NREL
Transforming ENERGY

PySAM Workshop

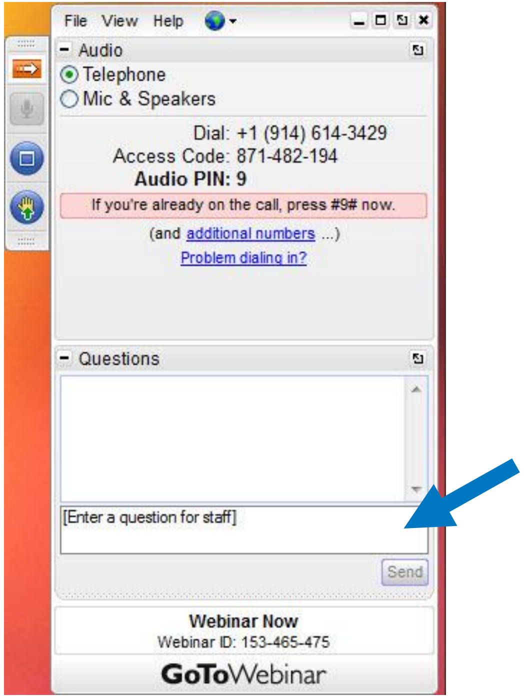
Darice Guittet
2020 SAM Webinars
Oct 14, 2020

SAM Webinars for 2020

Introduction to SAM Workshop	July 22
PV Systems in SAM 2020.2.29	Aug 5
Batteries in SAM 2020.2.29:	
Focus on Battery Technology	Aug 19
Behind-the-Meter Systems	Sep 2
Front-of-Meter Systems	Sep 16
PySAM Workshop	Oct 14

This webinar will be recorded and posted on the SAM website at https://sam.nrel.gov/

Questions and Answers



● Telephone
○ Mic & Speakers

Dial: +1 (914) 614-3429
Access Code: 871-482-194
Audio PIN: 9

If you're already on the call, press #9# now.

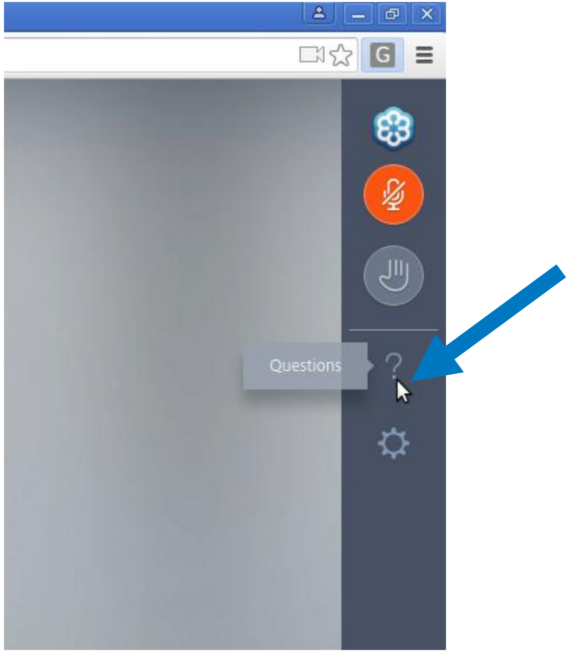
(and additional numbers ...)

Problem dialing in?

[Enter a question for staff]

Webinar Now
Webinar ID: 153-465-475

Desktop application



Instant Join Viewer

1 Introduction

2 How to Get Started

3 Model Initialization

4 Getting Module Information

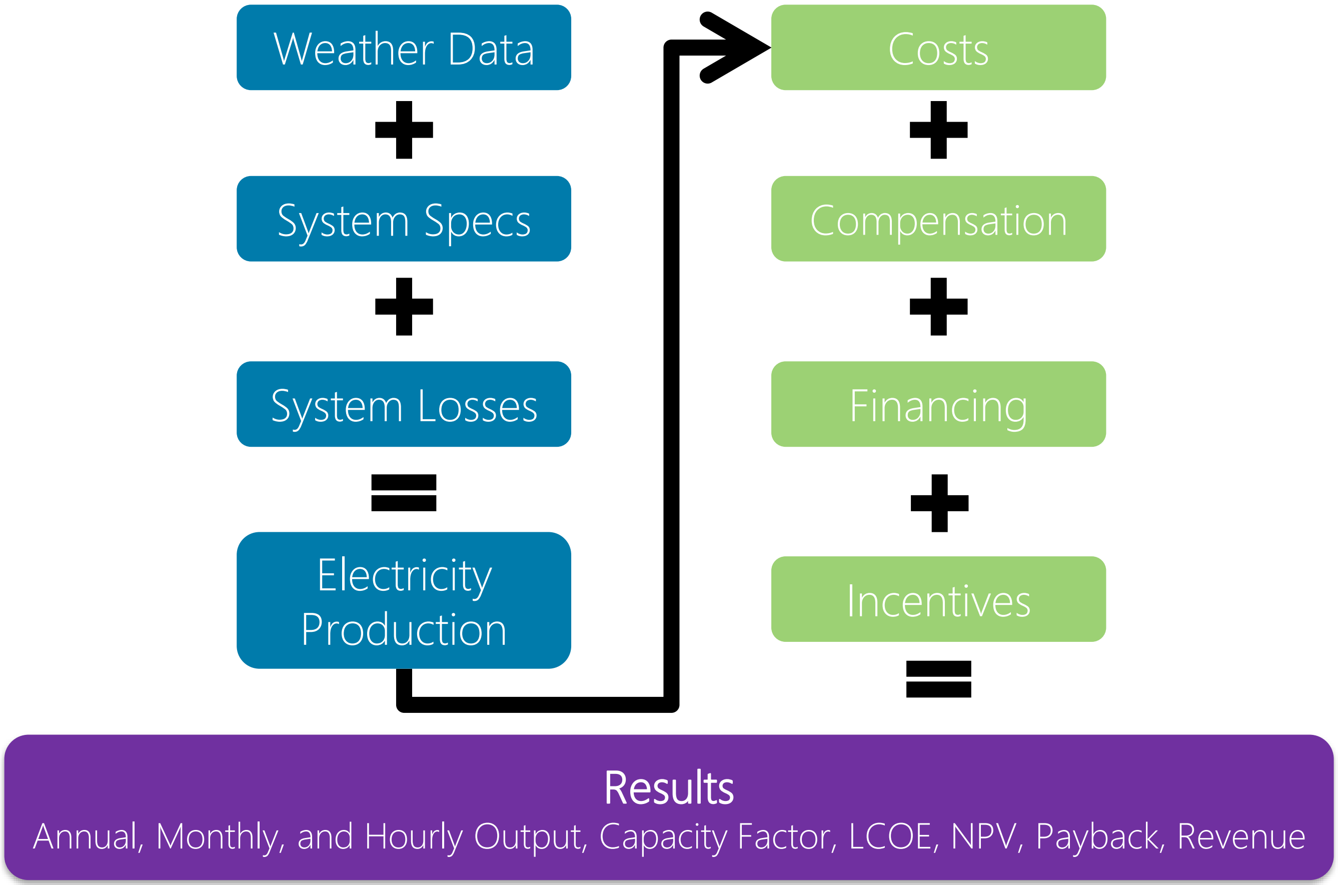
5 Detailed PV-Battery Commercial Example

