QCSchema

circa 20 May 2026, v0.50.0

Legend

ResultSchemaClass
field: brief field description
fixed_value_field: **True**
required_field: description
example_contents
field_with_options: shows choices for values. In this case: a string, a list of schema, a dictionary of schema (dummy key), and a rare-to-use choice.
- gradient
- [AtomicResult]
- {sub id: ManyBodyResult }
-

field_holding subschema: description

InputSchemaClass
field: description
another_field: description

another_field: description

V1_InputSchemaClass
field: description
another_field: note that fields of this class left-align with outer child class **ResultSchemaClass** since in v1, **<>Result** inherits from **<>Input**.
tentative_or_inherited_field: description

*light blue dotted – **input** schema*
*dark blue solid – **output** schema*

ACCESS QCSchema v1 in QCElemental
(PYDANTIC v1 API; c. 2019 field layout)

```
# longstanding until v0.70
> from qcelemental.models import Molecule

# longstanding until v0.70 but remove ASAP!
> from qcelemental.models.molecule import Molecule

# v0.50a1 until v1.0
> from qcelemental.models.v1 import Molecule
```

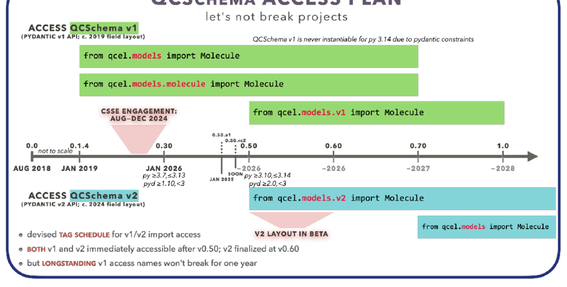
ACCESS QCSchema v2 in QCElemental
(PYDANTIC v2 API; c. 2024 field layout)

```
# v0.50a1 until v0.60 (beta layout)
# v0.60a1 onwards (final layout)
> from qcelemental.models.v2 import Molecule

# v0.70a1 onwards
> from qcelemental.models import Molecule
```

same info, two representations

QCSchema ACCESS PLAN
let's not break projects



RUN QCEngine CMDLINE

```
# run with full local resources. use run-procedure for opt/td/mbe in v1
sh> qcengine run cmscode qcskin

# run with restricted resources: 4 cores, 4 GiB
sh> qcengine run cmscode qcskin  --ncores 4 --memory 4

# run with restricted resources: 4 cores, 4 GiB. available to sub-qcng calls
sh> cat qcengine.yaml
all:
  hostname_pattern: "*"
  ncores: 4
  memory: 4
sh> qcengine run cmscode qcskin
```

RUN QCEngine PyAPI

```
# run with full local resources. use compute_procedure for opt/td/mbe in v1
> qcengine.compute(qcskin, cmscode)

# run with restricted resources: 4 cores, 4 GiB
> qcengine.compute(qcskin, cmscode, task_config({"nnodes": 4, "memory": 4}))

# run with restricted resources: 4 cores, 4 GiB. available to sub-qcng calls
sh> cat qcengine.yaml
all:
  hostname_pattern: "*"
  nnodes: 4
  memory: 4
> qcengine.compute(qcskin, cmscode)
```

Main Models in QCSchema (v2 names)

				v1 inherited	v2
AtomicResult	OptimizationResult	TorsionDriveResult	ManyBodyResult		
AtomicInput	OptimizationInput	TorsionDriveInput	ManyBodyInput	v1	v2
AtomicSpecification	OptimizationSpecification	TorsionDriveSpecification	ManyBodySpecification	v1	–
AtomicKeywords	OptimizationKeywords	TorsionDriveKeywords	ManyBodyKeywords	–	–
AtomicProtocols	OptimizationProtocols	TorsionDriveProtocols	ManyBodyProtocols	–	–
AtomicProperties	OptimizationProperties	TorsionDriveProperties	ManyBodyProperties	–	–
WavefunctionProperties				–	–
			Molecule	v1	v2
			BasisSet	v1	v2
			FailedOperation	–	v2

schema_version present?

