SAND2008-5883P

Nasgen Reference Manual

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Generated by Doxygen 1.4.0

Tue Jul 29 11:47:51 2008

Contents

Chapter 1

Nasgen Documentation

1.1 Introduction

The Nasgen utility is used for tranlating a Nastran input file into an Exodus file and a corresponding Salinas or Presto input files.

1.2 Architectural Documentation for Nasgen

This document is intended to provide background information for the Nasgen software architecture. It is targetted to software development personnel. Such personnel should gain information on the overall architecture of the software, the interactions between different components, and the class structures. We provide information on the call trees, dependency diagrams, and internal data for the classes. The documentation is automatically generated using special tags in the source code.

This document does NOT contain information about use of the software, nor the design decisions which drive the current architecture. The software usage is found in the user's manual, and a short reference is found in section Using Nasgen.

1.3 Using this Document

Various collections of information are available in this document. We recommend careful consideration of the table of contents. Some common starting locations for various actions are listed below.

- Call Trees. These provide a means of walking down through the software. The start of the tree can be found in the file list under `nasgen.C`.
- Dependency Diagrams. Again, these are most easily found in the file list. Each entry lists the included files for that file.
- Class Descriptions. Classes are organized in the *Data Structure* section. Each class in the code contains collaboration diagrams, private and public member functions, and detailed documentation for the functions associated with the class.
- Indices are provided for each function in the code.

1.3.1 What is Missing

We do not include the following:

- information about system header information.
- information from third party libraries (TPL), such as numerical libraries.
- a few, common include files are skipped because they significantly clutter the dependency graphs.

1.4 Using Nasgen

nasgen bdf_file *exodus_file* <options>

OPTIONS:

-i4 four byte reals in exodus file (default=8)

-u write UnUsed cards to log file

-S write a salinas input (∗.inp) file

-P write a presto (preliminary capability) 'support' input (∗.i) file

-e echo all BDF cards after parsing

Chapter 2

Nasgen Namespace Index

2.1 Nasgen Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Nasgen Class Index

3.1 Nasgen Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Nasgen File Index

4.1 Nasgen File List

Here is a list of all files with brief descriptions:

Chapter 5

Nasgen Namespace Documentation

5.1 std Namespace Reference

Chapter 6

Nasgen Class Documentation

6.1 BDF_Card Class Reference

in memory representation of nastran BDF card and continuations.

`#include <BDF_Card.H>`

Public Member Functions

- BDF_Card ()
- BDF_Card (FILE *nasfile, const char *str=0, long int *pstr=0, int warn=1)

  construct a BDF_Card from a file

- ~BDF_Card ()
- const char * Field (int i) const
- int Field (int i, Int *value) const

  reads an integer from the value in the ith field of the BDF card

- int Field (int i, double *value) const

  reads a double from the value in the ith field of the BDF card

- int NFields () const
- enum EType ElemType () const

  returns the type of this card

- int LoadType () const

  returns the load type of the card

- int PCardType () const

  returns the type of property card

- int NsetType () const

  returns the type of node set

- int chktypes (const char *types, int wordlen, int ncomp) const

determines the type of card

- void Echo () const

  echo BDF card to stdout

- void Write (FILE *) const

  write a BDF card to the log file

- int isBlank (int ifield) const

  returns true if the entire card is blank

- int isInt (int ifield) const

  returns true if the field is an integer

- int isReal (int i) const

  returns 1 if field is nonblank real

- int isContinue (int i) const

  returns 1 if field is a continuation field

- void MarkUsed () const

  mark a BDF card as being used.

- void MarkNotUsed () const

  mark a BDF card as NOT being used.

- int isUsed () const

  returns true if the card has been used

- void PutField (FILE *out, const char *str, int idx, double dflt=0) const

  print out a field. replace with default if needed

- bool StrMatch (int ifield, const char *str) const

  compare field with a string. Avoid whitespace and case issues.

- bool StrMatch (const char *str) const

  compare field with a string. Avoid whitespace and case issues.

- void Concat (char *buf, int len) const

  sets buf to the concatenated string containing the data in the

- void AppendField (const char *buf)

  append a field to an existing card.

