

---

# **gnirs-pype Documentation**

*Release 1.0.0.dev1*

**Viraja Khatu**

**Oct 28, 2019**



---

# Contents

---

<b>1</b>	<b>Package Documentation</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Running from the Command Line . . . . .	3
1.3	Running from Python . . . . .	3
1.4	Preparing the .cfg Input File . . . . .	16
1.5	Data Reduction Examples . . . . .	17
1.6	Tutorials . . . . .	17
1.7	Known Issues . . . . .	18
1.8	Maintaining GNIRS-Pype . . . . .	18
1.9	Future Work . . . . .	19
1.10	Changelog . . . . .	20
1.11	API . . . . .	20
1.12	Example of GNIRS-Pype File I/O . . . . .	20
<b>2</b>	<b>Indices and tables</b>	<b>21</b>



GNIRS-Pype is an under development Gemini data reduction framework for Gemini instruments and here we provide the documentation for its associated GNIRS pipeline.



### 1.1 Introduction

GNIRS-Pype, for now, uses Python 2.7. Please keep this in mind.

### 1.2 Running from the Command Line

GNIRS-Pype, for now, is started simply by running the gnirsPipeline script from the command line.

### 1.3 Running from Python

Python Programmers: even though GNIRS-Pype's API has not been defined yet, you can still run GNIRS-Pype Pipeline, Steps, Routines, and Tasks from a Python interpreter by importing them.

For example, to run the combine 2D spectra from the Python interpreter:

```
This is currently under development.
```

You might be able to figure out what imports you need by checking the tops of the relevant scripts.

#### 1.3.1 Examples of Running from the Command Line

Starting a Data Reduction from the Beginning

This is currently under development.

Starting a Data Reduction from a Specified Point

You can run each step, one at a time, from the command line like so. *You need a gnirs.cfg file in your current working directory to run individual steps. Each step requires the general config section and its unique and/or additional config sections to be populated.*

You can also run an individual step by turning them on or off in gnirsPipeline config and running the gnirsPipeline.

**gnirsSort:** To only sort and copy the GNIRS raw data or create symbolic links in between the qorking directories use a gnirs.cfg file like this:

```
# GNIRS-Pype configuration file.

[defaults]
gnirs-pipeVersion = 1.0.0
manualMode = False
overwrite = False

[ScienceDirectories]

[TelluricDirectories]

[CalibrationDirectories]

[gnirsPipeline]
getData = False
sort = True
checkData = False
calibrationReduction = False
scienceReduction = False
telluricReduction = False
combineSpectra2D = False
extractSpectralD = False
telluricCorrection = False
fluxCalibration = False
combineOrdersXD = False
calculateSpectrumSNR = False
writeDataSheet = False

[interactive]
nsprepareInter = False
nsflatInter = False
nscombineInter = False
nssdistInter = False
nswavelengthInter = False
nsfitcoordsInter = False
nsextractInter = False
hLineInter = False
continuumInter = False
telluricInter = False
tempInter = False

[getData]
rawPath = rawData/
program =

[sort]
proprietaryCookie =
telluricTimeThreshold = 5400.0 ; units [seconds]
```

(continues on next page)



(continued from previous page)

```
[checkData]
# No input parameters required

[calibrationReduction]
Start = 1
Stop = 5
cleanir_IRflats = False
cleanir_QHflats = False
cleanir_arcs = False
cleanir_pinholes = False

[scienceReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 20
checkPeaksMatch = True

[telluricReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 50

[combineSpectra2D]
# No input parameters required

[extractSpectral1D]
useApall = True
extractionApertureRadius = 4
checkPeaksMatch = True
toleranceOffset = 5
extractionFullSlit = False
extractionStepwise = False
extractionStepSize = 6 ; units [pixels]

[telluricCorrection]
Start = 1
Stop = 5
hLineMethod = vega

[fluxCalibration]
Start = 1
Stop = 6
telluricRA =
telluricDEC =
telluricSpectralType =
telluricMagnitude =
telluricTemperature =

[combineOrdersXD]
# no input parameters required

[calculateSpectrumSNR]
# This section is currently under development.
```

(continues on next page)