Version Handling in QCEngine

```
> *Result v1 = qcengine.compute( *Input v1 )
> *Result v2 = qcengine.compute( *Input v2 )
> *Result v2 = qcengine.compute( *Input v1 , return_version=2)
> *Result v1 = qcengine.compute( *Input v2 , return_version=1)
```

no v1 models for py≥3.14

object, unless `return_dict=True` *object or dict*

QC Job Models

Comparing v1 & v2 QCSchema

Trace the same data horizontally between v1/v2 pairs and, to some degree, between schema.

any SP program in QCEngine: adcc, cfour, gamess, entos, mrchem, qchem, torchani, terachem, nwchem, xtb, turbomole, mace, molpro, psi4, s-dftd3, mopac, aimnet2, dftd4, rdkit, openmm, mp2d, mctc-gcp

any optimizer procedure in QCEngine: optking, geometric, nwchemdriver, berny

any TD procedure in QCEngine: torsiondrive

any MBE procedure in QCEngine: qcmanybody

RUN QCSchema v1

```
> qcng.compute( AtomicInput , [ any program in QCEngine ] )
```

RUN QCSchema v2

```
> qcengine.compute(atin, [ any program in QCEngine ])
> qcengine.compute(atin) (nyi in QCEngine)
```

QCSchema v1 — convert_v(v) — QCSchema v2

AtomicResult
schema_version: **1**
provenance: creator, version, & RT info (QCElemental, 0.28.0)
id: tracker for databases
extras: free-form storage
molecule: Molecule

AtomicInput
schema_version: **1**
provenance: creator, version, & RT info (QCElemental, 0.28.0)
id: tracker for databases
extras: free-form storage
molecule: Molecule
specification: *how* to run single-point

AtomicSpecification
program: any SP in QCEngine
protocols: customize return layout

AtomicResultProtocols
stdout: save text output
wavefunction: save orbital info
native_files: save raw prog. files
error_correction: auto-edit&rerun

AtomicProtocols
stdout: save text output
wavefunction: save orbital info
native_files: save raw prog. files
error_correction: auto-edit&rerun

extras: free-form storage
driver: single-point derivative (E G H)
- energy
- gradient
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pV5Z
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc:[4,1])

success: **True**
provenance: *(overwrites input)* (xtb, 1.0.0, psinet, CPU E5-2630)
stdout: program's text output for humans
stderr: anything in error channel
native_files: requested raw program files
extras: *(merges with input)*
molecule: *(overwrites inp w/result frame)*
return_result: single-point derivative (E G H)
properties: key results for single-point

AtomicResultProperties (abr.)
calcinfo_natom
calcinfo_nmo
nuclear_repulsion_energy
return_energy: E
return_gradient: G
scf_total_*egh*: ref. energy, gradient, etc.
*mtd*_correlation_energy: post-scf energy
*mtd*_iterations: count to convergence
*mtd*_mpole_moment: dipole etc. array
wavefunction: orbital-level results

WavefunctionProperties (abr.)
basis: BasisSet
restricted: alpha == beta
orbitals_a/b
density_a/b
fock_a/b
eigenvalues_a/b
occupations_a/b

error: hold ComputeError for success=F

AtomicProperties (abr.)
calcinfo_natom
calcinfo_nmo
nuclear_repulsion_energy
return_energy: E
return_gradient: G
scf_total_*egh*: ref. energy, gradient, etc.
*mtd*_correlation_energy: post-scf energy
*mtd*_iterations: count to convergence
*mtd*_mpole_moment: dipole etc. array
wavefunction: orbital-level results

WavefunctionProperties (abr.)
basis: BasisSet
restricted: alpha == beta
orbitals_a/b
density_a/b
fock_a/b
eigenvalues_a/b
occupations_a/b

success: **True**
provenance: creator, version, & RT info (xtb, 1.0.0, psinet, CPU E5-2630)
id: tracker for databases
stdout: program's text output for humans
stderr: anything in error channel
native_files: requested raw program files
extras: free-form storage
molecule: orientation & frame for results
return_result: single-point derivative (E G H)
properties: key results for single-point

RUN QCSchema v1

```
> qcengine.compute_procedure(optin, [ any optimizer in QCEngine ])
```

RUN QCSchema v2

```
> qcengine.compute(optin, [ any optimizer in QCEngine ])
> qcengine.compute(optin) (nyi in QCEngine)
```