- BDF_Card & operator= (const BDF_Card &)

  copy operator

6.1.1 Detailed Description

in memory representation of nastran BDF card and continuations.

Definition at line 48 of file BDF_Card.H.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 BDF_Card::BDF_Card () [inline]

Definition at line 58 of file BDF_Card.H.

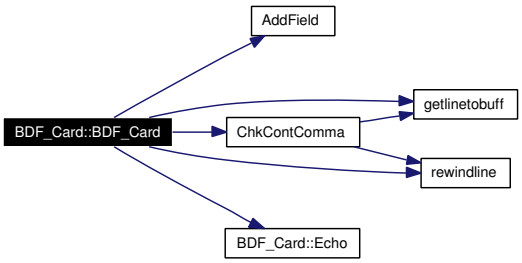
6.1.2.2 BDF_Card::BDF_Card (FILE * *nasfile*, const char * *str* = 0, long int * *pstr* = 0, int *warn* = 1)

construct a BDF_Card from a file

Definition at line 181 of file BDF_Card.C.

References AddField(), blank8, buff, CARD_LIST_SIZE, ChkContComma(), Echo(), getlinetobuff(), prevloc, rewindline(), and tmpcard.

Here is the call graph for this function:



6.1.2.3 BDF_Card::~BDF_Card () [inline]

Definition at line 68 of file BDF_Card.H.

6.1.3 Member Function Documentation

6.1.3.1 void BDF_Card::AppendField (const char * *buf*)

append a field to an existing card.

Definition at line 934 of file BDF_Card.C.

6.1.3.2 int BDF_Card::chktypes (const char * *types*, int *wordlen*, int *ncomp*) const

determines the type of card

Definition at line 631 of file BDF_Card.C.

Referenced by ElemType(), LoadType(), NsetType(), PCardType(), write_salinas::wrt_materials(), and write_presto::wrt_materials().

6.1.3.3 void BDF_Card::Concat (char * *buf*, int *len*) const

sets buf to the concatenated string containing the data in the

Definition at line 893 of file BDF_Card.C.

References Field().

Referenced by StrMatch(), and SuperList::SuperList().

Here is the call graph for this function:

6.1.3.4 void BDF_Card::Echo () const

echo BDF card to stdout

Definition at line 431 of file BDF_Card.C.

Referenced by BDF_Card(), coords::coords(), and sideset::sideset().

6.1.3.5 enum EType BDF_Card::ElemType () const

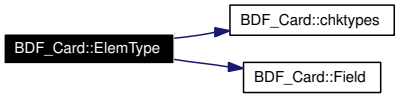
returns the type of this card

Definition at line 510 of file BDF_Card.C.

References chktypes(), Field(), NELEMTYPES, and NOT_ELEM.

Referenced by sideset::AddPLoad4(), coordframe::coordframe(), NasCards::CountCards(), and elem_block::elem_block().

Here is the call graph for this function:



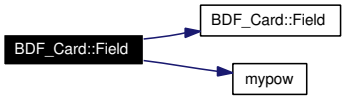
6.1.3.6 int BDF_Card::Field (int *i*, double * *value*) const

reads a double from the value in the ith field of the BDF card

Definition at line 472 of file BDF_Card.C.

References Field(), and mypow().

Here is the call graph for this function:



6.1.3.7 int BDF_Card::Field (int *i*, Int * *value*) const

reads an integer from the value in the ith field of the BDF card

Definition at line 460 of file BDF_Card.C.

References Field().

Here is the call graph for this function:

6.1.3.8 const char* BDF_Card::Field (int *i*) const [inline]

Definition at line 75 of file BDF_Card.H.

