**simulation module**

The simulation module contains the classes necessary to perform a simulation of a Bunch traversing a Lattice.

**Classes**

*class* **Aperture\_operation**

*Public Functions*

**Aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

template < typename T >

void **apply\_impl**(T & t, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < typename T >

void **dump\_particles**(T & t, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

double **get\_x\_offset**()

double **get\_y\_offset**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

void **deposit\_charge**(double charge)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Aperture\_operation**()

*Public Static Attributes*

const char **charge\_attribute**[]

*class* **Aperture\_operation\_extractor**

*Public Functions*

**Aperture\_operation\_extractor**()

[*Aperture\_operation\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation_8h_1a9c1470f76d12197b99fb947e3acc35dc) **extract**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice)

**~Aperture\_operation\_extractor**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Aperture\_operation\_extractor\_map**

*Public Functions*

**Aperture\_operation\_extractor\_map**()

void **set\_extractor**(std::string const & name, [*Aperture\_operation\_extractor\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation__extractor_8h_1a16e909e978a33466e3c907d8ce50b5f7) operation\_extractor)

[*Aperture\_operation\_extractor\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation__extractor_8h_1a16e909e978a33466e3c907d8ce50b5f7) **get\_extractor**(std::string const & name)

std::list< std::string > **get\_extractor\_names**()

**~Aperture\_operation\_extractor\_map**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Bunch\_simulator**

*Public Functions*

**Bunch\_simulator**([*Bunch\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0bunch_8h_1a692c3b3b613b19e0fd4c4d2603f2d554) bunch\_sptr)

**Bunch\_simulator**([*Bunch\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0bunch_8h_1a692c3b3b613b19e0fd4c4d2603f2d554) bunch\_sptr, [*Diagnostics\_actions\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0diagnostics__actions_8h_1a66fb03954b109df7fe165640ed59d173) diagnostics\_actions\_sptr)

**Bunch\_simulator**()

[*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & **get\_bunch**()

[*Bunch\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0bunch_8h_1a692c3b3b613b19e0fd4c4d2603f2d554) **get\_bunch\_sptr**()

[*Diagnostics\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_diagnostics__actions) & **get\_diagnostics\_actions**()

[*Diagnostics\_actions\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0diagnostics__actions_8h_1a66fb03954b109df7fe165640ed59d173) **get\_diagnostics\_actions\_sptr**()

void **add\_per\_turn**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int turn\_period = 1)

void **add\_per\_turn**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, std::list< int > const & turn\_numbers)

void **add\_per\_step**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int step\_period = 1)

void **add\_per\_step**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, std::list< int > const & step\_numbers, int turn\_period = 1)

void **add\_per\_forced\_diagnostics\_step**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int turn\_period = 1)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Bunch\_simulator**()

*class* **Bunch\_train\_simulator**

*Public Functions*

**Bunch\_train\_simulator**(Bunch\_train\_sptr bunch\_train\_sptr)

**Bunch\_train\_simulator**()

[*Bunch\_train*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch__train) & **get\_bunch\_train**()

Bunch\_train\_sptr **get\_bunch\_train\_sptr**()

Diagnostics\_actionss & **get\_diagnostics\_actionss**()

void **add\_per\_turn**(int which, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int period = 1)

void **add\_per\_turn**(int which, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, std::list< int > const & turn\_numbers)

void **add\_per\_step**(int which, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int period = 1)

void **add\_per\_step**(int which, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, std::list< int > const & step\_numbers, int turn\_period = 1)

void **add\_per\_forced\_diagnostics\_step**(int which, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int turn\_period = 1)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Bunch\_train\_simulator**()

*class* **Chef\_map\_operation\_extractor**

*Public Functions*

**Chef\_map\_operation\_extractor**([*Chef\_lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice_8h_1a22ad176949b22cf1f6e2a236134553df) chef\_lattice\_sptr, int map\_order)

**Chef\_map\_operation\_extractor**()

[*Independent\_operations*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a17e1efe32ee93cfdc83c48340187f487) **extract**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle, [*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & slices)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Chef\_mixed\_operation\_extractor**

*Public Functions*

**Chef\_mixed\_operation\_extractor**([*Chef\_lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice_8h_1a22ad176949b22cf1f6e2a236134553df) chef\_lattice\_sptr, int map\_order)

**Chef\_mixed\_operation\_extractor**()

[*Independent\_operations*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a17e1efe32ee93cfdc83c48340187f487) **extract**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle, [*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & slices)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Chef\_propagate\_operation**

*Public Functions*

**Chef\_propagate\_operation**([*Chef\_lattice\_section*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_chef__lattice__section) const & chef\_lattice\_section)

**Chef\_propagate\_operation**()

Default constructor for serialization use only.

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Chef\_propagate\_operation**()

*class* **Chef\_propagate\_operation\_extractor**

*Public Functions*

**Chef\_propagate\_operation\_extractor**([*Chef\_lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice_8h_1a22ad176949b22cf1f6e2a236134553df) chef\_lattice\_sptr, int map\_order)

**Chef\_propagate\_operation\_extractor**()

[*Independent\_operations*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a17e1efe32ee93cfdc83c48340187f487) **extract**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle, [*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & slices)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Chef\_propagator**

*Public Functions*

**Chef\_propagator**([*Chef\_lattice\_section\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice__section__fwd_8h_1a764c8616d58e5008caeac5d267927ff4) chef\_lattice\_section\_sptr)

**Chef\_propagator**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Circular\_aperture\_operation**

A circular aperture with radius in meters determined by the [*Lattice\_element*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element) attribute “circular\_aperture\_radius”.