QCSchema v1 — convert_v(v) — QCSchema v2

OptimizationResult
schema_version: **1**
provenance: creator name and version info (QCElemental, 0.28.0)
id: tracker for databases
extras: free-form storage
initial_molecule: Molecule

OptimizationInput
schema_version: **1**
provenance: creator name and version info (QCElemental, 0.28.0)
id: tracker for databases
extras: free-form storage
initial_molecule: Molecule
specification: how to run *geometry opts*

OptimizationSpecification
program: any optimizer in QCEngine
protocols: customize opt return layout

OptimizationProtocols
trajectory: save fewer opt steps

keywords: knobs in the geo opt program (coordsys: tric, program: [ ])
input_specification: how to run *gradients* for opt

QCInputSpecification
schema_version: **1**
extras: free-form storage
driver: single-point derivative (G)
- energy
- gradient
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pV5Z
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc: [4,1])

success: True
provenance: *(overwrites input)* (geometric, 1.0.2, psinet, CPU E5-2630)
stdout: Opt program's text output for humans
stderr: anything in error channel
extras: *(merges with input)*
final_molecule: optimized geom. for results
energies: energy for opt steps [E]
trajectory: full results for opt steps [AtomicResult]
error: hold ComputeError for success=F

OptimizationProtocols
trajectory_results: save any opt steps

extras: free-form storage
keywords: knobs in the geo opt program (coordsys: tric, maxiter: 100)
specification: how to run *gradients* for opt
- single-point

AtomicSpecification
program: any SP in QCEngine
protocols: customize return layout

AtomicProtocols
stdout: save text output
wavefunction: save orbital info
native_files: save raw prog. files
error_correction: auto-edit&rerun

extras: free-form storage
driver: single-point derivative
- energy
- gradient (G)
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pV5Z
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc: [4,1])
- MBE
  ManyBodySpecification

success: **True**
provenance: creator, version, & RT info (geometric, 1.0.2, psinet, CPU E5-2630)
id: tracker for databases
stdout: Opt program's text output for humans
stderr: anything in error channel
native_files: requested raw program files *(nyi)*
extras: free-form storage
final_molecule: optimized geom. for results
properties: key results for optimization

OptimizationProperties
nuclear_repulsion_energy
return_energy: E
return_gradient: G
optimization_iterations: count to geo conv.
final_rms_force: measure of convergence
final_max_force: largest remaining gradient
final_rms_displacement: meas. avg. geo chg.
final_max_displacement: meas. geo chg.
opt_coordinate_system: intrnl, Cart., tric *(nyi)*
trajectory_properties: key results for opt steps
- [AtomicProperties]
- [ManyBodyProperties]

trajectory_results: full results for opt steps
- [AtomicResult]
- [ManyBodyResult]

QCSchema v1 — convert_v(v) — QCSchema v2

TorsionDriveResult
schema_version: **1**
provenance: creator name and version info (QCElemental, 0.28.0)
extras: free-form storage
initial_molecule: [Molecule]

TorsionDriveInput
schema_version: **2**
provenance: creator name and version info (QCElemental, 0.28.0)
id: tracker for databases
initial_molecule: [Molecule]
specification: *how* to run *torsion drive* opts

TorsionDriveSpecification
program: any TD in QCEngine
protocols: customize TD return layout

TorsionDriveProtocols
scan_results: save any optimization data

extras: free-form storage
keywords: knobs in the TD program

TorsionDriveKeywords
dihedrals: dimensions to scan
dihedral_ranges: limits of scan
grid_spacing: IE dihedral steps of scan
energy_decrease_thresh: point trigger
energy_upper_limit: new point trigger

specification: how to run *geometry opts*

OptimizationSpecification
program: any optimizer in QCEngine
protocols: customize opt return layout

OptimizationProtocols
trajectory: save fewer opt steps

keywords: knobs in the geo opt program (coordsys: tric, maxiter: 100)
input_specification: how to run *gradients* for opt

QCInputSpecification
schema_version: **1**
extras: free-form storage
driver: single-point derivative
- energy
- gradient (G)
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pVQZ
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc: [4,1])

success: True
provenance: *(overwrites input)* (torsiondrive, 1.1.0, psinet, CPU E5-2630)
stdout: TD program's text output for humans
stderr: anything in error channel
extras: *(merges with input)*
final_molecules: optimized molecules for angles {dihedral id: Molecule }
final_energies: optimized energy for angles {dihedral id: E }
optimization_history: full results for all opts {dihedral id: [OptimizationResult] }
error: hold ComputeError for success=F