Referenced by sideset::AddPLoad2(), sideset::AddPLoad4(), Concat(), coordframe::coordframe(), coords::coords(), NasCards::CountCards(), elem_block::DimensCheck(), ElemType(), Field(), NasCards::FindBdf(), coordframe::GetBasisVectors(), coordframe::GetTransMatrix(), isInt(), isReal(), NasCards::NasCards(), NasCards::NotUsed(), NsetType(), write_salinas::pbarl(), write_presto::pbarl(), write_salinas::Pbush(), write_presto::Pbush(), PCardType(), pid_cards::pid_cards(), PutDouble(), elem_block::SetDefaultAttributes(), sideset::sideset(), StrMatch(), coordframe::TransformToBasic(), Write(), exodus_file::write_cframe(), write_salinas::write_freq(), write_presto::write_freq(), write_salinas::write_params(), write_presto::write_params(), write_salinas::write_rbe3(), write_presto::write_rbe3(), write_salinas::wrt_cframe(), write_presto::wrt_cframe(), write_salinas::wrt_materials(), write_presto::wrt_materials(), write_salinas::wrt_mpc(), write_presto::wrt_mpc(), write_salinas::wrt_pcomp(), write_presto::wrt_pcomp(), and write_presto::wrt_sections().

6.1.3.9 int BDF_Card::isBlank (int *ifield*) const

returns true if the entire card is blank

Definition at line 661 of file BDF_Card.C.

Referenced by coords::coords(), isInt(), isReal(), write_salinas::write_rbe3(), and write_presto::write_rbe3().

6.1.3.10 int BDF_Card::isContinue (int *i*) const

returns 1 if field is a continuation field

Definition at line 705 of file BDF_Card.C.

Referenced by write_salinas::write_rbe3(), and write_presto::write_rbe3().

6.1.3.11 int BDF_Card::isInt (int *ifield*) const

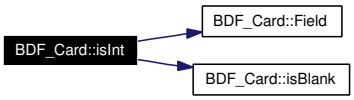
returns true if the field is an integer

Definition at line 676 of file BDF_Card.C.

References Field(), and isBlank().

Referenced by NasCards::CountCards().

Here is the call graph for this function:



6.1.3.12 int BDF_Card::isReal (int *i*) const

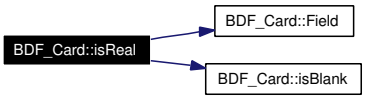
returns 1 if field is nonblank real

Definition at line 689 of file BDF_Card.C.

References Field(), and isBlank().

Referenced by write_salinas::write_params(), write_presto::write_params(), write_salinas::write_rbe3(), and write_presto::write_rbe3().

Here is the call graph for this function:



6.1.3.13 int BDF_Card::isUsed () const

returns true if the card has been used

Definition at line 735 of file BDF_Card.C.

Referenced by NasCards::NotUsed(), and write_presto::wrt_sections().

6.1.3.14 int BDF_Card::LoadType () const

returns the load type of the card

Definition at line 574 of file BDF_Card.C.

References chktypes(), and NDLOADTYPES.

Referenced by NasCards::CountCards(), and sideset::sideset().

Here is the call graph for this function:

6.1.3.15 void BDF_Card::MarkNotUsed () const

mark a BDF card as NOT being used.

Definition at line 728 of file BDF_Card.C.

6.1.3.16 void BDF_Card::MarkUsed () const

mark a BDF card as being used.

Definition at line 718 of file BDF_Card.C.

Referenced by coords::coords(), coordframe::GetTransMatrix(), NasCards::NasCards(), elem_block::SetDefaultAttributes(), sideset::sideset(), coordframe::TransformToBasic(), exodus_file::write_cframe(), write_salinas::write_freq(), write_presto::write_freq(), write_salinas::write_params(), write_presto::write_params(), write_salinas::wrt_cframe(), write_presto::wrt_cframe(), write_salinas::wrt_materials(), write_presto::wrt_materials(), write_salinas::wrt_mpc(), and write_presto::wrt_mpc().

6.1.3.17 int BDF_Card::NFields () const [inline]

Definition at line 85 of file BDF_Card.H.

Referenced by sideset::AddPLoad2(), coords::coords(), NasCards::CountCards(), NasCards::FindBdf(), NasCards::NotUsed(), operator=(), write_salinas::Pbush(), write_presto::Pbush(), StrMatch(), SuperList::SuperList(), Write(), exodus_file::write_cframe(), write_salinas::wrt_cframe(), write_presto::wrt_cframe(), write_salinas::wrt_materials(), write_presto::wrt_materials(), write_salinas::wrt_mpc(), write_presto::wrt_mpc(), and write_presto::wrt_sections().

