**utils module**

The utils module contains utilities of general use, i.e., not specific to Synergia or even accelerator physics.

**Classes**

*class* **Command\_line\_arg**

*Public Functions*

**Command\_line\_arg**(const char \* char\_ptr)

bool **is\_equal\_pair**()

std::string **get\_lhs**()

std::string **get\_rhs**()

template < typename T >

T **extract\_value**()

template < >

bool **extract\_value**()

*class* **Commxx**

[*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) is a wrapper around MPI communicator (MPI\_Comm) objects.

Python: (jfa: needs update) The equivalent functionality is provided by mpi4py Comm objects. mpi4py Comm objects may be passed from python anywhere a [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) object is expected.

*Public Functions*

**Commxx**()

Construct a [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) object using MPI\_COMM\_WORLD.

**Commxx**(bool per\_host)

Construct a [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) object, optionally creating separate communicators on each unique host for communication avoidance.

**Commxx**(*[Commxx\_sptr](http://compacc.fnal.gov/~amundson/html/utils.html" \l "project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9)* parent\_sptr, bool per\_host)

Construct a [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) object based on the parent communicator, optionally creating separate communicators on each unique host for communication avoidance.

**Commxx**(*[Commxx\_sptr](http://compacc.fnal.gov/~amundson/html/utils.html" \l "project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9)* parent\_sptr, std::vector< int > const & ranks, bool per\_host = false)

Construct a [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) object using only the specified ranks on the parent communicator.

[*Commxx\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9) **get\_parent\_sptr**()

int **get\_rank**()

Get communicator rank.

int **get\_size**()

Get communicator size.

bool **has\_this\_rank**()

Test to see if the communicator contains this rank.

MPI\_Comm **get**()

Extract the MPI\_comm object wrapped by the [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) instance.

template < class Archive >

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

template < class Archive >

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

**~Commxx**()

*class* **Commxx\_divider**

*Public Functions*

**Commxx\_divider**()

**Commxx\_divider**(int subsize, bool per\_host)

[*Commxx\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9) **get\_commxx\_sptr**(*[Commxx\_sptr](http://compacc.fnal.gov/~amundson/html/utils.html" \l "project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9)* const & parent)

template < class Archive >

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

**~Commxx\_divider**()

*class* **Distributed\_fft2d**

*Public Functions*

**Distributed\_fft2d**(std::vector< int > const & shape, [*Commxx\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9) comm\_sptr, int planner\_flags = FFTW\_ESTIMATE, std::string const & wisdom\_filename = “”)

int **get\_data\_size**()

int **get\_workspace\_size**()

[*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) & **get\_comm**()

[*Commxx\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9) **get\_comm\_sptr**()

int **get\_lower**()

int **get\_upper**()

std::vector< int > const & **get\_uppers**()

std::vector< int > const & **get\_lengths**()

std::vector< int > const & **get\_lengths\_1d**()

std::vector< int > const & **get\_shape**()

void **transform**([*MArray2dc\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a639587aeb9cc120ca9f62ff4a8978da9) & in, [*MArray2dc\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a639587aeb9cc120ca9f62ff4a8978da9) & out)

void **inv\_transform**([*MArray2dc\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a639587aeb9cc120ca9f62ff4a8978da9) & in, [*MArray2dc\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a639587aeb9cc120ca9f62ff4a8978da9) & out)

double **get\_roundtrip\_normalization**()

**~Distributed\_fft2d**()

*class* **Distributed\_fft3d**

*Public Functions*

**Distributed\_fft3d**(std::vector< int > const & shape, [*Commxx\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9) comm\_sptr, int planner\_flags = FFTW\_ESTIMATE, std::string const & wisdom\_filename = “”)

[*Commxx\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0commxx_8h_1a8e4a45ebebd319814e88c81a65f0dbd9) **get\_comm\_sptr**()

[*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) & **get\_comm**()

int **get\_lower**()

int **get\_upper**()

std::vector< int > const & **get\_uppers**()

std::vector< int > const & **get\_lengths**()

std::vector< int > const & **get\_shape**()

std::vector< int > **get\_padded\_shape\_real**()

std::vector< int > **get\_padded\_shape\_complex**()

void **transform**([*MArray3d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a9febc18851e0710136053f19d4d3d068) & in, [*MArray3dc\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a884f5a39216a91268374180ba4901a51) & out)

void **inv\_transform**([*MArray3dc\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a884f5a39216a91268374180ba4901a51) & in, [*MArray3d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a9febc18851e0710136053f19d4d3d068) & out)

double **get\_roundtrip\_normalization**()

**~Distributed\_fft3d**()

*class* **Hdf5\_chunked\_array2d\_writer**

*Public Functions*

**Hdf5\_chunked\_array2d\_writer**(H5::H5File \* file\_ptr, std::string const & name, [*Const\_MArray2d\_view*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae880252591703b779ca997953f66fcba) const & initial\_data, int chunk\_size = 0)

**Hdf5\_chunked\_array2d\_writer**(H5::H5File \* file\_ptr, std::string const & name, [*Const\_MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a0c7d8b792425b7d17a90ca4d88b439d9) const & initial\_data, int chunk\_size = 0)

void **write\_chunk**([*Const\_MArray2d\_ref*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1a0c7d8b792425b7d17a90ca4d88b439d9) const & data)

void **write\_chunk**([*Const\_MArray2d\_view*](http://compacc.fnal.gov/~amundson/html/utils.html#project0multi__array__typedefs_8h_1ae880252591703b779ca997953f66fcba) const & data)

