSkinNode**

Skin mesh node

vertices: List[*SkinVertex*]

List of vertex positions, texcoords, normals, tangents

enum rollnw.model.**MdlTriangleMode**(*value*)

Triangle mode

Member Type
int

Valid values are as follows:

triangle = <MdlTriangleMode.triangle: 1>

triangle_strip = <MdlTriangleMode.triangle_strip: 2>

class rollnw.model.**MdlTrimeshNode**

Trimesh Node

ambient: *Vector3*

beaming: bool

bitmap: str

bmax: *Vector3*

bmin: *Vector3*

center: *Vector3*

diffuse: *Vector3*

displtype: int

indices: List[int]

List of vertex indices

lightmapped: int

materialname: str

multimaterial: List[str]

render: bool

renderhint: str

rotatetexture: bool

shadow: bool

shininess: float

showdispl: bool

specular: *Vector3*

textures: List[str]

tilefade: int

transparencyhint: int

vertices: List[*Vertex*]

List of vertex positions, texcoords, normals, tangents

class rollnw.model.**ModelEmitterFlag**

Emitter flags

affected_by_wind: ClassVar[int]

bounce: ClassVar[int]

inherit: ClassVar[int]

inherit_local: ClassVar[int]

inherit_part: ClassVar[int]

inherit_vel: ClassVar[int]

is_tinted: ClassVar[int]

p2p: ClassVar[int]

p2p_sel: ClassVar[int]

random: ClassVar[int]

splat: ClassVar[int]

class rollnw.model.**SkinVertex**

Skin Vertex data

bones: *IVector4*

normal: *Vector3*

position: *Vector3*

tangent: *Vector4*

tex_coords: *Vector2*

weights: *Vector4*

class rollnw.model.**Vertex**

Vertex data

normal: *Vector3*

position: *Vector3*

tangent: *Vector3*

tex_coords: *Vector2*

6.21 rollnw.nwn1

`rollnw.nwn1.attacks_per_second(obj: Creature, type, versus: ObjectBase) → float`

Number of attacks per second

`rollnw.nwn1.base_attack_bonus(obj: Creature) → int`

Calculates base attack bonus

`rollnw.nwn1.calculate_ac_versus(obj: ObjectBase, versus: ObjectBase | None = None, is_touch_attack: bool = False) → int`

Calculate Armor Class versus another object

`rollnw.nwn1.calculate_item_ac(obj: Item) → int`

Calculates the armor class of a piece of armor

`rollnw.nwn1.can_equip_item(obj: Creature, item: Item, slot: int)`

Determines if an item can be equipped

`rollnw.nwn1.can_use_monk_abilities(obj: Creature) → Tuple[bool, int]`

Determines if monk class abilities are usable and monk class level

`rollnw.nwn1.effect_ability_modifier(ability, modifier) → Effect`

Creates an ability modifier effect

`rollnw.nwn1.effect_armor_class_modifier(type, modifier) → Effect`

Creates an armor class modifier effect

`rollnw.nwn1.effect_attack_modifier(attack, modifier) → Effect`

Creates an attack modifier effect

`rollnw.nwn1.effect_haste() → Effect`

Creates a haste effect

`rollnw.nwn1.effect_skill_modifier(skill, modifier) → Effect`

Creates an skill modifier effect

`rollnw.nwn1.equip_index_to_attack_type(equip)`

Converts an equip index to an attack type

`rollnw.nwn1.equip_item(obj: Creature, item: Item, slot: int)`

Equip an item

`rollnw.nwn1.get_ability_modifier(obj: Creature, ability, base: bool = False) → int`

Gets creatures ability modifier

`rollnw.nwn1.get_ability_score(obj: Creature, ability, base: bool = False) → int`

Gets creatures ability score

`rollnw.nwn1.get_caster_level(obj: Creature, class_: int) → int`

Gets creatures caster level

`rollnw.nwn1.get_dex_modifier(obj: Creature) → int`

Gets creatures dexterity modifier as modified by armor, etc.

`rollnw.nwn1.get_equipped_item(obj: Creature, slot)`

Gets an equipped item

`rollnw.nwn1.get_skill_rank(obj: Creature, skill, versus=None, base=False)`

Determines creatures skill rank

`rollnw.nwn1.get_spell_dc(obj: Creature, class_: int, spell: int) → int`

Gets spell DC

`rollnw.nwn1.get_weapon_by_attack_type(obj: Creature, type) → Item`

Gets an equipped weapon by attack type

`rollnw.nwn1.is_flanked(target: Creature, attacker: Creature) → bool`

`rollnw.nwn1.is_ranged_weapon(item: Item) → bool`

Determines if weapon is ranged

`rollnw.nwn1.is_shield(baseitem) → bool`

Determines if item is a shield

`rollnw.nwn1.itemprop_ability_modifier(ability, modifier) → ItemProperty`

Creates ability modifier item property

`rollnw.nwn1.itemprop_armor_class_modifier(value) → ItemProperty`

Creates armor modifier item property

`rollnw.nwn1.itemprop_attack_modifier(value) → ItemProperty`

Creates attack modifier item property

`rollnw.nwn1.itemprop_enhancement_modifier(value) → ItemProperty`

Creates enhancement modifier item property

`rollnw.nwn1.itemprop_haste() → ItemProperty`

Creates haste item property

`rollnw.nwn1.itemprop_skill_modifier(skill, modifier) → ItemProperty`

Creates skill modifier item property

`rollnw.nwn1.queue_remove_effect_by(obj: ObjectBase, creator: ObjectHandle)`

Queues remove effect events by effect creator

`rollnw.nwn1.resolve_attack(obj: Creature, type, versus: ObjectBase)`

Resolves an attack

`rollnw.nwn1.resolve_attack_bonus(obj: Creature, type, versus: ObjectBase | None = None) → int`

Calculates attack bonus

`rollnw.nwn1.resolve_attack_damage(obj: Creature, versus: ObjectBase, data: AttackData) → int`