6.1.3.18 int BDF_Card::NsetType () const

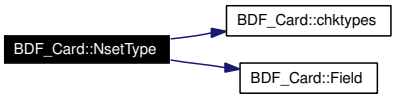
returns the type of node set

Definition at line 611 of file BDF_Card.C.

References chktypes(), Field(), NNSETTYPES, and TLEN.

Referenced by NasCards::CountCards().

Here is the call graph for this function:



6.1.3.19 BDF_Card & BDF_Card::operator= (const BDF_Card &)

copy operator

Definition at line 908 of file BDF_Card.C.

References mUsed, and NFields().

Here is the call graph for this function:

6.1.3.20 int BDF_Card::PCardType () const

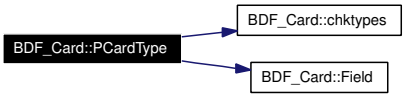
returns the type of property card

Definition at line 592 of file BDF_Card.C.

References chktypes(), Field(), PCARDTYPES, and TLEN.

Referenced by NasCards::CountCards(), pid_cards::pid_cards(), and write_presto::wrt_sections().

Here is the call graph for this function:



6.1.3.21 void BDF_Card::PutField (FILE * *out*, const char * *str*, int *idx*, double *dflt* = 0) const

print out a field. replace with default if needed

Definition at line 832 of file BDF_Card.C.

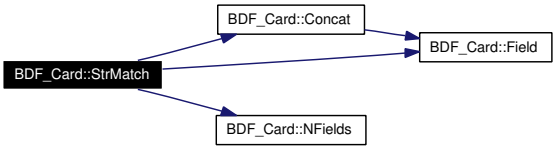
6.1.3.22 bool BDF_Card::StrMatch (const char * *str*) const

compare field with a string. Avoid whitespace and case issues.

Definition at line 871 of file BDF_Card.C.

References Concat(), Field(), and NFields().

Here is the call graph for this function:



6.1.3.23 bool BDF_Card::StrMatch (int *ifield*, const char * *str*) const

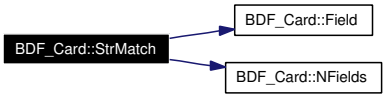
compare field with a string. Avoid whitespace and case issues.

Definition at line 849 of file BDF_Card.C.

References Field(), and NFields().

Referenced by SuperList::SuperList(), write_salinas::write_params(), and write_presto::write_params().

Here is the call graph for this function:



6.1.3.24 void BDF_Card::Write (FILE *) const

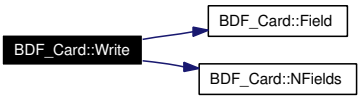
write a BDF card to the log file

Definition at line 741 of file BDF_Card.C.

References blank8, Field(), and NFields().

Referenced by NasCards::NotUsed().

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- BDF_Card.H
- BDF_Card.C

6.2 CaseCard Class Reference

a container to store a single case control card as a long string

`#include <CaseCard.H>`

Public Member Functions

- CaseCard ()

  default constructor

- ~CaseCard ()
- int ReadData (const char *str, char *buf)

  fill a CaseCard with data

- const char * CString () const
- bool Usage () const
- void SetUsed ()
- void SetNull ()

  reset card to original

6.2.1 Detailed Description

a container to store a single case control card as a long string

Definition at line 16 of file CaseCard.H.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 CaseCard::CaseCard ()

default constructor

Definition at line 11 of file CaseCard.C.

6.2.2.2 CaseCard::~CaseCard () [inline]

Definition at line 23 of file CaseCard.H.

6.2.3 Member Function Documentation

6.2.3.1 const char* CaseCard::CString () const [inline]

Definition at line 29 of file CaseCard.H.

Referenced by NasCards::FindCaseKeywordId(), NasCards::MatchCaseCard(), and NasCards::NotUsed().

6.2.3.2 int CaseCard::ReadData (const char * *str*, char * *buf*)

fill a CaseCard with data

Definition at line 22 of file CaseCard.C.