TDKeywords
dihedrals: dimensions to scan
dihedral_ranges: limits of scan
grid_spacing: IE dihedral steps of scan
energy_decrease_thresh: point trigger
energy_upper_limit: new point trigger

optimization_spec: how to run *geometry opts*

OptimizationSpecification
schema_version: **1**
procedure: any optimizer in QCEngine
protocols: customize opt return layout

OptimizationProtocols
trajectory: save fewer opt steps

keywords: knobs in the geo opt program (coordsys: tric, program: [ ])
input_specification: how to run *gradients* for opt

QCInputSpecification
schema_version: **1**
extras: free-form storage
driver: single-point derivative
- energy
- gradient (G)
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pV5Z
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc: [4,1])

success: True
provenance: *(overwrites input)* (torsiondrive, 1.1.0, psinet, CPU E5-2630)
stdout: TD program's text output for humans
stderr: anything in error channel
extras: *(merges with input)*
final_molecules: optimized molecules for angles {dihedral id: Molecule }
final_energies: optimized energy for angles {dihedral id: E }
optimization_history: full results for all opts {dihedral id: [OptimizationResult] }
error: hold ComputeError for success=F

TorsionDriveProperties
calcinfo_ngrid: count dihedral scan pts optimized

scan_properties: key results for best opt steps {dihedral id: OptimizationProperties }
scan_results: full results for all opts {dihedral id: [OptimizationResult] }

success: **True**
provenance: creator, version, & RT info (torsiondrive, 1.1.0, psinet, CPU E5-2630)
id: tracker for databases
stdout: TD program's text output for humans
stderr: anything in error channel
native_files: requested raw program files *(nyi)*
extras: free-form storage
final_molecules: optimized molecules for angles {dihedral id: Molecule }

QCSchema v1 — convert_v(v) — QCSchema v2

live at GH:MolSSI/QCManyBody repo

ManyBodyResult
schema_version: **1**
input_data: record of starting conditions

ManyBodyInput
schema_version: **1**
extras: free-form storage
molecule: Molecule
specification: *how* to run MBE single-point

ManyBodySpecification
schema_version: **1**
protocols: customize MBE return layout

ManyBodyProtocols
component_results: save any sub data

extras: free-form storage
driver: MBE target derivative
- energy
- gradient (E G H)
- Hessian

keywords: knobs in the MBE program

ManyBodyKeywords
schema_version: **1**
bsse_type: any combination of
- nocp/mbe
- cp/ssfc
- vmfc

embedding_charges
return_total_data: IE or total
levels:
- {2: hi, 4: lo}
- {1: hi, 2: md, supersystem: lo}

max_nbody: truncate MBE early
supersystem_ie_only: no MBE analysis

specification: how to run MBE *subsystems*
- single

AtomicSpecification*
schema_version: **1**
program: any SP in QCEngine
protocols: customize return layout

AtomicResultProtocols
stdout: save text output
wavefunction: save orbital info
native_files: save raw prog. files
error_correction: auto-edit&rerun

extras: free-form storage
driver: single-point derivative
- energy
- gradient (E G H)
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pV5Z
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc:[4,1])
- mapping
  {hi: AtomicSpecification, lo: AtomicSpecification }

success: True
provenance: creator, version, & RT info (QCManyBody, 0.3.0, psinet, CPU E5-2630)
id: tracker for databases
stdout: program's text output for humans
stderr: anything in error channel
extras: free-form storage
return_result: MBE derivative (E G H)
properties: key results for MBE

ManyBodyResultProperties
schema_version: 1
calcinfo_nfr: *M*
return_energy: E
return_gradient: G
bsse_corrected_interaction_egh_through_m_body:
- defined as E^{m}_{IE}
- defined for all *bsse* as bsse_type allows
- defined for all *egh* as driver allows
- def. for all *m* as max_nbody/levels allows

bsse_corrected_total_egh_through_m_body:
- defined as E^{m}

bsse_corrected_m_body_contribution_to_egh:
- defined as $E^{(m)}$

component_properties: key results for subsys {cluster id: AtomicResultProperties }
component_results: full results for subsystems {cluster id: AtomicResult }