6 Helper Functions

7 Questions / More Examples

1. **Introduction**
2. How to Get Started
3. Model Initialization
4. Getting Module Information
5. Detailed PV-Battery Commercial Example
6. Helper Functions
7. Questions / More Examples

What is SAM?



Weather Data + System Specs + System Losses = Electricity Production

Costs + Compensation + Financing + Incentives =

Results

Annual, Monthly, and Hourly Output, Capacity Factor, LCOE, NPV, Payback, Revenue

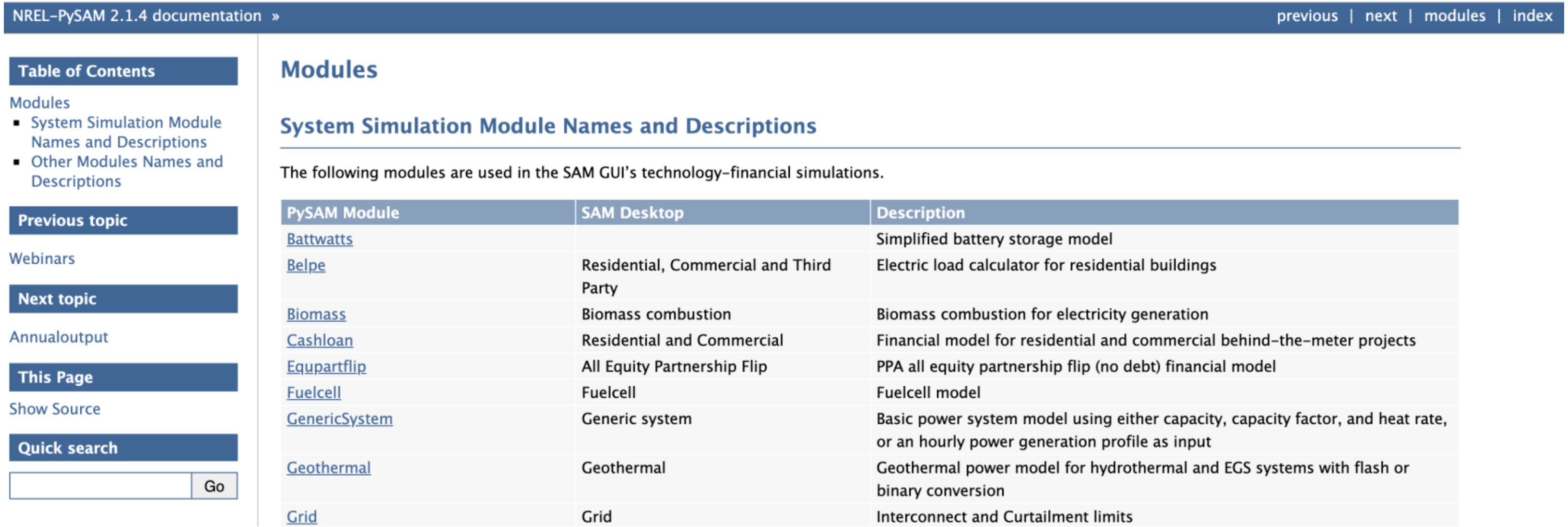
- Photovoltaics
  - Detailed & PVWatts
  - High Concentration PV
- Battery Storage
  - Detailed & PVWatts
  - Generic System
- Concentrating solar power
- Wind
- Fuel Cell
- Geothermal
- Solar water heating
- Biomass
- Marine Energy

- Distributed
  - Residential
  - Commercial
  - Third-party ownership
- Power Purchase Agreements
  - Single owner
  - Equity flips
  - Sale-leaseback
- Merchant Plant
- Host/Developer
- Simple LCOE calculator

What is PySAM?

Python package that enables you to run the underlying modules that make up a simulation in SAM

- Unit modules called compute_modules in the SSC code


NREL-PySAM 2.1.4 documentation »
previous | next | modules | index
Table of Contents
Modules
System Simulation Module Names and Descriptions
Other Modules Names and Descriptions
Previous topic
Webinars
Next topic
Annualoutput
This Page
Show Source
Quick search
Go

Modules

System Simulation Module Names and Descriptions

The following modules are used in the SAM GUI's technology-financial simulations.