**~Hdf5\_chunked\_array2d\_writer**()

*class* **Hdf5\_file**

*Public Type*

**Flag enum**

*Values:*

* truncate -
* read\_write -
* read\_only -

*Public Functions*

**Hdf5\_file**(std::string const & file\_name, [*Flag*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_hdf5__file_1aadfcad244963cb52b97406f4aa80e51c) flag)

**Hdf5\_file**()

void **open**([*Flag*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_hdf5__file_1aadfcad244963cb52b97406f4aa80e51c) flag)

void **close**()

void **flush**()

H5::H5File & **get\_h5file**()

template < typename T >

void **write**(T const & data, std::string const & name)

template < typename T >

T **read**(std::string const & name)

template < class Archive >

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

template < class Archive >

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

**~Hdf5\_file**()

template < typename T >

*class* **Hdf5\_serial\_writer**

*Public Functions*

**Hdf5\_serial\_writer**([*Hdf5\_file\_sptr*](http://compacc.fnal.gov/~amundson/html/utils.html#project0hdf5__file_8h_1aabf20c82608aa7edaf304a96e5eb56d9) file\_sptr, std::string const & name, bool resume = false)

**Hdf5\_serial\_writer**()

void **append**(T & data)

template < class Archive >

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

template < class Archive >

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

**~Hdf5\_serial\_writer**()

template < typename T >

*class* **Hdf5\_writer**

*Public Functions*

**Hdf5\_writer**(H5::H5File \* file\_ptr, std::string const & name)

void **write**(T const & data)

**~Hdf5\_writer**()

*class* **Logger**

**Typedefs**

typedef boost::shared\_ptr< [*Commxx*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_commxx) > **Commxx\_sptr**

typedef boost::const\_multi\_array\_ref< double, 1 > **Const\_MArray1d\_ref**

typedef boost::const\_multi\_array\_ref< std::complex< double >, 1 > **Const\_MArray1dc\_ref**

typedef boost::const\_multi\_array\_ref< int, 1 > **Const\_MArray1int\_ref**

typedef boost::const\_multi\_array\_ref< double, 2 > **Const\_MArray2d\_ref**

typedef boost::detail::multi\_array::const\_multi\_array\_view< double, 2 > **Const\_MArray2d\_view**

typedef boost::const\_multi\_array\_ref< std::complex< double >, 2 > **Const\_MArray2dc\_ref**

typedef boost::const\_multi\_array\_ref< double, 3 > **Const\_MArray3d\_ref**

typedef boost::const\_multi\_array\_ref< std::complex< double >, 3 > **Const\_MArray3dc\_ref**

typedef boost::shared\_ptr< [*Distributed\_fft2d*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_distributed__fft2d) > **Distributed\_fft2d\_sptr**

typedef boost::shared\_ptr< [*Distributed\_fft3d*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_distributed__fft3d) > **Distributed\_fft3d\_sptr**

typedef boost::shared\_ptr< [*Hdf5\_file*](http://compacc.fnal.gov/~amundson/html/utils.html#project0class_hdf5__file) > **Hdf5\_file\_sptr**

typedef boost::multi\_array< double, 1 > **MArray1d**

typedef boost::multi\_array\_ref< double, 1 > **MArray1d\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< double, 1 > **MArray1d\_view**

typedef boost::multi\_array< std::complex< double >, 1 > **MArray1dc**

typedef boost::multi\_array\_ref< std::complex< double >, 1 > **MArray1dc\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< std::complex< double >, 1 > **MArray1dc\_view**

typedef boost::multi\_array< int, 1 > **MArray1int**

typedef boost::multi\_array\_ref< int, 1 > **MArray1int\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< int, 1 > **MArray1int\_view**

typedef boost::multi\_array< double, 2 > **MArray2d**

typedef boost::multi\_array\_ref< double, 2 > **MArray2d\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< double, 2 > **MArray2d\_view**

typedef boost::multi\_array< std::complex< double >, 2 > **MArray2dc**

typedef boost::multi\_array\_ref< std::complex< double >, 2 > **MArray2dc\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< std::complex< double >, 2 > **MArray2dc\_view**

typedef boost::multi\_array< double, 3 > **MArray3d**

typedef boost::multi\_array\_ref< double, 3 > **MArray3d\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< double, 3 > **MArray3d\_view**

typedef boost::multi\_array< std::complex< double >, 3 > **MArray3dc**

typedef boost::multi\_array\_ref< std::complex< double >, 3 > **MArray3dc\_ref**

typedef boost::detail::multi\_array::multi\_array\_view< std::complex< double >, 3 > **MArray3dc\_view**

typedef Raw\_multi\_array < double, 1 > **Raw\_MArray1d**

typedef Raw\_multi\_array < double, 2 > **Raw\_MArray2d**

typedef Raw\_multi\_array < std::complex< double >, 2 > **Raw\_MArray2dc**

typedef Raw\_multi\_array < double, 3 > **Raw\_MArray3d**