Resolves damage from an attack

`rollnw.nwn1.resolve_concealment(obj: ObjectBase, type, target: ObjectBase, vs_ranged: bool) → Tuple[int, bool]`

Resolves an concealment - i.e. the highest of concealment and miss chance

`rollnw.nwn1.resolve_critical_multiplier(obj: Creature, type, versus: ObjectBase | None = None) → int`

Resolves critical multiplier

`rollnw.nwn1.resolve_critical_threat(obj: Creature, type) → int`

Resolves critical multiplier

`rollnw.nwn1.resolve_damage_immunity(obj: ObjectBase, dmg_type, versus: ObjectBase | None = None) → int`

Resolves damage immunity

`rollnw.nwn1.resolve_damage_modifiers(obj: Creature, versus: ObjectBase, data: AttackData) → None`

Resolves resistance, immunity, and reduction

`rollnw.nwn1.resolve_damage_reduction(obj: ObjectBase, power: int, versus: ObjectBase | None = None) → Tuple[int, Effect]`

Resolves damage reduction

`rollnw.nwn1.resolve_damage_resistance(obj: ObjectBase, dmg_type, versus: ObjectBase | None = None) → Tuple[int, Effect]`

Resolves damage resistance

`rollnw.nwn1.resolve_dual_wield_penalty(obj: Creature) → Tuple[int, int]`

Resolves dual wield attack bonus penalty

`rollnw.nwn1.resolve_iteration_penalty(obj: Creature, attack_type)`

Resolves iteration attack bonus penalty

`rollnw.nwn1.resolve_number_of_attacks(obj: Creature, offhand: bool = False) → Tuple[int, int]`

Calculates number of attacks

`rollnw.nwn1.resolve_target_state(obj: Creature, versus: ObjectBase)`

Resolves damage from an attack

`rollnw.nwn1.resolve_unarmed_damage(obj: Creature) → DiceRoll`

Resolves unarmed damage

`rollnw.nwn1.resolve_weapon_damage(obj: Creature, weapon: Item) → DiceRoll`

Resolves weapon damage

`rollnw.nwn1.resolve_weapon_power(obj: Creature, weapon: Item) → int`

Resolves weapon power

`rollnw.nwn1.unequip_item(obj: Creature, slot: int)`

Unequips an item

`rollnw.nwn1.weapon_is_finessable(obj: Creature, weapon: Item) → bool`

Determines if a weapon is finessable

`rollnw.nwn1.weapon_iteration(obj: Creature, weapon: Item) → int`

Calculates weapon iteration, e.g. 5 or 3 for monk weapons

6.22 rollnw.script

class `rollnw.script.AssignExpression`

Assignment operation expression

lhs: *VariableExpression* | *DotExpression*

Expression being assigned to. Note that in a simple language like NWScript this can only be a variable expression or a dot expression (i.e. assigning a struct member)

operator: *NssToken*

The assignment operator, '=', '+=', etc, etc.

rhs: *Expression*

The expression being assigned

class rollnw.script.**Ast**

Class containing a parsed ast

__getitem__(*index: int*) → *Declaration*

Gets a toplevel declaration

__iter__() → Iterator[*Declaration*]

Gets an iterator of toplevel declarations

__len__() → int

Gets number of toplevel declarations

comments() → List[*Comment*]

Gets all comments in Ast

defines: **dict[str, str]**

Defines from `#define` directive. Only used in command script, i.e. nwscript.nss

find_comment(*line*) → str

Finds first comment that the source range of which ends on `line` or `line` - 1

includes: **List[*Include*]**

Scripts that are included in the current script

class rollnw.script.**AstNode**

Base Ast Node class

complete(*needle: str*) → List[*Symbol*]

Find completions for any Ast Node

@note This function does not traverse dependencies

class rollnw.script.**BinaryExpression**

Binary operation expression

lhs: *Expression*

Lefthand side of the binary expression

operator: *NssToken*

The binary operator, '+', '-', etc, etc.

rhs: *Expression*

Righthand side of the binary expression

class rollnw.script.**BlockStatement**

Block statement

range

Range in source code

Type

SourceRange

__getitem__(*idx: int*) → *Statement*

Gets a statement in the block

__iter__() → Iterator[*Statement*]

Gets iterator of statements

__len__() → int

Gets the number of statements

class rollnw.script.**CallExpression**

Call operation expression

__getitem__(*idx: int*) → *Expression*

Gets an argument

__iter__() → Iterator[*Expression*]

Gets iterator of arguments

__len__() → int

Gets the number of arguments

expr: *Expression*

The expression prior to `(...)`

class rollnw.script.**Comment**

Abstracts Comment

__str__() → str

Return str(self).

class rollnw.script.**ComparisonExpression**

Comparison operation expression

lhs: *Expression*

Lefthand side of the Comparison expression

operator: *NssToken*

The Comparison operator, '==', '<', etc, etc.

rhs: *Expression*

Righthand side of the Comparison expression

class rollnw.script.**ConditionalExpression**

Conditional operation expression

false_branch: *Statement*

The branch where `test` is False

test: *Expression*

The expression that is tested

true_branch: *Statement*

The branch where `test` is True

class rollnw.script.**Context**(*include_paths: List[str] = []*, *command_script: str = 'nwscript'*)

Provides a context for parsing a NWScript file

Every context contains its own resource manager that has as a parent the global resource manager. Ultimately, this will be changed to each context having its own unique resource manager.

Parameters

- **include_paths** (*[str], optional*) – A list of include paths to load into internal resource manager. Default: `[]`.
- **command_script** (*str, optional*) – Command script to load. Default: "nwscript".