If the radius is not defined, the default value of 1000.0 m will be used.

*Public Functions*

**Circular\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Circular\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator==**([*Circular\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_circular__aperture__operation) const & circular\_aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Circular\_aperture\_operation**()

*Public Static Attributes*

const double **default\_radius**

const char **aperture\_type**[]

const char **attribute\_name**[]

*class* **Collective\_operator**

*Public Functions*

**Collective\_operator**(std::string const & name)

**Collective\_operator**()

Default constructor for serialization use only.

[*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator) \* **clone**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, Diagnosticss const & per\_operation\_diagnosticss, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Collective\_operator**()

*Public Static Attributes*

const char **type\_name**[]

*class* **Dense\_mapping**

*Public Functions*

**Dense\_mapping**([*Fast\_mapping*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping) const & fast\_mapping)

[*MArray1d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a9326a326b0bb8363bac5c6dc1feba89b) **get\_constant\_term**()

[*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) **get\_linear\_term**()

**~Dense\_mapping**()

*class* **Diagnostics\_actions**

*Public Type*

typedef std::list< [*Periodic*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_diagnostics__actions_1_1_periodic) > **Periodics**

typedef std::list< int > **Numbers**

typedef std::list< [*Listed*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_diagnostics__actions_1_1_listed) > **Listeds**

typedef std::list< [*Periodic\_listed*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_diagnostics__actions_1_1_periodic__listed) > **Periodic\_listeds**

*Public Functions*

**Diagnostics\_actions**()

void **set\_bunch\_sptr**([*Bunch\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0bunch_8h_1a692c3b3b613b19e0fd4c4d2603f2d554) bunch\_sptr)

bool **have\_bunch\_sptr**()

[*Bunch\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0bunch_8h_1a692c3b3b613b19e0fd4c4d2603f2d554) **get\_bunch\_sptr**()

void **add\_per\_turn**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int turn\_period = 1)

void **add\_per\_turn**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, std::list< int > const & turn\_numbers)

void **add\_per\_step**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int step\_period = 1)

void **add\_per\_step**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, std::list< int > const & step\_numbers, int turn\_period = 1)

void **add\_per\_forced\_diagnostics\_step**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr, int turn\_period = 1)

void **add\_per\_operator**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr)

void **add\_per\_operation**([*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr)

void **first\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch)

void **turn\_end\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int turn\_num)

void **step\_end\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int turn\_num, int step\_num)

Diagnosticss & **get\_per\_operator\_diagnosticss**()

Diagnosticss & **get\_per\_operation\_diagnosticss**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Diagnostics\_actions**()

*class* **Listed**

*Public Functions*

**Listed**(Numbers const & numbers, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr)

**Listed**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*Public Members*

Numbers **numbers**

[*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) **diagnostics\_sptr**

*class* **Periodic**

*Public Functions*

**Periodic**(int period, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr)

**Periodic**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*Public Members*

int **period**

[*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) **diagnostics\_sptr**

*class* **Periodic\_listed**

*Public Functions*

**Periodic\_listed**(int turn\_period, Numbers const & step\_numbers, [*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) diagnostics\_sptr)

**Periodic\_listed**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*Public Members*

int **turn\_period**

Numbers **step\_numbers**

[*Diagnostics\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0diagnostics_8h_1a27a65c14c4e8fc1bc909d405ff0a0bf7) **diagnostics\_sptr**

*class* **Dummy\_collective\_operator**

*Public Functions*

**Dummy\_collective\_operator**(std::string const & name)

**Dummy\_collective\_operator**()

Default constructor for serialization use only.

[*Dummy\_collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_dummy__collective__operator) \* **clone**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Dummy\_collective\_operator**()

*class* **Elliptical\_aperture\_operation**

An elliptical aperture with horizontal and vertical radii in meters determined by the Lattice\_element\_attributes “elliptical\_aperture\_horizontal\_radius” and “elliptical\_aperture\_vertical\_radius”, respectively.

Both radii must be specified. Failing to do so will cause an exception.

*Public Functions*

**Elliptical\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Elliptical\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator==**([*Elliptical\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_elliptical__aperture__operation) const & Elliptical\_aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Elliptical\_aperture\_operation**()

*Public Static Attributes*

const char **aperture\_type**[]

const char **attribute\_name**[]

*class* **Fast\_mapping**

*Public Functions*

**Fast\_mapping**(int order)

**Fast\_mapping**(std::string const & filename)

**Fast\_mapping**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle, Mapping const & chef\_mapping, double mapping\_length)

**Fast\_mapping**()

Default constructor for serialization use only.

void **set\_length**(double length)

double **get\_length**()

int **get\_order**()

void **add\_term**(int index, [*Fast\_mapping\_term*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping__term) const & term)

std::vector< std::vector< [*Fast\_mapping\_terms*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0fast__mapping_8h_1ae9ebeee6b56233f3dbf9a3141e2fcbd8) > > const & **get\_terms**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch)

std::string **as\_string**()

void **write\_to\_file**(std::string const & filename)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Fast\_mapping\_operation**

*Public Functions*

**Fast\_mapping\_operation**([*Fast\_mapping*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping) const & mapping)

**Fast\_mapping\_operation**()

Default constructor for serialization use only.