PySAM Module	SAM Desktop	Description
Battwatts		Simplified battery storage model
Belpe	Residential, Commercial and Third Party	Electric load calculator for residential buildings
Biomass	Biomass combustion	Biomass combustion for electricity generation
Cashloan	Residential and Commercial	Financial model for residential and commercial behind-the-meter projects
Equpartflip	All Equity Partnership Flip	PPA all equity partnership flip (no debt) financial model
Fuelcell	Fuelcell	Fuelcell model
GenericSystem	Generic system	Basic power system model using either capacity, capacity factor, and heat rate, or an hourly power generation profile as input
Geothermal	Geothermal	Geothermal power model for hydrothermal and EGS systems with flash or binary conversion
Grid	Grid	Interconnect and Curtailment limits

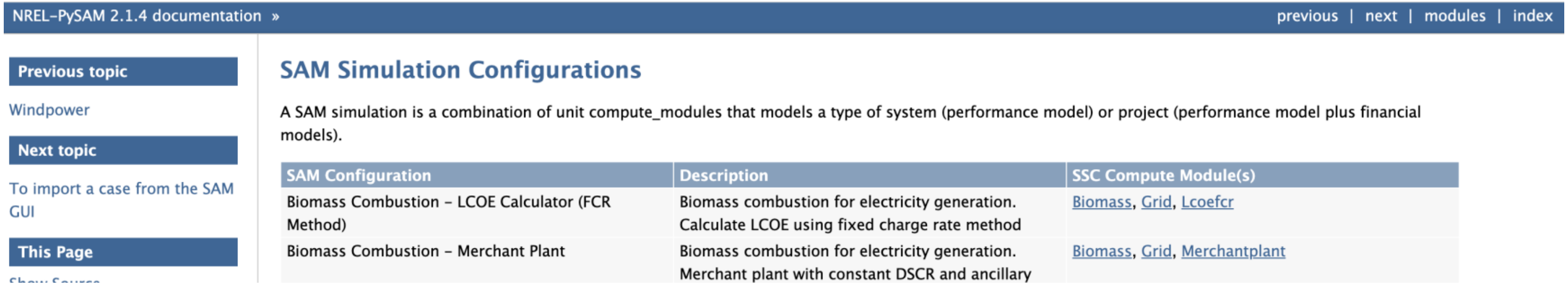
What is PySAM?

Python package that enables you to run the underlying modules that make up a simulation in SAM

- Unit modules called compute_modules in the SSC code

A single simulation is a process chaining together multiple unit modules

- Order
- Information needs to be passed from one to the next


NREL-PySAM 2.1.4 documentation »
previous | next | modules | index
Previous topic
Windpower
Next topic
To import a case from the SAM GUI
This Page
Show Source

SAM Simulation Configurations

A SAM simulation is a combination of unit compute_modules that models a type of system (performance model) or project (performance model plus financial models).

SAM Configuration	Description	SSC Compute Module(s)
Biomass Combustion – LCOE Calculator (FCR Method)	Biomass combustion for electricity generation. Calculate LCOE using fixed charge rate method	Biomass, Grid, Lcoefcr
Biomass Combustion – Merchant Plant	Biomass combustion for electricity generation. Merchant plant with constant DSCR and ancillary	Biomass, Grid, Merchantplant

What is PySAM?

Python package that enables you to run the underlying modules that make up a simulation in SAM

- Unit modules called compute_modules in the SSC code

A single simulation is a process chaining together multiple unit modules

- Order
- Information needs to be passed from one to the next
- Assembled behind the scenes in SAM user interface

PySAM, and SAM's other software development kits, expose these unit modules so that they can be customized and embedded in software applications

What is PySAM?

Python package that enables you to run the underlying modules that make up a simulation in SAM

- Unit modules called compute_modules in the SSC code

A single simulation is a process chaining together multiple unit modules

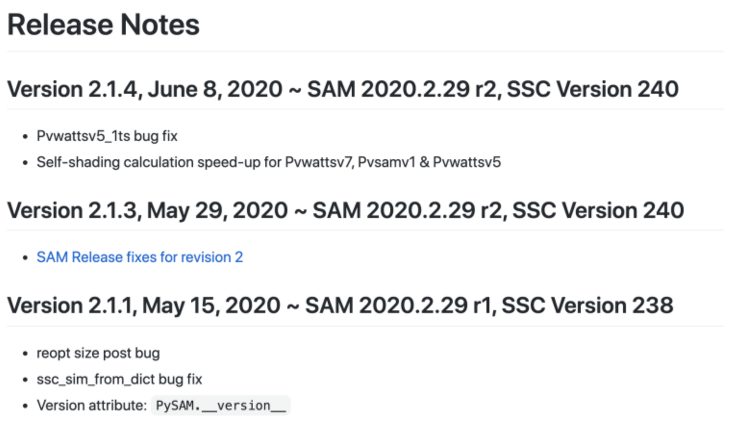
- Order
- Information needs to be passed from one to the next
- Assembled behind the scenes in SAM user interface

PySAM, and SAM's other software development kits, expose these unit modules so that they can be customized and embedded in software applications

PySAM does NOT contain all the features in the SAM GUI

PySAM Versions

Official



Release Notes

Version 2.1.4, June 8, 2020 ~ SAM 2020.2.29 r2, SSC Version 240

- Pvwattsv5_1ts bug fix
- Self-shading calculation speed-up for Pvwattsv7, Pvsamv1 & Pvwattsv5

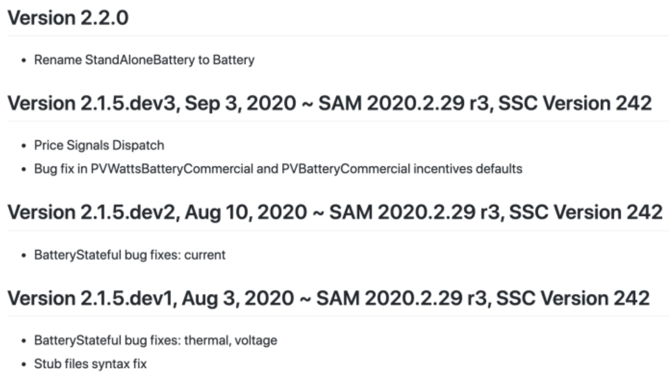
Version 2.1.3, May 29, 2020 ~ SAM 2020.2.29 r2, SSC Version 240

- SAM Release fixes for revision 2

Version 2.1.1, May 15, 2020 ~ SAM 2020.2.29 r1, SSC Version 238

- reopt size post bug
- ssc_sim_from_dict bug fix
- Version attribute: `PySAM.__version__`

Development



Version 2.2.0

- Rename StandAloneBattery to Battery

Version 2.1.5.dev3, Sep 3, 2020 ~ SAM 2020.2.29 r3, SSC Version 242

- Price Signals Dispatch
- Bug fix in PVWattsBatteryCommercial and PVBatteryCommercial incentives defaults