__init__(*include_paths: List[str] = []*, *command_script: str = 'nwscript'*)

add_include_path(*path: str*)

Adds path to internal resman

command_script() → *Nss* | None

Gets the command script for the current context

get(*resref: str*, *is_command_script: bool = False*) → *Nss* | None

Gets a script from the context's internal resman

class rollnw.script.**DeclList**

__getitem__(*idx: int*) → *Declaration*

Gets a declaration

__iter__() → Iterator[*Declaration*]

Gets iterator of statements

__len__() → int

Gets the number of declarations

class rollnw.script.**Declaration**

Base Declaration class type

identifier() → str

Get declaration identifier

class rollnw.script.**Diagnostic**

Information for a script diagnostic

location: *SourceRange*

Source range in script

message: **str**

A helpful message

script: **str**

Name of script

severity: *DiagnosticSeverity*

The severity of the diagnostic

type: *DiagnosticType*

The type of the diagnostic

enum rollnw.script.**DiagnosticSeverity**(*value*)

Member Type

`int`

Valid values are as follows:

error = <DiagnosticSeverity.error: 1>

hint = <DiagnosticSeverity.hint: 2>

information = <DiagnosticSeverity.information: 3>

warning = <DiagnosticSeverity.warning: 4>

enum rollnw.script.**DiagnosticType**(*value*)

Member Type
int

Valid values are as follows:

lexical = <DiagnosticType.lexical: 1>

parse = <DiagnosticType.parse: 2>

semantic = <DiagnosticType.semantic: 3>

class rollnw.script.**DoStatement**

Do statement

block: *BlockStatement*

The do block statement

test: *Expression*

The test at the end of the block

class rollnw.script.**DotExpression**

Dot operation expression

lhs: *VariableExpression* | *CallExpression*

In NWScript the only two possible expressions on the left hand of the dot are var_expr.var_expr or call_expr.var_expr

rhs: *VariableExpression*

The right hand side of a dot operator

class rollnw.script.**EmptyExpression**

Empty expression only used in case of expression parsing erros

class rollnw.script.**EmptyStatement**

Empty statement

class rollnw.script.**ExprStatement**

Expression statement

expr: *Expression*

An expression

class rollnw.script.**Expression**

Base Expression AST node

class rollnw.script.**ForStatement**

For statement

block: *Statement*

While this is called `block`, any (single) statement can follow a for loop.

increment: *Expression* | None

An optional increment expression

init: *AstNode* | None

An optional initialization. Normally this is a Declaration or just an expression

test: *Expression* | None

An optional expression that determines if the loop is to continue

class rollnw.script.**FunctionDecl**

Function declaration

__getitem__(*idx: int*) → *Declaration*

Gets a parameter

__iter__() → Iterator[*Declaration*]

Gets iterator of parameters

__len__() → int

Gets the number of parameters

class rollnw.script.**FunctionDefinition**

Function definition

block: *BlockStatement*

Block of the function

decl: *FunctionDecl*

Declaration of the function definition

class rollnw.script.**GroupingExpression**

Grouping operation expression

expr: *Expression*

Expression contained in the grouping parenthesis.

class rollnw.script.**IfStatement**

If statement

false_branch: *Statement*

The optional branch where `test` is False

test: *Expression*

The expression that is tested

true_branch: *Statement*

The branch where `test` is True

class rollnw.script.**Include**

Abstracts a script include

location: *SourceRange*

Source range in script

resref: str

Resref of included script

script: *Nss*

Loaded script

used: **int**

Number of times include is used in script file

class rollnw.script.**InlayHint**

An inlay source code hint for an LSP

message: **str**

Helpful message to display inline or a type, etc.

position: *SourcePosition*

The postion where the hint should be displayed

class rollnw.script.**JumpStatement**

Jump statement

expr: *Expression* | **None**

Optional expression when returning a value

operator: *NssToken*

Token representing the jump statement (i.e. `return`, `break`, `continue`)

class rollnw.script.**LabelStatement**

Label statement

expr: *Expression* | **None**

Expression when label is a `case`.

label: *NssToken*

Token representing the label statement (i.e. `case`, `default`)

class rollnw.script.**LiteralExpression**

Literal expression

data: **int** | **str** | **float** | *Location*

Data of the literal value

literal: *NssToken*

Token of the literal value

class rollnw.script.**LiteralVectorExpression**

Literal vector expression

x: *NssToken*

Token representation for `x` value

y: *NssToken*

Token representation for `y` value

z: *NssToken*

Token representation for `z` value

class rollnw.script.**LogicalExpression**

Logical operation expression

lhs: *Expression*

Lefthand side of the logical expression

operator: *NssToken*

The logical operator, '||', '&&', etc, etc.

rhs: *Expression*

Righthand side of the logical expression

class rollnw.script.**Nss**(*path: str*, *ctx:* Context, *is_command_script: bool = False*)

Implementation of nwscript

__init__(*path: str*, *ctx:* Context, *is_command_script: bool = False*)

Constructs Nss object

ast() → *Ast*

Gets the parsed script

complete(*needle: str*) → List[*Symbol*]

Generates a list of potential completions (excluding dependencies)

complete_at(*needle: str*, *line: int*, *character: int*) → List[*Symbol*]

Get all completions (including dependencies)

complete_dot(*needle: str*, *line: int*, *character: int*) → List[*Symbol*]

Get all completions for struct fields

diagnostics() → List[*Diagnostic*]

errors() → int

Gets number of errors encountered while parsing

exports() → List[*Symbol*]

Gets all of the scripts exports, i.e. top level declarations

static from_string(*string: str*, *ctx:* Context, *is_command_script: bool = False*) → *Nss*

Loads Nss from string

locate_export(*is_type: bool*, *search_dependencies: bool = False*) → *Symbol*

Locate export, i.e. a top level symbols

locate_symbol(*symbol: str*, *line: int*, *character: int*) → *Symbol*

Locate symbol in source file

name() → str

Gets script's name

parse()

Parses the script

process_includes()

Process includes and dependencies

resolve()

Resolves and type-checks Ast

signature_help(*line: int*, *character: int*) → *SignatureHelp*

Gets signature help for a call expression that contains the provided position

view_from_range(*range: SourceRange*) → str

Gets string view of the source at `range`

warnings() → int

Gets number of errors encountered while parsing

class rollnw.script.**NssLexer**(*script: str*)

A nwscript lexer

__init__(*script: str*)

Constructs lexer from a string

current()

Gets next token

next()