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

[*Fast\_mapping*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping) const & **get\_fast\_mapping**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Fast\_mapping\_operation**()

*class* **Fast\_mapping\_term**

*Public Functions*

**Fast\_mapping\_term**(int order)

**Fast\_mapping\_term**(std::ifstream & stream)

**Fast\_mapping\_term**()

Default constructor for serialization use only.

**Fast\_mapping\_term**([*Fast\_mapping\_term*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping__term) const & fast\_mapping\_term)

int **order**()

double & **coeff**()

double const & **coeff**()

int & **index**(int which)

int const & **index**(int which)

void **write\_to\_stream**(std::ostream & stream)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Fast\_mapping\_term**()

*class* **Finite\_aperture\_operation**

An aperture to remove all particles with infinite and/or NaN coordinates.

*Public Functions*

**Finite\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Finite\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Finite\_aperture\_operation**()

*Public Static Attributes*

const char **aperture\_type**[]

const char **attribute\_name**[]

template < typename T >

*class* **Generic\_aperture\_extractor**

*Public Functions*

**Generic\_aperture\_extractor**()

[*Aperture\_operation\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation_8h_1a9c1470f76d12197b99fb947e3acc35dc) **extract**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice)

**~Generic\_aperture\_extractor**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Independent\_operation**

*Public Functions*

**Independent\_operation**(std::string const & type)

**Independent\_operation**()

Default constructor for serialization use only.

std::string const & **get\_type**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Independent\_operation**()

*class* **Independent\_operator**

*Public Functions*

**Independent\_operator**(std::string const & name, [*Operation\_extractor\_map\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operation__extractor_8h_1ad31e58d6a29a6115707617226347d8ad) operation\_extractor\_map\_sptr, [*Aperture\_operation\_extractor\_map\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation__extractor_8h_1a5a7443274fe19045547cecbcd1769244) aperture\_operation\_extractor\_map\_sptr)

**Independent\_operator**()

Default constructor for serialization use only.

void **append\_slice**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

[*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & **get\_slices**()

void **update\_operations**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle)

bool **need\_update**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

[*Independent\_operations*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a17e1efe32ee93cfdc83c48340187f487) const & **get\_operations**()

[*Independent\_operations*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a17e1efe32ee93cfdc83c48340187f487) & **get\_operations**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, Diagnosticss const & per\_operation\_diagnosticss, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger, [*Multi\_diagnostics*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_multi__diagnostics) & diagnostics)

void **print**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Independent\_operator**()

*Public Static Attributes*

const char **type\_name**[]

*class* **Independent\_stepper**

The [*Independent\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper) class generates evenly-spaced [*Independent\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__operator) steps through a [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice).

No collective effects are included.

*Public Functions*

**Independent\_stepper**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, int num\_steps)

Construct an [*Independent\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper).

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

* num\_steps -

the number of steps to take in the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

**Independent\_stepper**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, int num\_steps)

Deprecated.

Construct an [*Independent\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper)

**Parameters**

* lattice\_simulator -

the [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) for the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* num\_steps -

the number of steps to take in the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

**Independent\_stepper**()

Default constructor for serialization use only.

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Independent\_stepper**()

*class* **Independent\_stepper\_elements**

The [*Independent\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper__elements) class generates a constant number of [*Independent\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__operator) steps per thick element.

Thin elements are assigned a single step each. No collective effects are included.

*Public Functions*

**Independent\_stepper\_elements**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, int steps\_per\_element)

Construct an [*Independent\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper).

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

* steps\_per\_element -

the number of steps per thick element

**Independent\_stepper\_elements**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, int steps\_per\_element)

Deprecated.

Construct an [*Independent\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper)

**Parameters**

* lattice\_simulator -

the [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) for the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* steps\_per\_element -

the number of steps per thick element

**Independent\_stepper\_elements**()

Default constructor for serialization use only.

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Independent\_stepper\_elements**()

*class* **Lambertson\_aperture\_operation**

A Lambertson aperture with radius in meters determined by the [*Lattice\_element*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element) attribute “lambertson\_aperture\_radius”.

If the radius is not defined, the default value of 1000.0 m will be used.