ManyBodyProtocols
cluster_results: save any sub data

extras: free-form storage
driver: MBE target derivative
- energy
- gradient (E G H)
- Hessian

keywords: knobs in the MBE program

ManyBodyKeywords
bsse_type: any combination of
- nocp/mbe
- cp/ssfc
- vmfc

embedding_charges
return_total_data: IE or total
levels:
- {2: hi, 4: lo}
- {1: hi, 2: md, supersystem: lo}

max_nbody: truncate MBE early
supersystem_ie_only: no MBE analysis

specification: how to run MBE *subsystems*
- single

AtomicSpecification
program: any SP in QCEngine
protocols: customize return layout

AtomicProtocols
stdout: save text output
wavefunction: save orbital info
native_files: save raw prog. files
error_correction: auto-edit&rerun

extras: free-form storage
driver: single-point derivative
- energy
- gradient (E G H)
- Hessian
- properties

model: model chemistry or FF
method: B3LYP-D3, CCSD(T)
basis: 6-31G*, cc-pV5Z
keywords: knobs in SP program (scftyp: uhf, cc_conv: 7, docc:[4,1])
- mapping
  {hi: AtomicSpecification, lo: AtomicSpecification }

success: **True**
provenance: creator, version, & RT info (QCManyBody, 0.5.0, psinet, CPU E5-2630)
id: tracker for databases
stdout: program's text output for humans
stderr: anything in error channel
native_files: requested raw program files
extras: free-form storage
molecule: orientation & frame for results
return_result: MBE derivative (E G H)
properties: key results for MBE

ManyBodyProperties
calcinfo_nfr: *M*
return_energy: E
return_gradient: G
bsse_corrected_interaction_egh_through_m_body:
- defined as E^{m}_{IE}
- defined for all *bsse* as bsse_type allows
- defined for all *egh* as driver allows
- def. for all *m* as max_nbody/levels allows

bsse_corrected_total_egh_through_m_body:
- defined as E^{m}

bsse_corrected_m_body_contribution_to_egh:
- defined as $E^{(m)}$

cluster_properties: key results for subsystems {cluster id: AtomicProperties }
cluster_results: full results for subsystems {cluster id: AtomicResult }

Listed below are the **schema_name** *values for major models in the respective columns above. Changes bolded.*

qcschema_output
qcschema_input

qcschema_atomic_result
qcschema_atomic_input
qcschema_atomic_specification
qcschema_atomic_protocols
qcschema_atomic_properties
qcschema_wavefunction_properties

qcschema_optimization_output
qcschema_optimization_input

qcschema_input

qcschema_optimization_result
qcschema_optimization_input
qcschema_optimization_specification
qcschema_optimization_protocols
qcschema_atomic_specification
qcschema_optimization_properties

qcschema_torsion_drive_output
qcschema_torsion_drive_input

qcschema_optimization_specification
qcschema_input

qcschema_torsion_drive_result
qcschema_torsion_drive_input
qcschema_torsion_drive_specification
qcschema_torsion_drive_protocols
qcschema_torsion_drive_keywords
qcschema_optimization_specification
qcschema_atomic_specification

qcschema_manybodyoutput
qcschema_manybodyinput
qcschema_manybodyspecification

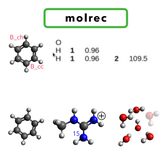
qcschema_manybodykeywords
qcschema_atomicspecification
qcschema_manybodyproperties

qcschema_many_body_output
qcschema_many_body_input
qcschema_many_body_specification
qcschema_many_body_protocols
qcschema_many_body_keywords
qcschema_atomic_specification
qcschema_many_body_properties

Molecule Models

Comparing v1 & v2 QCSchema

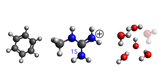
molrec



Molecule ("molrec")
schema_version: **psi4**
provenance: creator info
name: user label
comment: further user text
elem: elemental symb. for atoms
geom: Cartesian coord. in units
molecular_charge: net elst. charge
molecular_multiplicity: total int mult.
mass: per atom atomic masses
real: per atom real vs. ghost/virtual
elbl: further per atom labels
elez: per atom atomic numbers
elea: per atom isotope #
fragment_separators: gr'p atoms idx
fragment_charges: partitioned chg
fragment_multiplicities: ptn'd int mult
fix_com: conserve input COM
fix_orientation: conserve input frame
fix_symmetry: use lesser point group
connectivity: bond info (unused)
units: Bohr or Ångstrom for **geom**
input_units_to_au: conversion factor
geom_unsettled: replaces **geom** for Z-matrices or Cartesian w/variables
variables: names & vals, complete?

