**Document Title**

Subject

Author

Morphkit logo

# Home

Welcome to the documentation for Morphkit, a Python toolkit for processing the output of the Morpheus Morphological analyzer.

This package was created as part of a research project to create a [Text-Fabric dataset](https://tonyjurg.github.io/N1904addons/) containing the Morpheus analytical data for each word of the [Nestle1904](https://centerblc.github.io/N1904/) Greek New Testament. A number of functions are specificly related to this use case.

# Features

* Lightweight and modular morphological toolkit.
* Compatible with Morpheus environments.
* Designed for use with Greek New Testament texts (SP tags).
* Basic support for Latin.

# Using this package

[Installation](#_e3a4aa6ae53c85312025264fe419cff9)

How to install this package in your Python environments

[Usage](#_90b9c83e37e6110bda92bda3da542389)

How to use this package

# GitHub

You can find the project’s source code on [GitHub](https://github.com/tonyjurg/morphkit) and report issues or suggestions at the [issue tracker](https://github.com/tonyjurg/morphkit/issues).

## Installation

Since this not a ‘normal package’ (yet), you need to import it by adding the relative path to the location where the **directory** /morphkit can be found to the path-search (using sys.path.insert). After this, your notebook will be able to load morphkit.

**import** **sys**  
sys.path.insert(0, "../../morphkit") *# relative to notebook dir*  
**import** **morphkit**

### Dependencies

You also need to have install the [dependencies](https://github.com/tonyjurg/morphkit/blob/main/morphkit/requirements.txt):

* [beta-code-py](https://pypi.org/project/beta-code-py/) (for Beta Code ↔ Unicode conversion)
* [requests](https://pypi.org/project/requests/) (for API calls using HTTP to Morpheus)

### Morpheus

Since the purpose of this package it to interface with an Morpheus API endpoint, you need access to such a service. A Morpheus API endpoint can be set up locally, e.g., via the [perseidsproject/morpheus-api Docker image](https://hub.docker.com/r/perseidsproject/morpheus-api).

## Usage

Some example use cases.

### Obtain Morpheus Analytic blocks for a Greek word

*# convert unicode greek to betacode*  
**import** **beta\_code**  
bc\_word=beta\_code.greek\_to\_beta\_code(u'του')  
api\_endpoint="10.0.1.156:1315"  
print(morphkit.get\_word\_blocks(bc\_word,api\_endpoint))

This will output three Morpheus Analytic blocks (one shown):

:raw tou  
  
:workw tou=  
:lem o(  
:prvb  
:aug1  
:stem tou= indeclform  
:suff  
:end masc/neut gen sg indeclform article

### Get the compact analysis results

print(morphkit.get\_word\_blocks(bc\_word,api\_endpoint,output="compact"))

This prints (for the same word as the previous example) the compact notation:

tou  
<NL>N tou=,o( masc/neut gen sg indeclform article</NL><NL>N tou=,ti/s gen sg attic indeclform indecl</NL><NL>N tis gen sg attic enclitic indeclform indef</NL>

### Perform a full analysis

The following will analyze a word and produce a full dictionary of Morpheus data augmented with Part of Speech and morph tag information.

result=morphkit.analyze\_word\_with\_morpheus("mo/non",api\_endpoint)

This provides a Python dictionary like below (truncated):

{'raw\_bc': 'mo/non',  
 'raw\_uc': 'μόνον',  
 'blocks': 2,  
 'analyses': [{'raw\_bc': 'mo/non',  
 'raw\_uc': 'μόνον',  
 'workw\_bc': 'mo/non',  
 'workw\_uc': 'μόνον',  
 'lem\_full\_bc': 'mo/nos',  
 'lem\_full\_uc': 'μόνος',  
 'lem\_base\_bc': 'mo/nos',  
 'lem\_base\_uc': 'μόνος',  
 'stem\_bc': 'mon',  
 'stem\_uc': 'μον',  
 'stem\_codes': ['os\_h\_on'],  
 'end\_bc': 'on',  
 'end\_uc': 'ον',  
 'gender': 'masc',  
 'case': 'acc',  
 'number': 'sg',  
 'end\_codes': ['os\_h\_on'],  
 'pos': 'noun',  
 'morph': 'N-ASM'},  
 {'raw\_bc': 'mo/non',  
 'raw\_uc': 'μόνον',  
 'workw\_bc': 'mo/non',  
 ...

### Limited Latin support

Latin words can be analyzed and the results stored in a Python dictionary:

**import** **pprint** **as** **pp**  
raw\_text=morphkit.get\_word\_blocks("dico",api\_endpoint,language="latin")  
blocks=morphkit.split\_into\_raw\_blocks(raw\_text)  
all\_parses = []  
**for** block **in** blocks:  
 raw\_beta, parses = morphkit.parse\_word\_block(block,"latin")  
 all\_parses.append(parses)  
 pp.pprint(parses)

This procudes a dictiorary like:

[{'end': 'o\_',  
 'end\_codes': ['conj1'],  
 'lem\_base': 'dico#',  
 'lem\_full': 'dico#1',  
 'lem\_homonym': 1,  
 'mood': 'indicative',  
 'number': 'sg',  
 'person': '1',  
 'raw': 'dico',  
 'stem': 'dic',  
 'stem\_codes': ['conj1', 'are\_vb'],  
 'tense': 'present',  
 'voice': 'active',  
 'workw': 'dico\_'}]  
[{'end': 'o\_',  
 'end\_codes': ['conj3'],  
 'lem\_base': 'dico#',

To see these examples in action, you can download [this Jupyter Notebook](https://github.com/tonyjurg/morphkit/blob/main/notebooks/Morphkit_usage_examples.ipynb).

# Summary of functions