*Public Functions*

**Lambertson\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Lambertson\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator==**([*Lambertson\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lambertson__aperture__operation) const & lambertson\_aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosit, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Lambertson\_aperture\_operation**()

*Public Static Attributes*

const char **aperture\_type**[]

const char **attribute\_name**[]

*class* **Lattice\_simulator**

*Public Functions*

**Lattice\_simulator**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order)

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

**Lattice\_simulator**()

**Lattice\_simulator**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator)

void **set\_slices**([*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & slices)

[*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & **get\_slices**()

int **get\_map\_order**()

void **set\_bucket\_length**()

double **get\_bucket\_length**()

bucket length is in z\_lab frame

int **get\_number\_buckets**()

[*Operation\_extractor\_map\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operation__extractor_8h_1ad31e58d6a29a6115707617226347d8ad) **get\_operation\_extractor\_map\_sptr**()

[*Aperture\_operation\_extractor\_map\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation__extractor_8h_1a5a7443274fe19045547cecbcd1769244) **get\_aperture\_operation\_extractor\_map\_sptr**()

[*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice) & **get\_lattice**()

[*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) **get\_lattice\_sptr**()

[*Chef\_lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_chef__lattice) & **get\_chef\_lattice**()

[*Chef\_lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice_8h_1a22ad176949b22cf1f6e2a236134553df) **get\_chef\_lattice\_sptr**()

void **update**()

void **calculate\_element\_lattice\_functions**()

void **calculate\_slice\_lattice\_functions**()

void **calculate\_element\_et\_lattice\_functions**()

void **calculate\_slice\_et\_lattice\_functions**()

void **calculate\_element\_lb\_lattice\_functions**()

void **calculate\_slice\_lb\_lattice\_functions**()

void **calculate\_element\_dispersion\_functions**()

void **calculate\_slice\_dispersion\_functions**()

Lattice\_functions const & **get\_lattice\_functions**([*Lattice\_element*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element) & lattice\_element)

Lattice\_functions const & **get\_lattice\_functions**([*Lattice\_element\_slice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element__slice) & lattice\_element\_slice)

ET\_lattice\_functions const & **get\_et\_lattice\_functions**([*Lattice\_element*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element) & lattice\_element)

ET\_lattice\_functions const & **get\_et\_lattice\_functions**([*Lattice\_element\_slice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element__slice) & lattice\_element\_slice)

LB\_lattice\_functions const & **get\_lb\_lattice\_functions**([*Lattice\_element*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element) & lattice\_element)

LB\_lattice\_functions const & **get\_lb\_lattice\_functions**([*Lattice\_element\_slice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element__slice) & lattice\_element\_slice)

Dispersion\_functions const & **get\_dispersion\_functions**([*Lattice\_element*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element) & lattice\_element)

Dispersion\_functions const & **get\_dispersion\_functions**([*Lattice\_element\_slice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice__element__slice) & lattice\_element\_slice)

void **print\_cs\_lattice\_functions**()

void **print\_et\_lattice\_functions**()

void **print\_lb\_lattice\_functions**()

void **print\_dispersion\_closedOrbit**()

void **print\_lattice\_functions**()

std::pair< double, double > **get\_both\_tunes**(bool use\_eigen\_tune = false)

double **get\_horizontal\_tune**(bool use\_eigen\_tune = false)

double **get\_vertical\_tune**(bool use\_eigen\_tune = false)

bool **is\_ring**()

[*Normal\_form\_sage\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0lattice__simulator_8h_1a04e7fcd3e6a75493dc8db4279dbc29e7) **get\_normal\_form\_sptr**()

[*Const\_MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a0c7d8b792425b7d17a90ca4d88b439d9) **get\_linear\_one\_turn\_map**(bool sliced = true)

void **convert\_human\_to\_normal**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) coords)

void **convert\_normal\_to\_human**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) coords)

bool **check\_linear\_normal\_form**()

std::vector< double > **get\_stationary\_actions**(const double stdx, const double stdy, const double std\_cdt)

void **adjust\_tunes**(double horizontal\_tune, double vertical\_tune, [*Lattice\_elements*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element_8h_1ae81b6b4f394d59398056cc93bdb2c56c) const & horizontal\_correctors, [*Lattice\_elements*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element_8h_1ae81b6b4f394d59398056cc93bdb2c56c) const & vertical\_correctors, double tolerance = 1.0e-5, int verbosity = 0)

double **get\_slip\_factor**(double dpp = 1.e-4)

double **get\_momentum\_compaction**(double dpp = 1.e-4)

double **get\_horizontal\_chromaticity**(double dpp = 1.e-4)

double **get\_vertical\_chromaticity**(double dpp = 1.e-4)

void **adjust\_chromaticities**(double horizontal\_chromaticity, double vertical\_chromaticity, [*Lattice\_elements*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element_8h_1ae81b6b4f394d59398056cc93bdb2c56c) const & horizontal\_correctors, [*Lattice\_elements*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element_8h_1ae81b6b4f394d59398056cc93bdb2c56c) const & vertical\_correctors, double tolerance = 1.0e-4, int max\_steps = 6)

template < class Archive >

void **save**(Archive & ar, const unsigned int version)

template < class Archive >

void **load**(Archive & ar, const unsigned int version)

**BOOST\_SERIALIZATION\_SPLIT\_MEMBER**()

**~Lattice\_simulator**()

*class* **Operation\_extractor**

*Public Functions*

**Operation\_extractor**([*Chef\_lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice_8h_1a22ad176949b22cf1f6e2a236134553df) chef\_lattice\_sptr, int map\_order)

**Operation\_extractor**()

[*Chef\_lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0chef__lattice_8h_1a22ad176949b22cf1f6e2a236134553df) & **get\_chef\_lattice\_sptr**()

int **get\_map\_order**()