Gets next token

class rollnw.script.**NssToken**

Nss token

loc: *SourceLocation*

The location of the token in a source file

type: *NssTokenType*

The type of the token

enum rollnw.script.**NssTokenType**(*value*)

Member Type
`int`

Valid values are as follows:

INVALID = <NssTokenType.INVALID: 1>

END = <NssTokenType.END: 2>

IDENTIFIER = <NssTokenType.IDENTIFIER: 3>

LPAREN = <NssTokenType.LPAREN: 4>

RPAREN = <NssTokenType.RPAREN: 5>

LBRACE = <NssTokenType.LBRACE: 6>

RBRACE = <NssTokenType.RBRACE: 7>

LBRACKET = <NssTokenType.LBRACKET: 8>

RBRACKET = <NssTokenType.RBRACKET: 9>

COMMA = <NssTokenType.COMMA: 10>

COLON = <NssTokenType.COLON: 11>

QUESTION = <NssTokenType.QUESTION: 12>

SEMICOLON = <NssTokenType.SEMICOLON: 13>

```
POUND = <NssTokenType.POUND: 14>

DOT = <NssTokenType.DOT: 15>

AND = <NssTokenType.AND: 16>

ANDAND = <NssTokenType.ANDAND: 17>

ANDEQ = <NssTokenType.ANDEQ: 18>

DIV = <NssTokenType.DIV: 19>

DIVEQ = <NssTokenType.DIVEQ: 20>

EQ = <NssTokenType.EQ: 21>

EQEQ = <NssTokenType.EQEQ: 22>

GT = <NssTokenType.GT: 23>

GTEQ = <NssTokenType.GTEQ: 24>

LT = <NssTokenType.LT: 25>

LTEQ = <NssTokenType.LTEQ: 26>

MINUS = <NssTokenType.MINUS: 27>

MINUSEQ = <NssTokenType.MINUSEQ: 28>

MINUSMINUS = <NssTokenType.MINUSMINUS: 29>

MOD = <NssTokenType.MOD: 30>

MODEQ = <NssTokenType.MODEQ: 31>

TIMES = <NssTokenType.TIMES: 32>

TIMESEQ = <NssTokenType.TIMESEQ: 33>

NOT = <NssTokenType.NOT: 34>

NOTEQ = <NssTokenType.NOTEQ: 35>

OR = <NssTokenType.OR: 36>

OREQ = <NssTokenType.OREQ: 37>

OROR = <NssTokenType.OROR: 38>

PLUS = <NssTokenType.PLUS: 39>

PLUSEQ = <NssTokenType.PLUSEQ: 40>

PLUSPLUS = <NssTokenType.PLUSPLUS: 41>

SL = <NssTokenType.SL: 42>

SLEQ = <NssTokenType.SLEQ: 43>

SR = <NssTokenType.SR: 44>
```

```
SREQ = <NssTokenType.SREQ: 45>

TILDE = <NssTokenType.TILDE: 46>

USR = <NssTokenType.USR: 47>

USREQ = <NssTokenType.USREQ: 48>

XOR = <NssTokenType.XOR: 49>

XOREQ = <NssTokenType.XOREQ: 50>

FLOAT_CONST = <NssTokenType.FLOAT_CONST: 51>

INTEGER_CONST = <NssTokenType.INTEGER_CONST: 52>

OBJECT_INVALID_CONST = <NssTokenType.OBJECT_INVALID_CONST: 53>

OBJECT_SELF_CONST = <NssTokenType.OBJECT_SELF_CONST: 54>

STRING_CONST = <NssTokenType.STRING_CONST: 55>

STRING_RAW_CONST = <NssTokenType.STRING_RAW_CONST: 56>

ACTION = <NssTokenType.ACTION: 57>

BREAK = <NssTokenType.BREAK: 58>

CASE = <NssTokenType.CASE: 59>

CASSOWARY = <NssTokenType.CASSOWARY: 60>

CONST = <NssTokenType.CONST: 61>

CONTINUE = <NssTokenType.CONTINUE: 62>

DEFAULT = <NssTokenType.DEFAULT: 63>

DO = <NssTokenType.DO: 64>

EFFECT = <NssTokenType.EFFECT: 65>

ELSE = <NssTokenType.ELSE: 66>

EVENT = <NssTokenType.EVENT: 67>

FLOAT = <NssTokenType.FLOAT: 68>

FOR = <NssTokenType.FOR: 69>

IF = <NssTokenType.IF: 70>

INT = <NssTokenType.INT: 71>

ITEMPROPERTY = <NssTokenType.ITEMPROPERTY: 72>

JSON = <NssTokenType.JSON: 73>

LOCATION = <NssTokenType.LOCATION: 74>

OBJECT = <NssTokenType.OBJECT: 75>
```

RETURN = <NssTokenType.RETURN: 76>

STRING = <NssTokenType.STRING: 77>

STRUCT = <NssTokenType.STRUCT: 78>

SQLQUERY = <NssTokenType.SQLQUERY: 79>

SWITCH = <NssTokenType.SWITCH: 80>

TALENT = <NssTokenType.TALENT: 81>

VECTOR = <NssTokenType.VECTOR: 82>

VOID = <NssTokenType.VOID: 83>

WHILE = <NssTokenType.WHILE: 84>

JSON_CONST = <NssTokenType.JSON_CONST: 85>

LOCATION_INVALID = <NssTokenType.LOCATION_INVALID: 86>

class rollnw.script.**PostfixExpression**

Postfix operation expression

lhs: *Expression*

Lefthand side of the postfix expression

operator: *NssToken*

The postix operator, '++', '–', etc.