(continued from previous page)

```
Start = 1
Stop = 1

[writeDataSheet]
# no input parameters required

[TelluricRegions] ; regions to use when tweaking the Telluric line removal
3 = *
4 = 14200:18000
5 = 11200:13400,14200:15200
6 = 11000:12000
7 = 9000:10000
8 = 8500:9500

[HLineRegions] ; regions to use when tweaking the H line removal
3 = 21400:21900
4 = 15500:18000
5 = 12750:12900
6 = 10700:11200
7 = 9000:10500
8 = 8500:9300

[ContinuumRegions] ; regions to use when fitting the Telluric continuum
3 = *
4 = *
5 = 11200:13300,14300:15200
6 = 9500:10900,11700:12600
7 = 8500:9200,9800:10700
8 = 7500:9500
```

And run the gnirsPipeline with:

```
This section is currently under development.
```

**gnirsBaselineCalibration:** To only reduce calibrations use a gnirs.cfg file like this:

```
# GNIRS-Pype configuration file.

[defaults]
gnirs-pipeVersion = 1.0.0
manualMode = False
overwrite = False

[ScienceDirectories]

[TelluricDirectories]

[CalibrationDirectories]

[gnirsPipeline]
getData = False
sort = False
checkData = False
calibrationReduction = True
scienceReduction = False
telluricReduction = False
combineSpectra2D = False
```

(continues on next page)

(continued from previous page)

```
extractSpectralD = False
telluricCorrection = False
fluxCalibration = False
combineOrdersXD = False
calculateSpectrumSNR = False
writeDataSheet = False

[interactive]
nsprepareInter = False
nsflatInter = False
nscombineInter = False
nssdistInter = False
nswavelengthInter = False
nsfitcoordsInter = False
nsextractInter = False
hLineInter = False
continuumInter = False
telluricInter = False
tempInter = False

[getData]
rawPath = rawData/
program =

[sort]
proprietaryCookie =
telluricTimeThreshold = 5400.0 ; units [seconds]

[checkData]
# No input parameters required

[calibrationReduction]
Start = 1
Stop = 5
cleanir_IRflats = False
cleanir_QHflats = False
cleanir_arcs = False
cleanir_pinholes = False

[scienceReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 20
checkPeaksMatch = True

[telluricReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 50

[combineSpectra2D]
# No input parameters required
```

(continues on next page)

```
[extractSpectralD]
useApall = True
extractionApertureRadius = 4
checkPeaksMatch = True
toleranceOffset = 5
extractionFullSlit = False
extractionStepwise = False
extractionStepSize = 6 ; units [pixels]

[telluricCorrection]
Start = 1
Stop = 5
hLineMethod = vega

[fluxCalibration]
Start = 1
Stop = 6
telluricRA =
telluricDEC =
telluricSpectralType =
telluricMagnitude =
telluricTemperature =

[combineOrdersXD]
# no input parameters required

[calculateSpectrumSNR]
# This section is currently under development.
Start = 1
Stop = 1

[writeDataSheet]
# no input parameters required

[TelluricRegions] ; regions to use when tweaking the Telluric line removal
3 = *
4 = 14200:18000
5 = 11200:13400,14200:15200
6 = 11000:12000
7 = 9000:10000
8 = 8500:9500

[HLineRegions] ; regions to use when tweaking the H line removal
3 = 21400:21900
4 = 15500:18000
5 = 12750:12900
6 = 10700:11200
7 = 9000:10500
8 = 8500:9300

[ContinuumRegions] ; regions to use when fitting the Telluric continuum
3 = *
4 = *
5 = 11200:13300,14300:15200
6 = 9500:10900,11700:12600
7 = 8500:9200,9800:10700
8 = 7500:9500
```

And run the gnirsPipeline with:

This section is currently under development.

**gnirsReduce Science:** To only reduce science data use a config.cfg file like this: *Make sure to populate ScienceDirectories, TelluricDirectories and CalibrationDirectories before running!*

```
# GNIRS-Pype configuration file.

[defaults]
gnirs-pipeVersion = 1.0.0
manualMode = False
overwrite = False

[ScienceDirectories]

[TelluricDirectories]

[CalibrationDirectories]

[gnirsPipeline]
getData = False
sort = False
checkData = False
calibrationReduction = False
scienceReduction = True
telluricReduction = False
combineSpectra2D = False
extractSpectra1D = False
telluricCorrection = False
fluxCalibration = False
combineOrdersXD = False
calculateSpectrumSNR = False
writeDataSheet = False

[interactive]
nsprepareInter = False
nsflatInter = False
nscombineInter = False
nssdistInter = False
nswavelengthInter = False
nsfitcoordsInter = False
nsextractInter = False
hLineInter = False
continuumInter = False
telluricInter = False
tempInter = False

[getData]
rawPath = rawData/
program =

[sort]
proprietaryCookie =
telluricTimeThreshold = 5400.0 ; units [seconds]

[checkData]
# No input parameters required
```