Version 2.1.5.dev2, Aug 10, 2020 ~ SAM 2020.2.29 r3, SSC Version 242

- BatteryStateful bug fixes: current

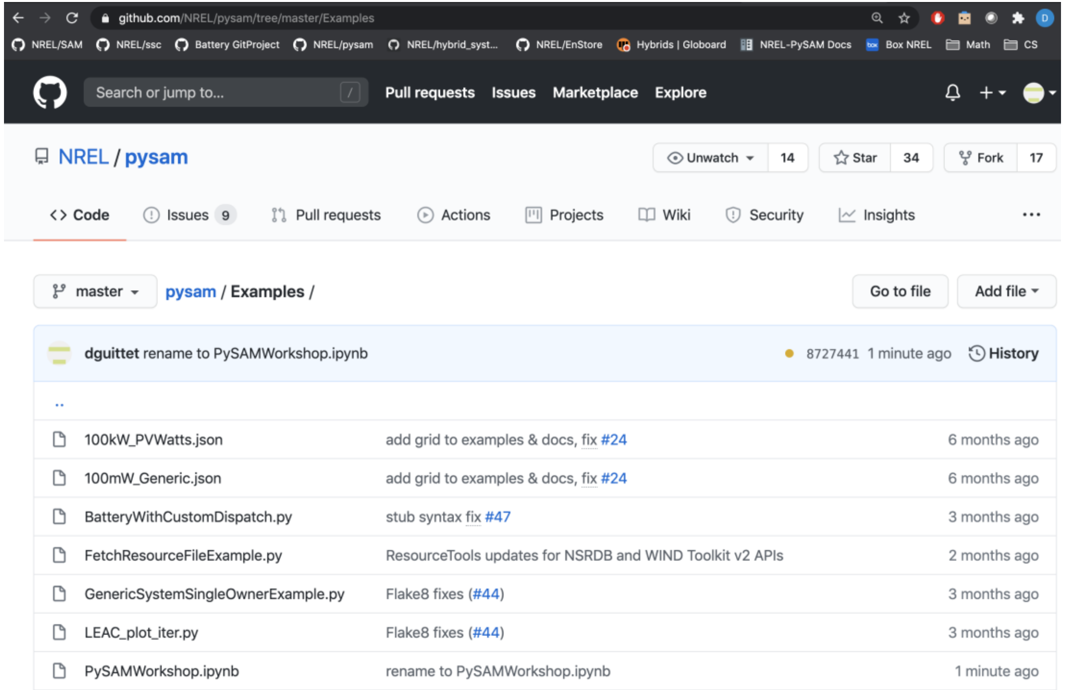
Version 2.1.5.dev1, Aug 3, 2020 ~ SAM 2020.2.29 r3, SSC Version 242

- BatteryStateful bug fixes: thermal, voltage
- Stub files syntax fix

New version of SAM and PySAM in mid-November

Workshop Notebook

https://github.com/NREL/pysam/blob/master/Examples/PySAMWorkshop.ipynb



dguittet rename to PySAMWorkshop.ipynb · 8727441 1 minute ago

..

100kW_PVWatts.json	add grid to examples & docs, fix #24	6 months ago
100mW_Generic.json	add grid to examples & docs, fix #24	6 months ago
BatteryWithCustomDispatch.py	stub syntax fix #47	3 months ago
FetchResourceFileExample.py	ResourceTools updates for NSRDB and WIND Toolkit v2 APIs	2 months ago
GenericSystemSingleOwnerExample.py	Flake8 fixes (#44)	3 months ago
LEAC_plot_iter.py	Flake8 fixes (#44)	3 months ago
PySAMWorkshop.ipynb	rename to PySAMWorkshop.ipynb	1 minute ago

1 Introduction

2 **How to Get Started**

3 Model Initialization

4 Getting Module Information

5 Detailed PV-Battery Commercial Example

6 Helper Functions

7 Questions / More Examples

Installation

Install Python or Anaconda, a Python distribution platform

- 64-bit Python 3.5-3.8 for Linux, Mac and Windows

```
pip install nrel-pysam
```

```
conda install -u nrel nrel-pysam nrel-pysam-stubs
```

- Note the name has NREL prefixed
- Nrel-pysam-stubs is automatically downloaded using pip

1 Introduction

2 How to Get Started

3 **Model Initialization**

4 Getting Module Information

5 Detailed PV-Battery Commercial Example

6 Helper Functions

7 Questions / More Examples

1 Introduction

2 How to Get Started

3 Model Initialization

4 **Getting Module Information**

5 Detailed PV-Battery Commercial Example

6 Helper Functions

7 Questions / More Examples

Getting Module Information

Explore the user interface:

- What technology and financial simulations are available
- Categories of inputs
- Inputs’ data requirements
- Interdependent inputs

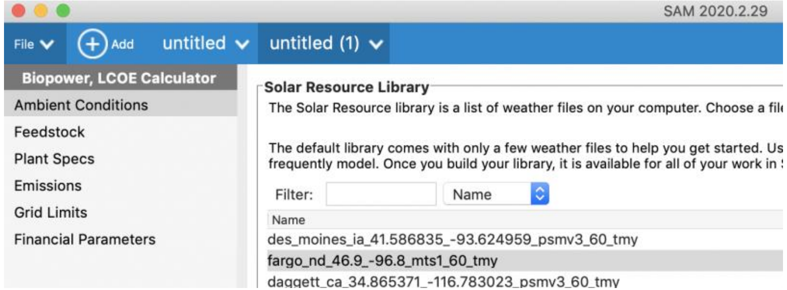
PySAM requires users to maintain consistency

- Changes in the correct order
- Data flow from one unit module to the next

Getting Module Information

- In SAM, a set of input pages is loaded to present all the inputs that are needed for the whole simulation.

- A list of all simulation configurations and their unit models in order can be found on the PySAM documentation site under “SAM Simulation Configurations"



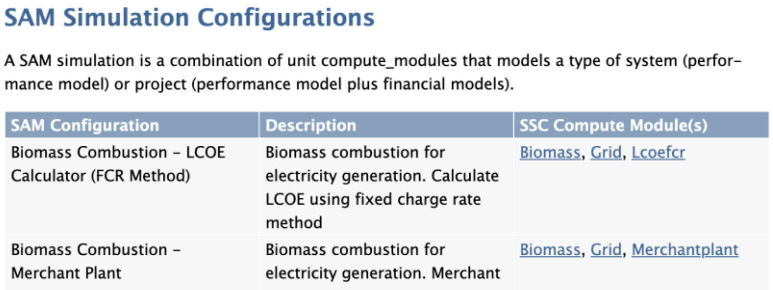
Solar Resource Library

The Solar Resource library is a list of weather files on your computer. Choose a fil

The default library comes with only a few weather files to help you get started. Us
frequently model. Once you build your library, it is available for all of your work in !