[*Independent\_operations*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a17e1efe32ee93cfdc83c48340187f487) **extract**([*Reference\_particle*](http://compacc.fnal.gov/~amundson/html/foundation.html#project0class_reference__particle) const & reference\_particle, [*Lattice\_element\_slices*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1a7f51f027f9907f3a4e08f37304f858c4) const & slices)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Operation\_extractor**()

*class* **Operation\_extractor\_map**

*Public Functions*

**Operation\_extractor\_map**()

void **set\_extractor**(std::string const & name, [*Operation\_extractor\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operation__extractor_8h_1a25fec47d7168068615642003eb1ac03c) operation\_extractor)

[*Operation\_extractor\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operation__extractor_8h_1a25fec47d7168068615642003eb1ac03c) & **get\_extractor**(std::string const & name)

std::list< std::string > **get\_extractor\_names**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Operation\_extractor\_map**()

*class* **Operator**

*Public Functions*

**Operator**(std::string const & name, std::string const & type)

**Operator**()

Default constructor for serialization use only.

std::string const & **get\_name**()

std::string const & **get\_type**()

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, Diagnosticss const & per\_operation\_diagnosticss, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

void **apply**([*Bunch\_train*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch__train) & bunch\_train, double time\_step, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, int verbosity, Train\_diagnosticss const & per\_operation\_train\_diagnosticss, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

void **print**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Operator**()

*class* **Polygon\_aperture\_operation**

A polygon aperture with vertices determined by the Lattice\_element\_attributes “pax1”, “pay1”, “pax2”, “pay2”, and so on.

And it also requires “the\_number\_of\_vertices”, which determines the number of vertices and must be greter than and equal to 3. Must have at least 3 vertcies. Failing to do so will cause an exception.

*Public Functions*

**Polygon\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Polygon\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator==**([*Polygon\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_polygon__aperture__operation) const & polygon\_aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Polygon\_aperture\_operation**()

*Public Static Attributes*

const char **aperture\_type**[]

const char **attribute\_name**[]

*class* **Propagate\_actions**

*Public Functions*

**Propagate\_actions**()

void **first\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch)

void **first\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Bunch\_train*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch__train) & bunch\_train)

void **turn\_end\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int turn\_num)

void **turn\_end\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Bunch\_train*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch__train) & bunch\_train, int turn\_num)

void **step\_end\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, [*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int turn\_num, int step\_num)

void **step\_end\_action**([*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) & stepper, [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) & step, [*Bunch\_train*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch__train) & bunch\_train, int turn\_num, int step\_num)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Propagate\_actions**()

*class* **Propagator**

*Public Functions*

**Propagator**([*Stepper\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0stepper_8h_1a0c6ff01fe7f23d407212dac215ba0227) stepper\_sptr)

**Propagator**()

[*Stepper\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0stepper_8h_1a0c6ff01fe7f23d407212dac215ba0227) **get\_stepper\_sptr**()

void **set\_checkpoint\_period**(int period)

int **get\_checkpoint\_period**()

void **set\_checkpoint\_dir**(std::string const & directory\_name)

std::string const & **get\_checkpoint\_dir**()

void **set\_checkpoint\_with\_xml**(bool with\_xml)

bool **get\_checkpoint\_with\_xml**()

void **set\_final\_checkpoint**(bool final\_checkpoint)

bool **get\_final\_checkpoint**()

void **set\_concurrent\_io**(int max)

int **get\_concurrent\_io**()

void **propagate**([*State*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_propagator_1_1_state) & state)

[*State*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_propagator_1_1_state) **get\_resume\_state**(std::string const & checkpoint\_dir)

jfa note: the lifetime of the pointers in state must be managed manually

void **resume**(std::string const & checkpoint\_dir, bool new\_max\_turns, int max\_turns, bool new\_verbosity, int verbosity)

void **propagate**([*Bunch\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__simulator) & bunch\_simulator, int num\_turns, int max\_turns = 0, int verbosity = 1)

void **propagate**([*Bunch\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__simulator) & bunch\_simulator, [*Propagate\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagate__actions) & general\_actions, int num\_turns, int max\_turns = 0, int verbosity = 1)

void **propagate**([*Bunch\_train\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__train__simulator) & bunch\_train\_simulator, int num\_turns, int max\_turns = 0, int verbosity = 1)

void **propagate**([*Bunch\_train\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__train__simulator) & bunch\_train\_simulator, [*Propagate\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagate__actions) & general\_actions, int num\_turns, int max\_turns = 0, int verbosity = 1)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Propagator**()

*Public Static Attributes*

const std::string **default\_checkpoint\_dir**

const std::string **description\_file\_name**

const std::string **propagator\_archive\_name**

const std::string **propagator\_xml\_archive\_name**

const std::string **state\_archive\_name**

const std::string **state\_xml\_archive\_name**

const std::string **log\_file\_name**

const std::string **stop\_file\_name**

const std::string **alt\_stop\_file\_name**

const int **default\_checkpoint\_period**

const int **default\_concurrent\_io**

*class* **State**

*Public Functions*

**State**([*Bunch\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__simulator) \* bunch\_simulator\_ptr, [*Propagate\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagate__actions) \* propagate\_actions\_ptr, int num\_turns, int first\_turn, int max\_turns, int verbosity)

**State**([*Bunch\_train\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__train__simulator) \* bunch\_train\_simulator\_ptr, [*Propagate\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagate__actions) \* propagate\_actions\_ptr, int num\_turns, int first\_turn, int max\_turns, int verbosity)

**State**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*Public Members*

[*Bunch\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__simulator) \* **bunch\_simulator\_ptr**

[*Bunch\_train\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__train__simulator) \* **bunch\_train\_simulator\_ptr**

[*Propagate\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagate__actions) \* **propagate\_actions\_ptr**

int **num\_turns**

int **first\_turn**

int **max\_turns**

int **verbosity**

*class* **Rectangular\_aperture\_operation**

A rectangular aperture with horizontal and vertical dimensions in meters determined by the Lattice\_element\_attributes “rectangular\_aperture\_width” and “rectangular\_aperture\_height”, respectively.