class rollnw.script.**SignatureHelp**

Data required for providing Signature Help in an LSP

active_param: **int**

The currently active parameter, i.e. where the cursor is in the parameter

decl: *Declaration*

The declaration for `expr`

expr: *CallExpression*

The current call expression

class rollnw.script.**SourceLocation**

Nss source location

length() → int

Length of the source location

range: *SourceRange*

Range in source code

view() → str

String view of the location

class rollnw.script.**SourcePosition**

Position in source code

column: **int**

Starting column

line: **int**

Starting line

class rollnw.script.**SourceRange**

Range into the source code

end: *SourcePosition*

End

start: *SourcePosition*

Start

class rollnw.script.**Statement**

Base statement class

class rollnw.script.**StructDecl**

Struct declaration

__getitem__(*idx: int*) → *Declaration*

Gets a struct member declaration

__iter__() → Iterator[*Declaration*]

Gets iterator of statements

__len__() → int

Gets the number of struct members

class rollnw.script.**SwitchStatement**

Switch statement

block: *BlockStatement*

The block of labels and stuff

target: *Expression*

The target expression for the switch

class rollnw.script.**Symbol**

Info regarding a particular symbol somewhere in a source file

comment: **str**

Comment associated with the line the symbol is on or the line prior

decl: *Declaration*

The declaration of the symbol

kind: *SymbolKind*

The symbols kind, for use with an LSP

node: *AstNode* | **None**

The ast node where the symbol was found, if availble

provider: *Nss*

The script in which the symbol was found

type: **str**

The symbols type as a string

view: **str**

A string view of the symbol in source

enum rollnw.script.**SymbolKind**(*value*)

Enum of different symbol kinds

Member Type
int

Valid values are as follows:

variable = <SymbolKind.variable: 1>

function = <SymbolKind.function: 2>

type = <SymbolKind.type: 3>

param = <SymbolKind.param: 4>

field = <SymbolKind.field: 5>

class rollnw.script.**UnaryExpression**

Unary operation expression

operator: *NssToken*

The postix operator, '++', '–', etc.

rhs: *Expression*

Righthand side of the postfix expression

class rollnw.script.**VarDecl**

Variable declaration

init: *Expression* | None

An optional expression to initialize declaration

class rollnw.script.**VariableExpression**

Variable expression

var: *NssToken*

Token containing variable identifier

class rollnw.script.**WhileStatement**

While statement

block: *Statement*

While this is called `block`, any (single) statement can follow a for loop.

test: *Expression*

The expression that determines if the loop is to continue

6.23 are

```yaml
all:
  ChanceLightning: int
  ChanceRain: int
  ChanceSnow: int
  Comments: string
  Creator_ID: int
  DayNightCycle: byte
  Expansion_List: # Obsolete
    - {}
  Flags: dword
  FogClipDist: float
  Height: int
  ID: int
  IsNight: byte
  LightingScheme: byte
  LoadScreenID: word
  ModListenCheck: int
  ModSpotCheck: int
  MoonAmbientColor: dword
  MoonDiffuseColor: dword
  MoonFogAmount: byte
  MoonFogColor: dword
  MoonShadows: byte
  Name: locstring
  NoRest: byte
  OnEnter: resref
  OnExit: resref
  OnHeartbeat: resref
  OnUserDefined: resref
  PlayerVsPlayer: byte
  ResRef: resref
  ShadowOpacity: byte
  SkyBox: byte
  SunAmbientColor: dword
  SunDiffuseColor: dword
  SunFogAmount: byte
  SunFogColor: dword
  SunShadows: byte
  Tag: string
  Tile_List:
    - $struct_id: 1
      Tile_AnimLoop1: byte
      Tile_AnimLoop2: byte
      Tile_AnimLoop3: byte
      Tile_Height: int
      Tile_ID: int
      Tile_MainLight1: byte
      Tile_MainLight2: byte
      Tile_Orientation: int
      Tile_SrcLight1: byte
```

(continues on next page)

(continued from previous page)

```
      Tile_SrcLight2: byte
  Tileset: resref
  Version: dword
  Width: int
  WindPower: int
```

6.24 bic

```
# [TODO] - A lot

instances:
  $inherit: utc.yaml
  Age: int
  LvlStatList:
    - $struct_id: 0
      EpicLevel: byte
      FeatList:
        - $struct_id: 0
          Feat: word
      KnownList{0-9}?:
        - $struct_id: 0
          Spell: word
      LvlStatAbility: byte?
      LvlStatClass: byte
      LvlStatHitDie: byte
      SkillList:
        - $struct_id: 0
          Rank: byte
      SkillPoints: word
      # Spells
```

6.25 dlg

```
all:
  DelayEntry: dword
  DelayReply: dword
  EndConverAbort: resref
  EndConversation: resref
  EntryList:
    - AnimLoop: byte
      Animation: dword
      Comment: cexostr
      Delay: dword
      Quest: cexostr
      RepliesList:
        - $struct_id: index
          Active: resref
```

(continues on next page)

(continued from previous page)

```
          Index: dword
          IsChild: byte
          LinkComment: cexostr
      Script: resref
      Sound: resref
      Speaker: cexostr
      Text: cexolocstr
  NumWords: dword
  PreventZoomIn: byte
  ReplyList:
    - $struct_id: index
      AnimLoop: byte
      Animation: dword
      Comment: cexostr
      Delay: dword
      EntriesList:
        - $struct_id: index
          Active: resref
          Index: dword
          IsChild: byte
          LinkComment: cexostr
      Quest: cexostr
      QuestEntry: dword
      Script: resref
      Sound: resref
      Text: cexolocstr
  StartingList:
    - $struct_id: index
      Active: resref
      Index: dword
```

6.26 fac

```
FactionList:
  - $struct_id: index
    FactionGlobal: word
    FactionName: string
    FactionParentID: dword
RepList:
  - $struct_id: index
    FactionID1: dword
    FactionID2: dword
    FactionRep: dword
```

6.27 gic

```yaml
Creature List:
  - $struct_id: 4
    Comment: string
Door List:
  - $struct_id: 8
    Comment: string
Encounter List:
  - $struct_id: 7
    Comment: string
List:
  - $struct_id: 0
    Comment: string
Placeable List:
  - $struct_id: 9
    Comment: string
SoundList:
  - $struct_id: 6
    Comment: string
    PlayInToolset: byte
StoreList:
  - $struct_id: 11
    Comment: string
TriggerList:
  - $struct_id: 1
    Comment: string
WaypointList:
  - $struct_id: 5
    Comment: string
```