Filter: Name

Name
des_moines_ia_41.586835_-93.624959_psmv3_60_tmy
fargo_nd_46.9_-96.8_mts1_60_tmy
daggett_ca_34.865371_-116.783023_psmv3_60_tmy



SAM Simulation Configurations

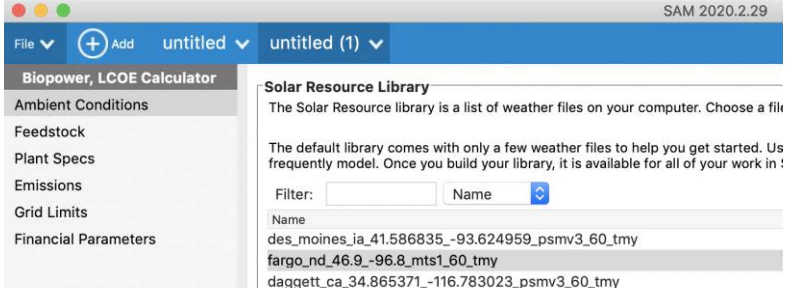
A SAM simulation is a combination of unit compute_modules that models a type of system (performance model) or project (performance model plus financial models).

SAM Configuration	Description	SSC Compute Module(s)
Biomass Combustion – LCOE Calculator (FCR Method)	Biomass combustion for electricity generation. Calculate LCOE using fixed charge rate method	Biomass, Grid, Lcoefcr
Biomass Combustion – Merchant Plant	Biomass combustion for electricity generation. Merchant	Biomass, Grid, Merchantplant

Getting Module Information

- For each unit module, PySAM inputs are categorized into groups. These groups roughly correspond to the SAM UI pages.

- But sometimes they don't.



Solar Resource Library

The Solar Resource library is a list of weather files on your computer. Choose a fil

The default library comes with only a few weather files to help you get started. Us
frequently model. Once you build your library, it is available for all of your work in :

Name
des_moines_ia_41.586835_-93.624959_psmv3_60_tmy
fargo_nd_46.9_-96.8_mts1_60_tmy
daggett_ca_34.865371_-116.783023_psmv3_60_tmy

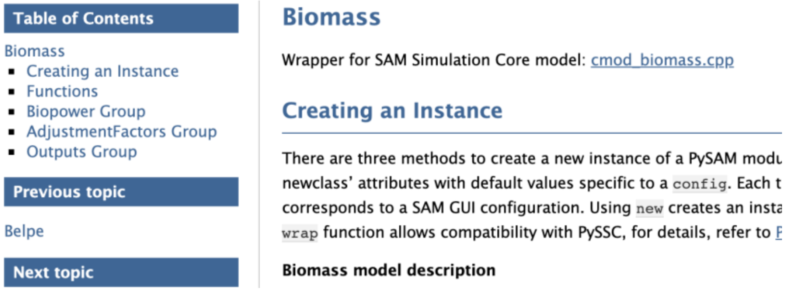


Table of Contents

Biomass
- Creating an Instance
- Functions
- Biopower Group
- AdjustmentFactors Group
- Outputs Group

Previous topic

Belpe

Next topic

Biomass

Wrapper for SAM Simulation Core model: cmod_biomass.cpp

Creating an Instance

There are three methods to create a new instance of a PySAM modu
newclass' attributes with default values specific to a `config`. Each t
corresponds to a SAM GUI configuration. Using `new` creates an insta
`wrap` function allows compatibility with PySSC, for details, refer to F

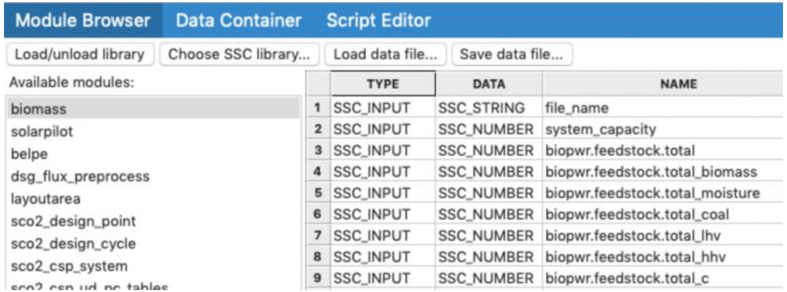
Biomass model description

Getting Module Information

- For each unit module, its inputs are categorized into groups. These groups roughly correspond to the SAM UI pages.

- But sometimes they don’t.

- SDKtool, variable tables in SSC source code



Available modules:

biomass
solarpilot
belpe
dsg_flux_preprocess
layoutarea
sco2_design_point
sco2_design_cycle
sco2_csp_system
sco2_csp_ud_pc_tables

	TYPE	DATA	NAME
1	SSC_INPUT	SSC_STRING	file_name
2	SSC_INPUT	SSC_NUMBER	system_capacity
3	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total
4	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_biomass
5	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_moisture
6	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_coal
7	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_lhv
8	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_hhv
9	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_c

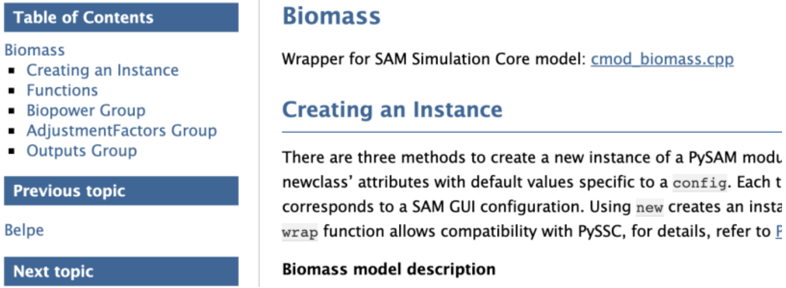


Table of Contents

Biomass
- Creating an Instance
- Functions
- Biopower Group
- AdjustmentFactors Group
- Outputs Group