***not QCSchema** but parseable in QCElemental*

Molecule v2

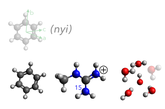


Molecule
schema_version: **2** *(used w/QCSk v1)*
provenance: creator info
id: tracker for databases
extras: free-form storage
name: user label
comment: further user text
symbols: elemental symb. for atoms
geometry: Cartesian coord. in Bohr
molecular_charge: net elst. charge
molecular_multiplicity: total mult.
masses: per atom atomic masses
real: per atom real vs. ghost/virtual
atom_labels: further per atom labels
atomic_numbers: per atom numbers
mass_numbers: per atom isotope #
fragments: group atoms into fragmts
fragment_charges: partitioned chg
fragment_multiplicities: ptn'd mult
fix_com: conserve input COM
fix_orientation: conserve input frame
fix_symmetry: use lesser point group
connectivity: bond info
validated: passed physics checks
identifiers: registration/classification

Identifiers (abr.)
molecular_formula
smiles
inchi
canonical_isomeric_smiles
canonical_smiles
pubchem_cid
pubchem_conformerid

qcschema_molecule

Molecule v3



Molecule
schema_version: **3** *(used w/QCSk v2)*
provenance: creator info
id: tracker for databases
extras: free-form storage
name: user label
comment: further user text
symbols: elemental symb. for atoms
geometry: Cartesian coord. in Bohr
molecular_charge: net elst. charge
molecular_multiplicity: total mult.
masses: per atom atomic masses
real: per atom real vs. ghost/virtual
atom_labels: further per atom labels
atomic_numbers: per atom numbers
mass_numbers: per atom isotope #
abc: periodic box vectors *(nyi)*
fragments: group atoms into fragmts
fragment_charges: partitioned chg
fragment_multiplicities: ptn'd mult
fix_com: conserve input COM
fix_orientation: conserve input frame
fix_symmetry: use lesser point group
connectivity: bond info
validated: passed physics checks
identifiers: registration/classification

Identifiers (abr.)
molecular_formula
smiles
inchi
canonical_isomeric_smiles
canonical_smiles
pubchem_cid
pubchem_conformerid

qcschema_molecule

Basis Set Models

Comparing v1 & v2 QCSchema

BasisSet
schema_version: **1**
name: BSE standard name, e.g., cc-pVDZ
description: text of characteristics
nbf: basis function count
atom_map: center & mol. correspondence
center_data: merge for all atoms/centers {center_id:
BasisCenter
electron_shells: elec. shells for this center [
ElectronShell
angular_momentum: *d*=[2]; *sp*=[0,1]
harmonic_type: shell repr. for AM≥2
- spherical
- cartesian

exponents: exp. for contracted shell
coefficients: nested list of coeff. Outer generalized; inner segmented.
]
ecp_electrons: #e^- replaced by field pots
ecp_potentials: field pots for this center [
ECPPotential
angular_momentum: *d*=[2]; *sp*=[0,1]
ecp_type:
- scalar
- spinorbit

r_exponents: exp. of *r* term
gaussian_exponents: exp. of e^{-r^2} term
coefficients: nested list of coeff. Outer generalized; inner segmented.
]
}

qcschema_basis_set

BasisSet
schema_version: **2**
name: BSE standard name, e.g., cc-pVDZ
description: text of characteristics
nbf: basis function count
atom_map: center & mol. correspondence
center_data: merge for all atoms/centers {center_id:
BasisCenter
electron_shells: elec. shells for this center [
ElectronShell
angular_momentum: *d*=[2]; *sp*=[0,1]
harmonic_type: shell repr. for AM≥2
- spherical
- cartesian

exponents: exp. for contracted shell
coefficients: nested list of coeff. Outer generalized; inner segmented.
normalization_scheme: lbl conv. *(nyi)*
- bse *(nyi)*
- cca *(nyi)*