Both dimensions must be specified. Failing to do so will cause an exception.

*Public Functions*

**Rectangular\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Rectangular\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator==**([*Rectangular\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_rectangular__aperture__operation) const & rectangular\_aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Rectangular\_aperture\_operation**()

*Public Static Attributes*

const char **aperture\_type**[]

const char **attribute\_name**[]

*class* **Resume**

*Public Functions*

**Resume**(std::string const & checkpoint\_dir = [*Propagator::default\_checkpoint\_dir*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagator_1a1200d65737ee525f363846965efa458b))

void **set\_checkpoint\_period**(int period)

int **get\_checkpoint\_period**()

void **set\_new\_checkpoint\_dir**(std::string const & directory\_name)

std::string const & **get\_new\_checkpoint\_dir**()

[*Content*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_resume_1_1_content) **get\_content**()

void **propagate**(bool new\_max\_turns, int max\_turns, bool new\_verbosity, int verbosity)

**~Resume**()

*class* **Content**

*Public Functions*

**Content**([*Bunch\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_bunch__simulator) \* bunch\_simulator\_ptr, [*Stepper\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0stepper_8h_1a0c6ff01fe7f23d407212dac215ba0227) stepper\_sptr)

*Public Members*

[*Bunch\_sptr*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0bunch_8h_1a692c3b3b613b19e0fd4c4d2603f2d554) **bunch\_sptr**

[*Stepper\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0stepper_8h_1a0c6ff01fe7f23d407212dac215ba0227) **stepper\_sptr**

[*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) **lattice\_sptr**

*class* **Split\_operator\_stepper**

The [*Split\_operator\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper) class generates evenly-spaced split-operator steps through a [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice).

One or more collective effects are included per step.

*Public Functions*

**Split\_operator\_stepper**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, [*Collective\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1aa488345d765c65fca30fb25e273ba122) collective\_operator, int num\_steps)

Construct a [*Split\_operator\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper) with a single [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator).

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

* collective\_operator -

the [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator) to apply in each step

* num\_steps -

the number of steps to take in the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

**Split\_operator\_stepper**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, [*Collective\_operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1afe4aab55fcbbc832ac759c026312654c) const & collective\_operators, int num\_steps)

Construct a [*Split\_operator\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper) with multiple Collective\_operators.

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

* collective\_operators -

the set of Collective\_operators to apply in each step

* num\_steps -

the number of steps to take in the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

**Split\_operator\_stepper**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, [*Collective\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1aa488345d765c65fca30fb25e273ba122) collective\_operator, int num\_steps)

Deprecated.

Construct a [*Split\_operator\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper) with a single [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator)

**Parameters**

* lattice\_simulator -

the [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) for the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* collective\_operator -

the [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator) to apply in each step

* num\_steps -

the number of steps to take in the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

**Split\_operator\_stepper**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, [*Collective\_operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1afe4aab55fcbbc832ac759c026312654c) const & collective\_operators, int num\_steps)

Deprecated.

Construct a [*Split\_operator\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper) with multiple Collective\_operators

**Parameters**

* lattice\_simulator -

the [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) for the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* collective\_operators -

the set of Collective\_operators to apply in each step

* num\_steps -

the number of steps to take in the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

**Split\_operator\_stepper**()

Default constructor for serialization use only.

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Split\_operator\_stepper**()

*class* **Split\_operator\_stepper\_choice**

Generate steps according with a list.

*Public Functions*

**Split\_operator\_stepper\_choice**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, List\_choice\_map const & list\_choice\_map, bool split\_else = true)

**Split\_operator\_stepper\_choice**(int num\_steps\_else, [*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, List\_choice\_map const & list\_choice\_map, bool split\_else = true)

**Split\_operator\_stepper\_choice**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, List\_choice\_map const & list\_choice\_map, bool split\_else = true)

Deprecated.

**Split\_operator\_stepper\_choice**(int num\_steps\_else, [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, List\_choice\_map const & list\_choice\_map, bool split\_else = true)

Deprecated.

**Split\_operator\_stepper\_choice**()

Default constructor for serialization use only.

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Split\_operator\_stepper\_choice**()

*class* **Split\_operator\_stepper\_elements**

The [*Split\_operator\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__elements) class generates a constant number of split-operator steps per thick element.

Thin elements are assigned a single step each. One or more collective effects are included per step.

*Public Functions*

**Split\_operator\_stepper\_elements**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, [*Collective\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1aa488345d765c65fca30fb25e273ba122) collective\_operator, int steps\_per\_element)

Construct a [*Split\_operator\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__elements) with a single [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator).