Previous topic

Belpe

Next topic

Biomass

Wrapper for SAM Simulation Core model: cmod_biomass.cpp

Creating an Instance

There are three methods to create a new instance of a PySAM modu
newclass’ attributes with default values specific to a `config`. Each t
corresponds to a SAM GUI configuration. Using `new` creates an insta
`wrap` function allows compatibility with PySSC, for details, refer to F

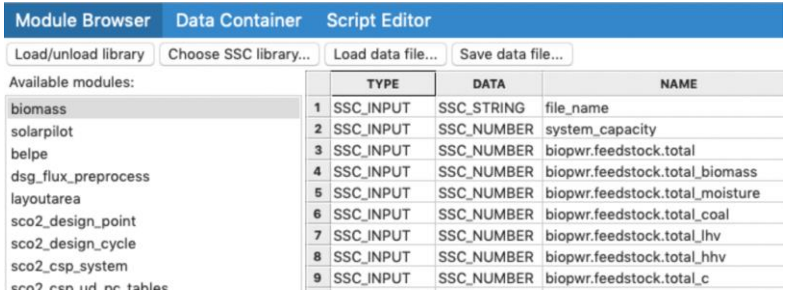
Biomass model description

Getting Module Information

- For each unit module, its inputs are categorized into groups. These groups roughly correspond to the SAM UI pages.

- But sometimes they don’t.

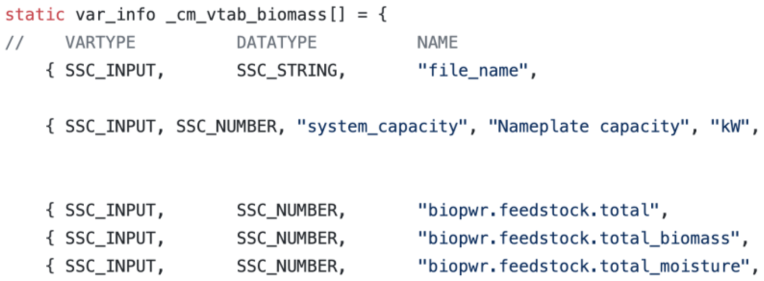
- SDKtool, variable tables in SSC source code



Available modules:

biomass
solarpilot
belpe
dsg_flux_preprocess
layoutarea
sco2_design_point
sco2_design_cycle
sco2_csp_system
sco2_csp_ud_pc_tables

	TYPE	DATA	NAME
1	SSC_INPUT	SSC_STRING	file_name
2	SSC_INPUT	SSC_NUMBER	system_capacity
3	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total
4	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_biomass
5	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_moisture
6	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_coal
7	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_lhv
8	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_hhv
9	SSC_INPUT	SSC_NUMBER	biopwr.feedstock.total_c

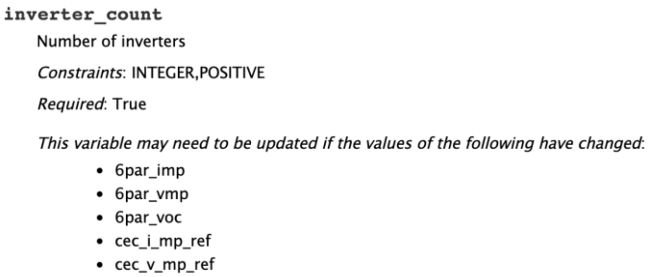


```
static var_info _cm_vtab_biomass[] = {
//    VARTYPE           DATATYPE          NAME
    { SSC_INPUT,        SSC_STRING,       "file_name",

    { SSC_INPUT, SSC_NUMBER, "system_capacity", "Nameplate capacity", "kW",


    { SSC_INPUT,        SSC_NUMBER,       "biopwr.feedstock.total",
    { SSC_INPUT,        SSC_NUMBER,       "biopwr.feedstock.total_biomass",
    { SSC_INPUT,        SSC_NUMBER,       "biopwr.feedstock.total_moisture",
```

Getting Module Information

Input consistency

- “Variable may need to be updated if the values of the following have changed”
- Intra-module input dependencies
- In SAM, automatically handled
- In PySAM, up to user



inverter_count

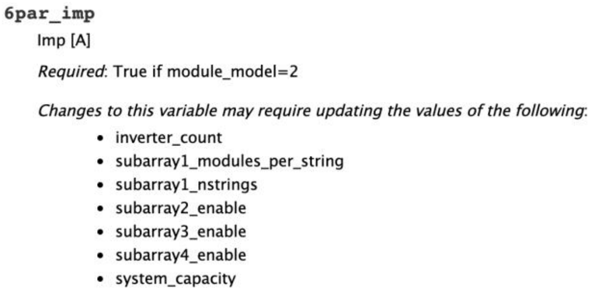
Number of inverters

Constraints: INTEGER,POSITIVE

Required: True

This variable may need to be updated if the values of the following have changed:

- 6par_imp
- 6par_vmp
- 6par_voc
- cec_i_mp_ref
- cec_v_mp_ref



6par_imp

Imp [A]

Required: True if module_model=2

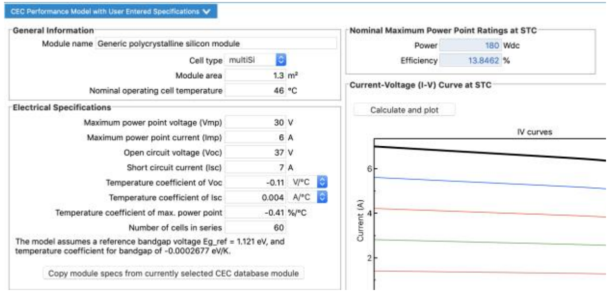
Changes to this variable may require updating the values of the following:

- inverter_count
- subarray1_modules_per_string
- subarray1_nstrings
- subarray2_enable
- subarray3_enable
- subarray4_enable
- system_capacity

Getting Module Information

Input consistency

- “Variable may need to be updated if the values of the following have changed”
- Intra-module input dependencies
- In SAM, automatically handled
- In PySAM, up to user