6.28 git

```yaml
AreaProperties:
  $struct_id: 100
  AmbientSndDay: int
  AmbientSndDayVol: int
  AmbientSndNight: int
  AmbientSndNitVol: int
  EnvAudio: int
  MusicBattle: int
  MusicDay: int
  MusicDelay: int
  MusicNight: int
Creature List:
  - $struct_id: 4
    utc.yml: instance
Door List:
  - $struct_id: 8
    utd.yml: instance
```

(continues on next page)

(continued from previous page)

```
Encounter List:
  - $struct_id: 7
    ute.yml: instance
List:
  - $struct_id: 0
    uti.yml: instance
Placeable List:
  - $struct_id: 9
    utp.yml: instance
SoundList:
  - $struct_id: 6
    uts.yml: instance
StoreList:
  - $struct_id: 11
    utm.yml: instance
TriggerList:
  - $struct_id: 1
    utt.yml: instance
VarTable: vartable.yml
WaypointList:
  - $struct_id: 5
    utw.yml: instance
```

6.29 ifo

```
all:
  Expansion_Pack: word
  Mod_Area_list:
    - $struct_id: 6
      Area_Name: resref
  Mod_CacheNSSList: # Obsolete
    - {}
  Mod_Creator_ID: int # Obsolete, always 2
  Mod_CustomTlk: string
  Mod_CutSceneList: # Obsolete
    - {}
  Mod_DawnHour: byte
  Mod_Description: locstring
  Mod_DuskHour: byte
  Mod_Entry_Area: resref
  Mod_Entry_Dir_X: float
  Mod_Entry_Dir_Y: float
  Mod_Entry_X: float
  Mod_Entry_Y: float
  Mod_Entry_Z: float
  Mod_Expan_List: # Obsolete
    - {}
  Mod_GVar_List: # Obsolete
    - {}
  Mod_HakList:
```

(continues on next page)

(continued from previous page)

```
    - $struct_id: 8
      Mod_Hak: string
  Mod_ID: void
  Mod_IsSaveGame: byte
  Mod_MinGameVer: string
  Mod_MinPerHour: byte
  Mod_Name: locstring
  Mod_OnAcquirItem: resref
  Mod_OnActvtItem: resref
  Mod_OnClientEntr: resref
  Mod_OnClientLeav: resref
  Mod_OnCutsnAbort: resref
  Mod_OnHeartbeat: resref
  Mod_OnModLoad: resref
  Mod_OnModStart: resref
  Mod_OnPlrChat: resref
  Mod_OnPlrDeath: resref
  Mod_OnPlrDying: resref
  Mod_OnPlrEqItm: resref
  Mod_OnPlrLvlUp: resref
  Mod_OnPlrRest: resref
  Mod_OnPlrUnEqItm: resref
  Mod_OnSpawnBtnDn: resref
  Mod_OnUnAqreItem: resref
  Mod_OnUsrDefined: resref
  Mod_StartDay: byte
  Mod_StartHour: byte
  Mod_StartMonth: byte
  Mod_StartMovie: resref
  Mod_StartYear: dword
  Mod_Tag: string
  Mod_UUID: string
  Mod_Version: dword
  Mod_XPScale: byte
  VarTable: vartable.yml
```

6.30 itp

```
# Note, this is not good

MAIN:
  - $struct_id: 1
    ID: byte
    LIST:
      - $struct_id: 1
        CR: float
        FACTION: string
        ID: byte
        LIST:
          - $struct_id: 1
```

(continues on next page)

(continued from previous page)

```
            NAME: string
            RESREF: resref
            STRREF: dword
        NAME: string
        RESREF: resref
        STRREF: dword
    STRREF: dword
```

6.31 jrl

```
Categories:
  - $struct_id: index
    Comment: cexostr
    EntryList:
      - $struct_id: index
        End: word
        ID: dword
        Text: cexolocstr
    Name: cexolocstr
    Picture: word
    Priority: dword
    Tag: cexostr
    XP: dword
```

6.32 utc

```
all:
  Appearance_Head: byte
  Appearance_Type: word
  ArmorPart_RFoot: byte
  BodyBag: byte
  BodyPart_Belt: byte
  BodyPart_LBicep: byte
  BodyPart_LFArm: byte
  BodyPart_LFoot: byte
  BodyPart_LHand: byte
  BodyPart_LShin: byte
  BodyPart_LShoul: byte
  BodyPart_LThigh: byte
  BodyPart_Neck: byte
  BodyPart_Pelvis: byte
  BodyPart_RBicep: byte
  BodyPart_RFArm: byte
  BodyPart_RHand: byte
  BodyPart_RShin: byte
  BodyPart_RShoul: byte
  BodyPart_RThigh: byte
```

(continues on next page)

(continued from previous page)

```
BodyPart_Torso: byte
CRAdjust: int
Cha: byte
ChallengeRating: float
ClassList:
  - Class: int
    ClassLevel: short
    KnownList0:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList1:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList2:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList3:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList4:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList5:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList6:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList7:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList8:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    KnownList9:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    MemorizedList0:
      - Spell: word
        SpellFlags: byte
        SpellMetaMagic: byte
    MemorizedList1:
```

(continues on next page)

(continued from previous page)

```
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList2:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList3:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList4:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList5:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList6:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList7:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList8:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
      MemorizedList9:
        - Spell: word
          SpellFlags: byte
          SpellMetaMagic: byte
  Color_Hair: byte
  Color_Skin: byte
  Color_Tattoo1: byte
  Color_Tattoo2: byte
  Con: byte
  Conversation: resref
  CurrentHitPoints: short
  DecayTime: dword
  Deity: cexostr
  Description: cexolocstr
  Dex: byte
  Disarmable: byte
  FactionID: word
  FeatList:
    - Feat: word
  FirstName: cexolocstr
  Gender: byte
```

(continues on next page)