]
ecp_electrons: #e^- replaced by field pots
ecp_potentials: field pots for this center [
ECPPotential
angular_momentum: *d*=[2]; *sp*=[0,1]
ecp_type:
- scalar
- spinorbit

r_exponents: exp. of *r* term
gaussian_exponents: exp. of e^{-r^2} term
coefficients: nested list of coeff. Outer generalized; inner segmented.
]
}

qcschema_basis_set

Error Handling Models

Comparing v1 & v2 QCSchema

FailedOperation
input_data: dict record of starting conditions
- AtomicInput
- OptimizationInput
- TorsionDriveInput
- ManyBodyInput

success: False
id: tracker for databases
extras: free-form storage
error: components to build error object

ComputeError
error_type: classification
- Input: bad user param, unrecoverable
- Resource: insufficient memory, cores, disk; likely recoverable on diff. node
- Random: likely recoverable from segfault, disk I/O
- Unknown: cannot classify

error_message: user text
extras: free-form storage

FailedOperation
schema_version: **2**
input_data: record of starting conditions
- AtomicInput
- OptimizationInput
- TorsionDriveInput
- ManyBodyInput

success: **False**
id: tracker for databases
extras: free-form storage
error: components to build error object

ComputeError
error_type: classification
- Input: bad user param, unrecoverable
- Resource: insufficient memory, cores, disk; likely recoverable on diff. node
- Random: likely recoverable from segfault, disk I/O
- Unknown: cannot classify

error_message: user text
extras: free-form storage

qcschema_failed_operation

QCEngine Return

Handling v1 & v2 QCSchema, Success & Failure Py314, Dict & Model

Return of qcengine.compute()

good_calc?	raise_error	return_dict	output
T	F (def)	F (def)	``AtomicResult`` object
T	T	F (def)	``AtomicResult`` object
T	T	T	dict of ``AtomicResult``
T	F (def)	T	dict of ``AtomicResult``
F	F (def)	F (def)	``FailedOperation`` object
F	T	F (def)	raises ``InputError`` (type encoded in FailedOp)
F	T	T	raises ``InputError`` (type encoded in FailedOp)
F	F (def)	T	dict of ``FailedOperation``

Return of qcengine.compute()

good_calc?	input_data ver	return_version	return_dict	output
T	1	–1 (def) or 1	F (def)	``v1.AtomicResult`` object (not avail. Py 3.14+ ^^)
T	1	2	F (def)	``v2.AtomicResult`` object
T	1	–1 (def) or 1	T	dict of ``v1.AtomicResult`` @@
T	1	2	T	dict of ``v2.AtomicResult``
T	2	–1 (def) or 2	F (def)	``v2.AtomicResult`` object
T	2	1	F (def)	``v1.AtomicResult`` object (not avail. Py 3.14+ ^^)
T	2	–1 (def) or 2	T	dict of ``v2.AtomicResult``
T	2	1	T	dict of ``v1.AtomicResult`` @@
F	1	–1 (def) or 1	F (def)	``v1.FailedOperation`` object (not avail. Py 3.14+ ^^) **
F	1	2	F (def)	``v2.FailedOperation`` object
F	1	–1 (def) or 1	T	dict of ``v1.FailedOperation`` @@ **
F	1	2	T	dict of ``v2.FailedOperation``
F	2	–1 (def) or 2	F (def)	``v2.FailedOperation`` object **
F	2	1	F (def)	``v1.FailedOperation`` object (not avail. Py 3.14+ ^^)
F	2	–1 (def) or 2	T	dict of ``v2.FailedOperation`` **
F	2	1	T	dict of ``v1.FailedOperation`` @@
** F		** –1		for errors *before* input ver detected, returns v2 if Py 3.14+ else v1
		^^ 1	^^ F	for Py 3.14+, returns v2.FailedOp for minimal API surprise
		@@ 1	@@ T	for Py 3.14+, returns dict of v1 (like all other Py) with brittle *envvar: `QCNG_USE_V1V2_SHIM=1`, otherwise dict of v2.FailedOp

_v1v2 *means a class in Pydantic API v2 with the QCSchema v1 layout exists. Its use is a last resort sometimes necessary for API continuity with Py314. Available as a target of* `convert_v`. ***Never** return its models, only its dictionaries. No assurance of continued availability. See environment variables* `QCEL_V1V2_SHIM_CODE` *and* `QCNG_USE_V1V2_SHIM`.