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

* collective\_operator -

the [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator) to apply in each step

* steps\_per\_element -

the number of steps per thick element

**Split\_operator\_stepper\_elements**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order, [*Collective\_operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1afe4aab55fcbbc832ac759c026312654c) const & collective\_operators, int steps\_per\_element)

Construct a [*Split\_operator\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__elements) with multiple Collective\_operators.

**Parameters**

* lattice\_sptr -

the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* map\_order -

order for Chef\_map operations

* collective\_operators -

the set of Collective\_operators to apply in each step

* steps\_per\_element -

the number of steps per thick element

**Split\_operator\_stepper\_elements**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, [*Collective\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1aa488345d765c65fca30fb25e273ba122) collective\_operator, int steps\_per\_element)

Deprecated.

Construct a [*Split\_operator\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__elements) with a single [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator)

**Parameters**

* lattice\_simulator -

the [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) for the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* collective\_operator -

the [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator) to apply in each step

* steps\_per\_element -

the number of steps per thick element

**Split\_operator\_stepper\_elements**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator, [*Collective\_operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1afe4aab55fcbbc832ac759c026312654c) const & collective\_operators, int steps\_per\_element)

Deprecated.

Construct a [*Split\_operator\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__elements) with multiple Collective\_operators

**Parameters**

* lattice\_simulator -

the [*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) for the [*Lattice*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0class_lattice)

* collective\_operators -

the set of Collective\_operators to apply in each step

* steps\_per\_element -

the number of steps per thick element

**Split\_operator\_stepper\_elements**()

Default constructor for serialization use only.

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Split\_operator\_stepper\_elements**()

*class* **Step**

*Public Functions*

**Step**(double length)

**Step**()

void **append**([*Operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a2e1fa2dff989733b6ee793c14e48f92b) operator\_sptr, double time\_fraction)

void **append**([*Operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a539dc9ab3300f1e2e5edd2d3cd13aaef) const & operators, double time\_fraction)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, Diagnosticss const & per\_operator\_diagnostics, Diagnosticss const & per\_operation\_diagnostics, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

void **apply**([*Bunch\_train*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch__train) & bunch\_train, int verbosity, Train\_diagnosticss const & per\_operator\_train\_diagnosticss, Train\_diagnosticss const & per\_operation\_train\_diagnosticss, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

[*Operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a539dc9ab3300f1e2e5edd2d3cd13aaef) const & **get\_operators**()

[*Operators*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a539dc9ab3300f1e2e5edd2d3cd13aaef) & **get\_operators**()

std::list< double > const & **get\_time\_fractions**()

double **get\_length**()

void **print**(int index)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

*class* **Stepper**

*Public Functions*

**Stepper**([*Lattice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice_8h_1aed3d29e01edf5749d82e95bcd935bd9b) lattice\_sptr, int map\_order)

**Stepper**([*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) const & lattice\_simulator)

Deprecated.

**Stepper**()

Default constructor for serialization use only.

[*Lattice\_simulator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lattice__simulator) & **get\_lattice\_simulator**()

[*Steps*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0step_8h_1a156fb34ec56353ce0f3b43b6c75ecf0d) & **get\_steps**()

void **force\_update\_operations\_no\_collective**()

void **print**()

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Stepper**()

*Public Static Attributes*

const std::string **force\_diagnostics\_attribute**

const double **fixed\_step\_tolerance**

*class* **Wire\_elliptical\_aperture\_operation**

An wire\_elliptical aperture with horizontal and vertical radii in meters determined by the Lattice\_element\_attributes “wire\_elliptical\_aperture\_horizontal\_radius” and “wire\_elliptical\_aperture\_vertical\_radius”, respectively.

Both radii must be specified. Also needs Lattice\_element\_attributes “wire\_elliptical\_aperture\_wire\_x”, “wire\_elliptical\_aperture\_wire\_width”, and “wire\_elliptical\_aperture\_gap.” Failing to do so will cause an exception.

*Public Functions*

**Wire\_elliptical\_aperture\_operation**([*Lattice\_element\_slice\_sptr*](http://compacc.fnal.gov/~amundson/html/lattice.html#project0lattice__element__slice_8h_1ab0bb2db459044dd329c1245604313030) slice\_sptr)

**Wire\_elliptical\_aperture\_operation**()

const char \* **get\_aperture\_type**()

bool **operator==**([*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) const & aperture\_operation)

bool **operator==**([*Wire\_elliptical\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_wire__elliptical__aperture__operation) const & Wire\_elliptical\_aperture\_operation)

bool **operator()**([*MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae229cef4126da69b5a71fef37eeac3c5) & particles, int part)

void **apply**([*Bunch*](http://compacc.fnal.gov/~amundson/html/bunch.html#project0class_bunch) & bunch, int verbosity, [*Logger*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_logger) & logger)

template < class Archive >

void **serialize**(Archive & ar, const unsigned int version)

**~Wire\_elliptical\_aperture\_operation**()

*Public Static Attributes*

const char **aperture\_type**[]

const char **attribute\_name**[]