(continued from previous page)

```yaml
  GoodEvil: byte
  HitPoints: short
  Int: byte
  Interruptable: byte
  IsImmortal: byte
  IsPC: byte
  LastName: cexolocstr
  LawfulChaotic: byte
  Lootable: byte
  MaxHitPoints: short
  NaturalAC: byte
  NoPermDeath: byte
  PerceptionRange: byte
  Phenotype: int
  Plot: byte
  PortraitId: word
  Race: byte
  ScriptAttacked: resref
  ScriptDamaged: resref
  ScriptDeath: resref
  ScriptDialogue: resref
  ScriptDisturbed: resref
  ScriptEndRound: resref
  ScriptHeartbeat: resref
  ScriptOnBlocked: resref
  ScriptOnNotice: resref
  ScriptRested: resref
  ScriptSpawn: resref
  ScriptSpellAt: resref
  ScriptUserDefine: resref
  SkillList:
    - Rank: byte
  SoundSetFile: word
  SpecAbilityList:
    - Spell: word
      SpellCasterLevel: byte
      SpellFlags: byte
  StartingPackage: byte
  Str: byte
  Subrace: cexostr
  Tag: cexostr
  Tail_New: dword # 1.69
  TemplateList:
    - TemplateID: word
  TemplateResRef: resref
  VarTable: vartable.yml
  WalkRate: int
  Wings_New: dword # 1.69
  Wis: byte
  fortbonus: short
  refbonus: short
  willbonus: short
```

(continues on next page)

(continued from previous page)

```yaml
blueprint:
  Comment: cexostr
  Equip_ItemList:
    - Dropable: byte
      EquippedRes: resref
  ItemList:
    - Dropable: byte
      InventoryRes: resref
      Pickpocketable: byte
      Repos_PosX: word
      Repos_Posy: word
  PaletteID: byte

instance:
  Equip_ItemList:
    - uti.yml: instance
      Dropable: byte
  ItemList:
    - uti.yml: instance
      Dropable: byte
      Pickpocketable: byte
      Repos_PosX: word
      Repos_Posy: word
```

6.33 utd

```yaml
all:
  AnimationState: byte
  Appearance: dword
  AutoRemoveKey: byte
  CloseLockDC: byte
  Conversation: resref
  CurrentHP: short
  Description: cexolocstr
  DisarmDC: byte
  Faction: dword
  Fort: byte
  GenericType_New: dword # 1.69
  GenericType: byte # until 1.69
  HP: short
  Hardness: byte
  Interruptable: byte
  KeyName: string
  KeyRequired: byte
  LinkedTo: string
  LinkedToFlags: byte
  LoadScreenID: word
  LocName: cexolocstr
  Lockable: byte
```

(continues on next page)

(continued from previous page)

```
  Locked: byte
  OnClick: resref
  OnClosed: resref
  OnDamaged: resref
  OnDeath: resref
  OnDisarm: resref
  OnFailToOpen: resref
  OnHeartbeat: resref
  OnLock: resref
  OnMeleeAttacked: resref
  OnOpen: resref
  OnSpellCastAt: resref
  OnTrapTriggered: resref
  OnUnlock: resref
  OnUserDefined: resref
  OpenLockDC: byte
  Plot: byte
  PortraitId: word
  Ref: byte
  Tag: string
  TemplateResRef: resref
  TrapDetectDC: byte
  TrapDetectable: byte
  TrapDisarmable: byte
  TrapFlag: byte
  TrapOneShot: byte
  TrapType: byte
  VarTable: vartable.yml
  Will: byte

blueprint:
  Comment: string
  PaletteID: byte

instance:
  Bearing: float
  X: float
  "Y": float
  Z: float
```

6.34 ute

```
all:
  Active: byte
  CreatureList:
    - $struct_id: 0
      Appearance: int
      CR: float
      ResRef: resref
      SingleSpawn: byte
```

(continues on next page)

(continued from previous page)

```
  Difficulty: int
  DifficultyIndex: int
  Faction: dword
  LocalizedName: cexolocstr
  MaxCreatures: int
  OnEntered: resref
  OnExhausted: resref
  OnExit: resref
  OnHeartbeat: resref
  OnUserDefined: resref
  PlayerOnly: byte
  RecCreatures: int
  Reset: byte
  ResetTime: int
  Respawns: int
  SpawnOption: int
  Tag: cexostr
  TemplateResRef: resref

blueprint:
  Comment: cexostr
  PaletteID: byte

instance:
  Geometry:
    - $struct_id: 1
      X: float
      "Y": float
      Z: float
  SpawnPointList:
    - $struct_id: 0
      Orientation: float
      X: float
      "Y": float
      Z: float
  XPosition: float
  YPosition: float
  ZPosition: float
```

6.35 uti

```
all:
  AddCost: dword
  ArmorPart_Belt: byte
  ArmorPart_LBicep: byte
  ArmorPart_LFArm: byte
  ArmorPart_LFoot: byte
  ArmorPart_LHand: byte
  ArmorPart_LShin: byte
  ArmorPart_LShoul: byte
```

(continues on next page)

(continued from previous page)

```yaml
  ArmorPart_LThigh: byte
  ArmorPart_Neck: byte
  ArmorPart_Pelvis: byte
  ArmorPart_RBicep: byte
  ArmorPart_RFArm: byte
  ArmorPart_RFoot: byte
  ArmorPart_RHand: byte
  ArmorPart_RShin: byte
  ArmorPart_RShoul: byte
  ArmorPart_RThigh: byte
  ArmorPart_Robe: byte
  ArmorPart_Torso: byte
  BaseItem: int
  Charges: byte
  Cloth1Color: byte
  Cloth2Color: byte
  Cost: dword
  Cursed: byte
  DescIdentified: locstring
  Description: locstring
  Identified: byte
  Leather1Color: byte
  Leather2Color: byte
  LocalizedName: locstring
  Metal1Color: byte
  Metal2Color: byte
  ModelPart1: byte
  ModelPart2: byte
  ModelPart3: byte
  Plot: byte
  PropertiesList:
  - ChanceAppear: byte
    CostTable: byte
    CostValue: word
    Param1: byte
    Param1Value: byte
    PropertyName: word
    Subtype: word
  StackSize: word
  Stolen: byte
  Tag: cexostr
  TemplateResRef: resref
  VarTable: vartable.yml

blueprint:
  Comment: cexostr
  PaletteID: byte

instance:
  XOrientation: float
  XPosition: float
  YOrientation: float
```