(continues on next page)

(continued from previous page)

```
[calibrationReduction]
Start = 1
Stop = 5
cleanir_IRflats = False
cleanir_QHflats = False
cleanir_arcs = False
cleanir_pinholes = False

[scienceReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 20
checkPeaksMatch = True

[telluricReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 50

[combineSpectra2D]
# No input parameters required

[extractSpectral1D]
useApall = True
extractionApertureRadius = 4
checkPeaksMatch = True
toleranceOffset = 5
extractionFullSlit = False
extractionStepwise = False
extractionStepSize = 6 ; units [pixels]

[telluricCorrection]
Start = 1
Stop = 5
hLineMethod = vega

[fluxCalibration]
Start = 1
Stop = 6
telluricRA =
telluricDEC =
telluricSpectralType =
telluricMagnitude =
telluricTemperature =

[combineOrdersXD]
# no input parameters required

[calculateSpectrumSNR]
# This section is currently under development.
Start = 1
Stop = 1
```

(continues on next page)

(continued from previous page)

```
[writeDataSheet]
# no input parameters required

[TelluricRegions] ; regions to use when tweaking the Telluric line removal
3 = *
4 = 14200:18000
5 = 11200:13400,14200:15200
6 = 11000:12000
7 = 9000:10000
8 = 8500:9500

[HLineRegions] ; regions to use when tweaking the H line removal
3 = 21400:21900
4 = 15500:18000
5 = 12750:12900
6 = 10700:11200
7 = 9000:10500
8 = 8500:9300

[ContinuumRegions] ; regions to use when fitting the Telluric continuum
3 = *
4 = *
5 = 11200:13300,14300:15200
6 = 9500:10900,11700:12600
7 = 8500:9200,9800:10700
8 = 7500:9500
```

And run the gnirsPipeline with:

```
This is currently under development.
```

**gnirsFluxCalibrate:** To only do a flux calibration use a gnirs.cfg file like this: *Make sure to populate scienceDirectoryList, telluricDirectoryList and calibrationDirectoryList before running!*

```
# GNIRS-Pype configuration file.

[defaults]
gnirs-pipeVersion = 1.0.0
manualMode = False
overwrite = False

[ScienceDirectories]

[TelluricDirectories]

[CalibrationDirectories]

[gnirsPipeline]
getData = False
sort = False
checkData = False
calibrationReduction = False
scienceReduction = False
telluricReduction = False
combineSpectra2D = False
```

(continues on next page)

(continued from previous page)

```
extractSpectral1D = False
telluricCorrection = False
fluxCalibration = True
combineOrdersXD = False
calculateSpectrumSNR = False
writeDataSheet = False

[interactive]
nsprepareInter = False
nsflatInter = False
nscombineInter = False
nssdistInter = False
nswavelengthInter = False
nsfitcoordsInter = False
nsextractInter = False
hLineInter = False
continuumInter = False
telluricInter = False
tempInter = False

[getData]
rawPath = rawData/
program =

[sort]
proprietaryCookie =
telluricTimeThreshold = 5400.0 ; units [seconds]

[checkData]
# No input parameters required

[calibrationReduction]
Start = 1
Stop = 5
cleanir_IRflats = False
cleanir_QHflats = False
cleanir_arcs = False
cleanir_pinholes = False

[scienceReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 20
checkPeaksMatch = True

[telluricReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 50

[combineSpectra2D]
# No input parameters required
```

(continues on next page)



(continued from previous page)

```
[extractSpectralD]
useApall = True
extractionApertureRadius = 4
checkPeaksMatch = True
toleranceOffset = 5
extractionFullSlit = False
extractionStepwise = False
extractionStepSize = 6 ; units [pixels]

[telluricCorrection]
Start = 1
Stop = 5
hLineMethod = vega

[fluxCalibration]
Start = 1
Stop = 6
telluricRA =
telluricDEC =
telluricSpectralType =
telluricMagnitude =
telluricTemperature =

[combineOrdersXD]
# no input parameters required

[calculateSpectrumSNR]
# This section is currently under development.
Start = 1
Stop = 1

[writeDataSheet]
# no input parameters required

[TelluricRegions] ; regions to use when tweaking the Telluric line removal
3 = *
4 = 14200:18000
5 = 11200:13400,14200:15200
6 = 11000:12000
7 = 9000:10000
8 = 8500:9500

[HLineRegions] ; regions to use when tweaking the H line removal
3 = 21400:21900
4 = 15500:18000
5 = 12750:12900
6 = 10700:11200
7 = 9000:10500
8 = 8500:9300

[ContinuumRegions] ; regions to use when fitting the Telluric continuum
3 = *
4 = *
5 = 11200:13300,14300:15200
6 = 9500:10900,11700:12600
7 = 8500:9200,9800:10700
8 = 7500:9500
```