CEC Performance Model with User Entered Specifications
Calculate and plot
Copy module specs from currently selected CEC database module
IV curves
Current (A)
6
4
2

General Information

Module name: Generic polycrystalline silicon module
Cell type: multiSi
Module area: 1.3 m²
Nominal operating cell temperature: 46 °C

Electrical Specifications

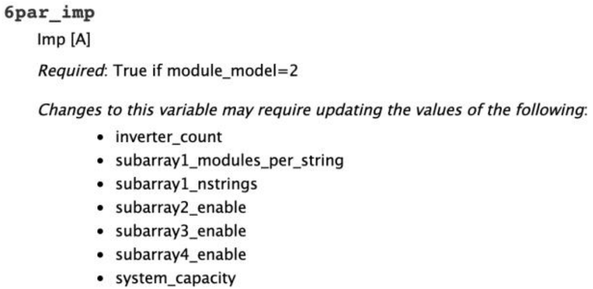
Maximum power point voltage (Vmp): 30 V
Maximum power point current (Imp): 6 A
Open circuit voltage (Voc): 37 V
Short circuit current (Isc): 7 A
Temperature coefficient of Voc: -0.11 V/°C
Temperature coefficient of Isc: 0.004 A/°C
Temperature coefficient of max. power point: -0.41 %/°C
Number of cells in series: 60

The model assumes a reference bandgap voltage Eg_ref = 1.121 eV, and temperature coefficient for bandgap of -0.0002677 eV/K.

Nominal Maximum Power Point Ratings at STC

Power: 180 Wdc
Efficiency: 13.8462 %

Current-Voltage (I-V) Curve at STC



`6par_imp`

Imp [A]

Required: True if module_model=2

Changes to this variable may require updating the values of the following:

- inverter_count
- subarray1_modules_per_string
- subarray1_nstrings
- subarray2_enable
- subarray3_enable
- subarray4_enable
- system_capacity

1 Introduction

2 How to Get Started

3 Model Initialization

4 Getting Module Information

5 **Detailed PV-Battery Commercial Example**

6 Helper Functions

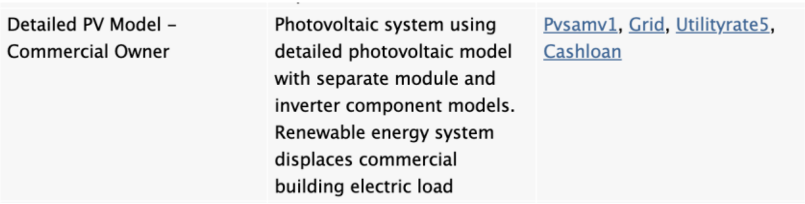
7 Questions / More Examples

Detailed PV-Battery - Commercial

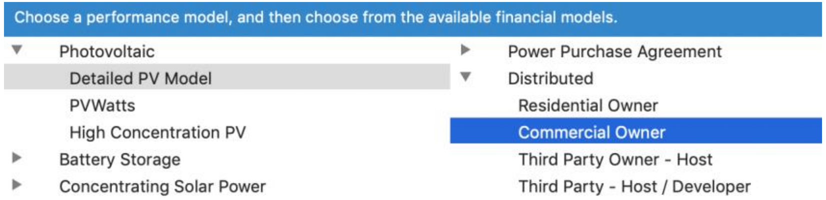
Using a set of PSM weather files for different years, calculate how the average net present value (NPV) of the default Detailed PV-Battery – Commercial owner system changes with the size of a four-hour battery.

lexington_or_45.446370_-119.687903_psmv3_30_1998.csv
lexington_or_45.446370_-119.687903_psmv3_30_1999.csv
lexington_or_45.446370_-119.687903_psmv3_30_2000.csv
lexington_or_45.446370_-119.687903_psmv3_30_2001.csv
lexington_or_45.446370_-119.687903_psmv3_30_2002.csv
lexington_or_45.446370_-119.687903_psmv3_30_2003.csv
lexington_or_45.446370_-119.687903_psmv3_30_2004.csv
lexington_or_45.446370_-119.687903_psmv3_30_2005.csv
lexington_or_45.446370_-119.687903_psmv3_30_2006.csv
lexington_or_45.446370_-119.687903_psmv3_30_2007.csv
lexington_or_45.446370_-119.687903_psmv3_30_2008.csv
lexington_or_45.446370_-119.687903_psmv3_30_2009.csv
lexington_or_45.446370_-119.687903_psmv3_30_2010.csv
lexington_or_45.446370_-119.687903_psmv3_30_2011.csv
lexington_or_45.446370_-119.687903_psmv3_30_2012.csv
lexington_or_45.446370_-119.687903_psmv3_30_2013.csv
lexington_or_45.446370_-119.687903_psmv3_30_2014.csv
lexington_or_45.446370_-119.687903_psmv3_30_2015.csv
lexington_or_45.446370_-119.687903_psmv3_30_2016.csv
lexington_or_45.446370_-119.687903_psmv3_30_2017.csv
lexington_or_45.446370_-119.687903_psmv3_30_2018.csv

Detailed PV-Battery - Commercial



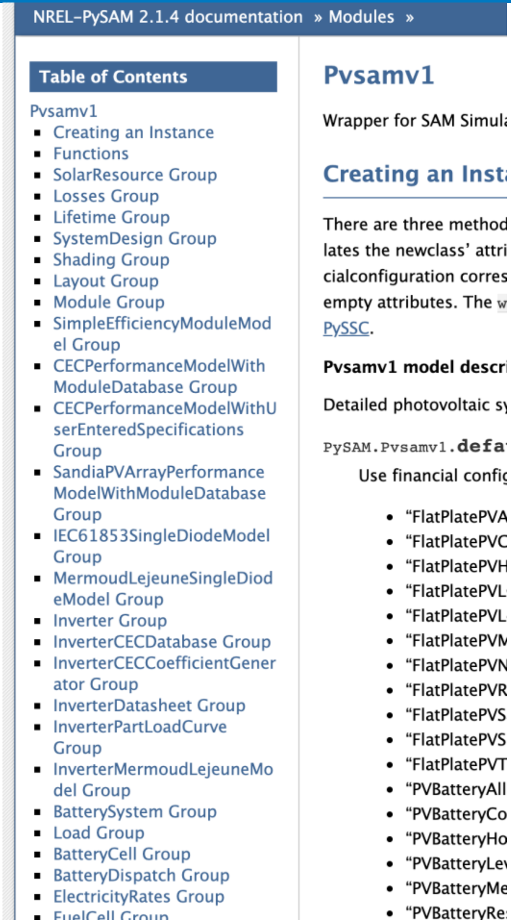
Detailed PV Model – Commercial Owner	Photovoltaic system using detailed photovoltaic model with separate module and inverter component models. Renewable energy system displaces commercial building electric load	Pvsamv1, Grid, Utilityrate5, Cashloan



Choose a performance model, and then choose from the available financial models.

