DEFIANCE User Documentation

Release 1.0

BPHK2023

Jul 01, 2024

CONTENTS

This is a suggested outline for adding new module documentation to *ns-3*. See `src/click/doc/click.rst` for an example.

The introductory paragraph is for describing what this code is trying to model.

For consistency (italicized formatting), please use *ns-3* to refer to ns-3 in the documentation (and likewise, **|ns2|** for ns-2). These macros are defined in the file `replace.txt`.

CHAPTER

ONE

MODEL DESCRIPTION

The source code for the new module lives in the directory `contrib/DEFIANCE`.

Add here a basic description of what is being modeled.

1.1 Design

Briefly describe the software design of the model and how it fits into the existing ns-3 architecture.

1.2 Scope and Limitations

What can the model do? What can it not do? Please use this section to describe the scope and limitations of the model.

1.3 References

Add academic citations here, such as if you published a paper on this model, or if readers should read a particular specification or other work.

CHAPTER

TWO

USAGE

This section is principally concerned with the usage of your model, using the public API. Focus first on most common usage patterns, then go into more advanced topics.

2.1 Building New Module

Include this subsection only if there are special build instructions or platform limitations.

2.2 Helpers

What helper API will users typically use? Describe it here.

2.3 Attributes

What classes hold attributes, and what are the key ones worth mentioning?

2.4 Output

What kind of data does the model generate? What are the key trace sources? What kind of logging output can be enabled?

2.5 Advanced Usage

Go into further details (such as using the API outside of the helpers) in additional sections, as needed.

2.6 Examples

What examples using this new code are available? Describe them here.

2.7 Troubleshooting

Add any tips for avoiding pitfalls, etc.

CHAPTER

THREE

VALIDATION

Describe how the model has been tested/validated. What tests run in the test suite? How much API and code is covered by the tests? Again, references to outside published work may help here.