And run the gnirsPipeline with:

This is currently under development.

**gnirsCombineOrdersXD Orders Combining:** To only combine different spectral orders use a gnirs.cfg file like this:  
*Make sure to populate scienceDirectoryList, telluricDirectoryList and calibrationDirectoryList before running!*

```
# GNIRS-Pype configuration file.

[defaults]
gnirs-pipeVersion = 1.0.0
manualMode = False
overwrite = False

[ScienceDirectories]

[TelluricDirectories]

[CalibrationDirectories]

[gnirsPipeline]
getData = False
sort = False
checkData = False
calibrationReduction = False
scienceReduction = False
telluricReduction = False
combineSpectra2D = False
extractSpectral1D = False
telluricCorrection = False
fluxCalibration = False
combineOrdersXD = True
calculateSpectrumSNR = False
writeDataSheet = False

[interactive]
nsprepareInter = False
nsflatInter = False
nscombineInter = False
nssdistInter = False
nswavelengthInter = False
nsfitcoordsInter = False
nsextractInter = False
hLineInter = False
continuumInter = False
telluricInter = False
tempInter = False

[getData]
rawPath = rawData/
program =

[sort]
proprietaryCookie =
telluricTimeThreshold = 5400.0 ; units [seconds]

[checkData]
# No input parameters required
```

(continues on next page)

(continued from previous page)

```
[calibrationReduction]
Start = 1
Stop = 5
cleanir_IRflats = False
cleanir_QHflats = False
cleanir_arcs = False
cleanir_pinholes = False

[scienceReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 20
checkPeaksMatch = True

[telluricReduction]
Start = 1
Stop = 5
cleanir = False
radiationCorrectionMethod = fixpix
radiationThreshold = 50

[combineSpectra2D]
# No input parameters required

[extractSpectral1D]
useApall = True
extractionApertureRadius = 4
checkPeaksMatch = True
toleranceOffset = 5
extractionFullSlit = False
extractionStepwise = False
extractionStepSize = 6 ; units [pixels]

[telluricCorrection]
Start = 1
Stop = 5
hLineMethod = vega

[fluxCalibration]
Start = 1
Stop = 6
telluricRA =
telluricDEC =
telluricSpectralType =
telluricMagnitude =
telluricTemperature =

[combineOrdersXD]
# no input parameters required

[calculateSpectrumSNR]
# This section is currently under development.
Start = 1
Stop = 1
```

(continues on next page)

(continued from previous page)

```
[writeDataSheet]
# no input parameters required

[TelluricRegions] ; regions to use when tweaking the Telluric line removal
3 = *
4 = 14200:18000
5 = 11200:13400,14200:15200
6 = 11000:12000
7 = 9000:10000
8 = 8500:9500

[HLineRegions] ; regions to use when tweaking the H line removal
3 = 21400:21900
4 = 15500:18000
5 = 12750:12900
6 = 10700:11200
7 = 9000:10500
8 = 8500:9300

[ContinuumRegions] ; regions to use when fitting the Telluric continuum
3 = *
4 = *
5 = 11200:13300,14300:15200
6 = 9500:10900,11700:12600
7 = 8500:9200,9800:10700
8 = 7500:9500
```

And run the gnirsPipeline with:

```
This is currently under development.
```

## 1.4 Preparing the .cfg Input File

GNIRS-Pype reads data reduction parameters with the Python 2.7 built-in ConfigParser. See <https://docs.python.org/2.7/library/configparser.html> for full documentation on the parser.

### 1.4.1 Interactive Input Preparation

This is currently under development.

### 1.4.2 An Example Input File

GNIRS-Pype includes a default configuration file. As of v1.0.1, it looks like this:

```
TODO(Viraja): Updated example
```

## 1.5 Data Reduction Examples

### 1.5.1 Observations of NGC 4736; GN-2011A-Q-126

This is currently under development.