Pvsamv1	Photovoltaic (detailed)	Detailed photovoltaic system model with separate components for module and inverter
Grid	Grid	Interconnect and Curtailment limits
Utilityrate5	Residential, Commercial, Third Party, Host Developer	Retail electricity bill calculator
Cashloan	Residential and Commercial	Financial model for residential and commercial behind-the-meter projects

Detailed PV-Battery - Commercial



NREL-PySAM 2.1.4 documentation » Modules »

Table of Contents

Pvsamv1

- Creating an Instance
- Functions
- SolarResource Group
- Losses Group
- Lifetime Group
- SystemDesign Group
- Shading Group
- Layout Group
- Module Group
- SimpleEfficiencyModuleModel Group
- CECPerformanceModelWithModuleDatabase Group
- CECPerformanceModelWithUserEnteredSpecifications Group
- SandiaPVArrayPerformanceModelWithModuleDatabase Group
- IEC61853SingleDiodeModel Group
- MermoudLejeuneSingleDiodeModel Group
- Inverter Group
- InverterCECDatabase Group
- InverterCECCoefficientGenerator Group
- InverterDatasheet Group
- InverterPartLoadCurve Group
- InverterMermoudLejeuneModel Group
- BatterySystem Group
- Load Group
- BatteryCell Group
- BatteryDispatch Group
- ElectricityRates Group
- FuelCell Group

Pvsamv1

Wrapper for SAM Simul

Creating an Inst

There are three method lates the newclass' attri cialconfiguration corres empty attributes. The w PySSC.

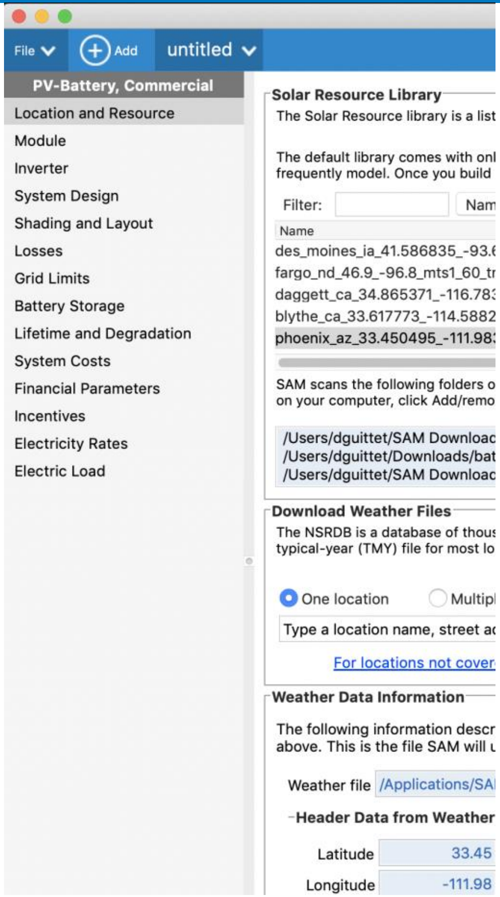
Pvsamv1 model descri

Detailed photovoltaic sy

`PySAM.Pvsamv1.defa`

Use financial confic

- "FlatPlatePVA
- "FlatPlatePVC
- "FlatPlatePVH
- "FlatPlatePVL
- "FlatPlatePVL
- "FlatPlatePVM
- "FlatPlatePVN
- "FlatPlatePVR
- "FlatPlatePVS
- "FlatPlatePVS
- "FlatPlatePVT
- "PVBatteryAll
- "PVBatteryCo
- "PVBatteryHo
- "PVBatteryLe
- "PVBatteryMe
- "PVBatteryRe



Solar Resource Library

The Solar Resource library is a list

The default library comes with onl frequently model. Once you build

Filter: Nam

Name
des_moines_ia_41.586835_-93.6
fargo_nd_46.9_-96.8_mts1_60_tr
daggett_ca_34.865371_-116.783
blythe_ca_33.617773_-114.5882
phoenix_az_33.450495_-111.983

SAM scans the following folders o on your computer, click Add/remo

/Users/dguittet/SAM Download
/Users/dguittet/Downloads/bat
/Users/dguittet/SAM Download

Download Weather Files

The NSRDB is a database of thous typical-year (TMY) file for most lo

● One location ○ Multipl

Type a location name, street ac

For locations not cover

Weather Data Information

The following information descr above. This is the file SAM will u

Weather file /Applications/SA

Header Data from Weather

Latitude 33.45

Longitude -111.98

1 Introduction

2 How to Get Started

3 Model Initialization

4 Getting Module Information

5 Detailed PV-Battery Commercial Example

6 **Helper Functions**

7 Questions / More Examples

Helper Functions

ResourceTools

- TMY_CSV_to_solar_data
  - *TMY csv file as 'solar_resource_data' dictionary for Pvsamv1, Pvwattsv5, Pvwattsv7, …*
- SRW_to_wind_data
  - *SRW csv file as 'wind_resource_data’ dictionary for Windpower*
- URDBv7_to_ElectricityRates
  - *Utility Rate Database API version 7 response as Utilityrate5 inputs*
- FetchResourceFiles
  - *Downloader for National Solar Radiation Database and Wind Toolkit*

BatteryTools

- battery_model_sizing
  - *Modifies model for desired power and capacity*

1. Introduction
2. How to Get Started
3. Model Initialization
4. Getting Module Information
5. Detailed PV-Battery Commercial Example
6. Helper Functions
7. **Questions / More Examples**