**Typedefs**

typedef boost::shared\_ptr< [*Aperture\_operation\_extractor\_map*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation__extractor__map) > **Aperture\_operation\_extractor\_map\_sptr**

typedef boost::shared\_ptr< [*Aperture\_operation\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation__extractor) > **Aperture\_operation\_extractor\_sptr**

typedef boost::shared\_ptr< [*Aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_aperture__operation) > **Aperture\_operation\_sptr**

typedef std::list< [*Aperture\_operation\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0aperture__operation_8h_1a9c1470f76d12197b99fb947e3acc35dc) > **Aperture\_operation\_sptrs**

typedef boost::shared\_ptr< [*Chef\_propagate\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_chef__propagate__operation) > **Chef\_propagate\_operation\_sptr**

typedef [*Generic\_aperture\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_generic__aperture__extractor) < [*Circular\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_circular__aperture__operation) > **Circular\_extractor**

typedef boost::shared\_ptr< [*Collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_collective__operator) > **Collective\_operator\_sptr**

typedef std::list< [*Collective\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1aa488345d765c65fca30fb25e273ba122) > **Collective\_operators**

typedef boost::shared\_ptr< [*Diagnostics\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_diagnostics__actions) > **Diagnostics\_actions\_sptr**

typedef boost::shared\_ptr< [*Dummy\_collective\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_dummy__collective__operator) > **Dummy\_collective\_operator\_sptr**

typedef std::list< [*Dummy\_collective\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a5c6abe9d37ad65d632c3a230424c1d72) > **Dummy\_collective\_operators**

typedef [*Generic\_aperture\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_generic__aperture__extractor) < [*Elliptical\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_elliptical__aperture__operation) > **Elliptical\_extractor**

typedef boost::shared\_ptr< [*Fast\_mapping\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping__operation) > **Fast\_mapping\_operation\_sptr**

typedef std::list< [*Fast\_mapping\_term*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_fast__mapping__term) > **Fast\_mapping\_terms**

typedef boost::shared\_ptr< [*Independent\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__operation) > **Independent\_operation\_sptr**

typedef std::list< [*Independent\_operation\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0independent__operation_8h_1a720fd24a4a345d8ec0f5889a2055512e) > **Independent\_operations**

typedef boost::shared\_ptr< [*Independent\_operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__operator) > **Independent\_operator\_sptr**

typedef std::list< [*Independent\_operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a1b24db163b0ad9835ba3af69b25b862d) > **Independent\_operators**

typedef boost::shared\_ptr< [*Independent\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper__elements) > **Independent\_stepper\_elements\_sptr**

typedef boost::shared\_ptr< [*Independent\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_independent__stepper) > **Independent\_stepper\_sptr**

typedef [*Generic\_aperture\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_generic__aperture__extractor) < [*Lambertson\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_lambertson__aperture__operation) > **Lambertson\_extractor**

typedef std::list< [*Listed*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_diagnostics__actions_1_1_listed) > **Listeds**

typedef boost::shared\_ptr< normalFormSage > **Normal\_form\_sage\_sptr**

typedef std::list< int > **Numbers**

typedef boost::shared\_ptr< [*Operation\_extractor\_map*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_operation__extractor__map) > **Operation\_extractor\_map\_sptr**

typedef boost::shared\_ptr< [*Operation\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_operation__extractor) > **Operation\_extractor\_sptr**

typedef boost::shared\_ptr< [*Operator*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_operator) > **Operator\_sptr**

typedef std::list< [*Operator\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0operator_8h_1a2e1fa2dff989733b6ee793c14e48f92b) > **Operators**

typedef std::list< [*Periodic\_listed*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_diagnostics__actions_1_1_periodic__listed) > **Periodic\_listeds**

typedef std::list< [*Periodic*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0struct_diagnostics__actions_1_1_periodic) > **Periodics**

typedef [*Generic\_aperture\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_generic__aperture__extractor) < [*Polygon\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_polygon__aperture__operation) > **Polygon\_extractor**

typedef boost::shared\_ptr< [*Propagate\_actions*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_propagate__actions) > **Propagate\_actions\_sptr**

typedef [*Generic\_aperture\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_generic__aperture__extractor) < [*Rectangular\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_rectangular__aperture__operation) > **Rectangular\_extractor**

typedef boost::shared\_ptr< [*Split\_operator\_stepper\_choice*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__choice) > **Split\_operator\_stepper\_choice\_sptr**

typedef boost::shared\_ptr< [*Split\_operator\_stepper\_elements*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper__elements) > **Split\_operator\_stepper\_elements\_sptr**

typedef boost::shared\_ptr< [*Split\_operator\_stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_split__operator__stepper) > **Split\_operator\_stepper\_sptr**

typedef boost::shared\_ptr< [*Step*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_step) > **Step\_sptr**

typedef boost::shared\_ptr< [*Stepper*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_stepper) > **Stepper\_sptr**

typedef std::list< [*Step\_sptr*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0step_8h_1afc85470140144878eef4b98547b9f887) > **Steps**

typedef [*Generic\_aperture\_extractor*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_generic__aperture__extractor) < [*Wire\_elliptical\_aperture\_operation*](http://compacc.fnal.gov/~amundson/html/simulation.html#project0class_wire__elliptical__aperture__operation) > **Wire\_elliptical\_extractor**