(continues on next page)

(continued from previous page)

```
  YPosition: float
  ZPosition: float
```

6.36 utm

```
all:
  BM_MarkDown: int
  BlackMarket: byte
  ID: byte
  IdentifyPrice: int
  LocName: locstring
  MarkDown: int
  MarkUp: int
  MaxBuyPrice: int
  OnOpenStore: resref
  OnStoreClosed: resref
  ResRef: resref
  StoreGold: int
  Tag: string
  VarTable: vartable.yml
  WillNotBuy:
    - BaseItem: int
  WillOnlyBuy:
    - BaseItem: int

blueprint:
  Comment: string
  StoreList:
    - ItemList:
        - Infinite: byte
          InventoryRes: resref
          Repos_PosX: word
          Repos_Posy: word

instance:
  StoreList:
    - Comment: string
      ItemList:
        - uti.yml: instance
          Infinite: byte
          Repos_PosX: word
          Repos_Posy: word
  XOrientation: float
  XPosition: float
  YOrientation: float
  YPosition: float
  ZPosition: float
```

6.37 utp

```
all:
  AnimationState: byte
  Appearance: dword
  AutoRemoveKey: byte
  BodyBag: byte
  CloseLockDC: byte
  Conversation: resref
  CurrentHP: short
  Description: locstring
  DisarmDC: byte
  Faction: dword
  Fort: byte
  HP: short
  Hardness: byte
  HasInventory: byte
  Interruptable: byte
  KeyName: string
  KeyRequired: byte
  LocName: locstring
  Lockable: byte
  Locked: byte
  OnClick: resref
  OnClosed: resref
  OnDamaged: resref
  OnDeath: resref
  OnDisarm: resref
  OnHeartbeat: resref
  OnInvDisturbed: resref
  OnLock: resref
  OnMeleeAttacked: resref
  OnOpen: resref
  OnSpellCastAt: resref
  OnTrapTriggered: resref
  OnUnlock: resref
  OnUsed: resref
  OnUserDefined: resref
  OpenLockDC: byte
  Plot: byte
  PortraitId: word
  Ref: byte
  Static: byte
  Tag: string
  TemplateResRef: resref
  TrapDetectDC: byte
  TrapDetectable: byte
  TrapDisarmable: byte
  TrapFlag: byte
  TrapOneShot: byte
  TrapType: byte
  Type: byte
```

(continues on next page)

(continued from previous page)

```
  Useable: byte
  VarTable: vartable.yml
  Will: byte

blueprint:
  Comment: string
  ItemList:
    - $struct_id: index
      InventoryRes: resref
      Repos_PosX: word
      Repos_Posy: word
  PaletteID: byte

instance:
  Bearing: float
  ItemList:
    - $struct_id: index
      uti.yml: instance
      Repos_PosX: word
      Repos_Posy: word
  X: float
  "Y": float
  Z: float
```

6.38 uts

```
all:
  Active: byte
  Continuous: byte
  Elevation: float
  Hours: dword
  Interval: dword
  IntervalVrtn: dword
  LocName: locstring
  Looping: byte
  MaxDistance: float
  MinDistance: float
  PitchVariation: float
  Positional: byte
  Priority: byte
  Random: byte
  RandomPosition: byte
  RandomRangeX: float
  RandomRangeY: float
  Sounds:
    - $struct_id: 0
      Sound: resref
  Tag: string
  TemplateResRef: resref
  Times: byte
```

(continues on next page)

(continued from previous page)

```
  Volume: byte
  VolumeVrtn: byte

blueprint:
  Comment: string
  PaletteID: byte

instance:
  GeneratedType: dword # Docs say byte, but GFFs say dword
  XPosition: float
  YPosition: float
  ZPosition: float
```

6.39 utt

```
all:
  AutoRemoveKey: byte # Obsolete
  Cursor: byte
  DisarmDC: byte
  Faction: dword
  HighlightHeight: float
  KeyName: string # Obsolete
  LinkedTo: string
  LinkedToFlags: byte
  LoadScreenID: word
  LocalizedName: locstring
  OnClick: resref
  OnDisarm: resref
  OnTrapTriggered: resref
  PortraitId: word
  ScriptHeartbeat: resref
  ScriptOnEnter: resref
  ScriptOnExit: resref
  ScriptUserDefine: resref
  Tag: string
  TemplateResRef: resref
  TrapDetectDC: byte
  TrapDetectable: byte
  TrapDisarmable: byte
  TrapFlag: byte
  TrapOneShot: byte
  TrapType: byte
  Type: int
  VarTable: vartable.yml

blueprint:
  Comment: string
  PaletteID: byte

instance:
```

(continues on next page)

(continued from previous page)

```
  Geometry:
    - $struct_id: 3
      PointX: float
      PointY: float
      PointZ: float
  XOrientation: float
  XPosition: float
  YOrientation: float
  YPosition: float
  ZOrientation: float
  ZPosition: float
```

6.40 utw

```
all:
  Appearance: byte
  Description: locstring
  HasMapNote: byte
  LinkedTo: string
  LocalizedName: locstring
  MapNote: locstring
  MapNoteEnabled: byte
  Tag: string
  TemplateResRef: resref
  VarTable: vartable.yml

blueprint:
  Comment: string
  PaletteID: byte

instance:
  XOrientation: float
  XPosition: float
  YOrientation: float
  YPosition: float
  ZPosition: float
```

6.41 vartable

```
VarTable:
  - Name: string
    Type: dword
    Value: string|int|float|location|object # Not sure if any of the others are ever␣
↪saved.
```

PYTHON MODULE INDEX

r

INDEX

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z