### 1.5.2 Example 2

This is currently under development.

### 1.5.3 Example 3

This is currently under development.

## 1.6 Tutorials

### 1.6.1 H Line Removal

This is currently under development.

### 1.6.2 Custom Telluric Corrections

This is currently under development.

### 1.6.3 Combining XD Orders

This is currently under development.

## 1.7 Known Issues

1.7.1 gnirsPipeline.py

1.7.2 gnirsGetData.py

1.7.3 sort\_data.py

1.7.4 make\_lists.py

1.7.5 link\_cals.py

1.7.6 gnirsCheckData.py

1.7.7 gnirsBaselineCalibration.py

1.7.8 gnirsReduce.py

1.7.9 gnirsCombineSpectra2D.py

1.7.10 gnirsExtractSpectra1D.py

1.7.11 gnirsTelluric.py

1.7.12 gnirsFluxCalibrate.py

1.7.13 gnirsCombineOrdersXD.py

1.7.14 gnirsCalculateSpectramSNR.py

1.7.15 gnirsWriteDataSheet.py

1.7.16 General Issues

- A longstanding bug (see [astropy](#) ) in astropy has made it difficult to build GNIRS-Pype as a binary executable.

## 1.8 Maintaining GNIRS-Pype

### 1.8.1 Documentation

Right now there exists four forms of documentation.

Paper

README.rst

.rst Files in the docs/ directory

This file, others like it in the docs/ directory and the README are written in reStructuredText. This markup language integrates well with Python's automatic documentation builder (we used Sphinx) and Github as well as being human readable. You can read more about reStructuredText [here](#).

Comments and DocStrings in Source Code

## 1.8.2 Tests

This is a TODO. Currently we do not have automated tests.

## 1.8.3 Pipeline Structure

This is currently under development.

GNIRS-Pype is built at the lowest level from Python and IRAF subroutines. It is built so that it is relatively easy to change the implementation of the underlying tasks.

## 1.8.4 Updates

To update GNIRS-Pype, do ...

This is currently under development.

Version Numbers

GNIRS-Pype uses semantic versioning(see <http://semver.org/>). This means version numbers come in

MAJOR.MINOR.PATCH
-------------------

In brief, when releasing a version of Nifty that is not backward-compatible with old test recipes, or changes break the public API, it is time to increment the MAJOR version number.

## 1.8.5 Code Conventions

For naming variables and functions, a mix of camelCase and lower\_case\_with\_underscores was used.

Code style was influenced by the [Google Python Style Guide](#).

GNIRS-Pype uses the Google docstring style. Examples of docstrings can be found [here](#).

## 1.9 Future Work

Throughout the code, vkhatu has placed many TODO notes. These are things that should be reviewed at some point.

Future work:

- Support other GNIRS gratings (10l/mm and 111l/mm)
- Implement more telluric and flux calibration methods.
- Implement instrument signature removal routine.
- Implement differential atmospheric refraction correction routine.
- Implement full automatic Gemini (on server) reduction routine.

- Python 3 compatability (if possible)
- Incorporate more command line options
- Compiling as a self-contained executable
- Full AstroConda integration

## 1.10 Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#) and this project adheres to [Semantic Versioning](#).

### 1.10.1 Unreleased

All in-development changes will be tracked here.

- Adding unit tests for each step and integration test for pipeline.

### 1.10.2 1.0.0 - 2019-08-19

This is currently under development.

## 1.11 API

Note: I didn't have time to implement this using Sphinx automodule. GNIRS-Pype has fairly good docstrings and you can use individual steps, routines and tasks by importing them. This is a TODO.

## 1.12 Example of GNIRS-Pype File I/O

Note: This is for v1.0.0. It is moderately correct up telluric correction.

This is an example of how the GNIRS-Pype directory tree appears after each step of the data reduction. These directory trees were created using a custom **gnirstree** bash command:

```
find . -name .git -prune -o -print | sed -e 's;[^/]*/;|____;g;s;____|; |;g'
```

Add the following line to your ~/.bash\_profile to create the **gnirstree** alias:

```
alias gnirstree="find . -name .git -prune -o -print | sed -e 's;[^/]*/;|____;g;s;____  
↪|; |;g'"
```

### 1.12.1 Example Data Reduction Products:

```
This is currently under development.
```



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`