References SetNull().

Referenced by NasCards::NasCards().

Here is the call graph for this function:

6.2.3.3 void CaseCard::SetNull ()

reset card to original

Definition at line 101 of file CaseCard.C.

Referenced by NasCards::NasCards(), and ReadData().

6.2.3.4 void CaseCard::SetUsed () [inline]

Definition at line 34 of file CaseCard.H.

6.2.3.5 bool CaseCard::Usage () const [inline]

Definition at line 31 of file CaseCard.H.

The documentation for this class was generated from the following files:

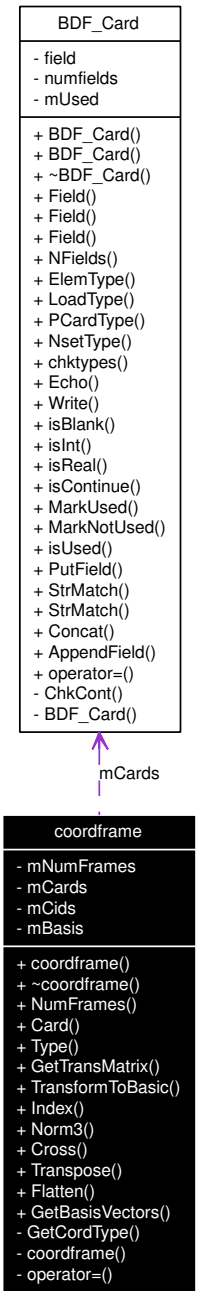
- CaseCard.H
- CaseCard.C

6.3 coordframe Class Reference

A class to hold the array of all coordinate frames.

`#include <coordframe.H>`

Collaboration diagram for coordframe:



Public Member Functions

- coordframe (const NasCards *)

constructs the coordinate frames from nastran cards

- ~coordframe ()

  destructs all coordinate frames

- int NumFrames () const
- const BDF_Card * Card (int index) const
- enum Cord_Type Type (int index) const
- int GetTransMatrix (double **R, int cid, const double location[3]) const

  Computes the transformation matrix, R, for a coordinate frame.

- int TransformToBasic (Real x[3], int cid, int depth=0, bool no_origin=0) const

  transforms a vector to basic coordinate frame

- int Index (int cid) const

  returns the index given the frame id.

- void Norm3 (double x[3]) const

  normalizes a 3 vector

- void Cross (double c[3], const double a[3], const double b[3]) const

  compute vector cross product

- void Transpose (double **M) const

  transpose matrix

- void Flatten ()

  store the basis vectors, e1, e2, e3 and origin for each frame.

- int GetBasisVectors (double e1[3], double e2[3], double e3[3], double origin[3], int idx, int depth) const

  compute the normalized basis vectors and origin

6.3.1 Detailed Description

A class to hold the array of all coordinate frames.

Definition at line 18 of file coordframe.H.

6.3.2 Constructor & Destructor Documentation

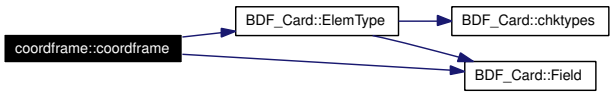
6.3.2.1 coordframe::coordframe (const NasCards *)

constructs the coordinate frames from nastran cards

Definition at line 21 of file coordframe.C.

References CORD, BDF_Card::ElemType(), and BDF_Card::Field().

Here is the call graph for this function:



6.3.2.2 coordframe::~coordframe ()

destructs all coordinate frames

Definition at line 61 of file coordframe.C.

6.3.3 Member Function Documentation

6.3.3.1 const BDF_Card* coordframe::Card (int *index*) const [inline]

Definition at line 36 of file coordframe.H.

Referenced by GetBasisVectors(), GetTransMatrix(), and TransformToBasic().

6.3.3.2 void coordframe::Cross (double *c*[3], const double *a*[3], const double *b*[3]) const

compute vector cross product

Definition at line 327 of file coordframe.C.

Referenced by GetBasisVectors(), and GetTransMatrix().

6.3.3.3 void coordframe::Flatten ()

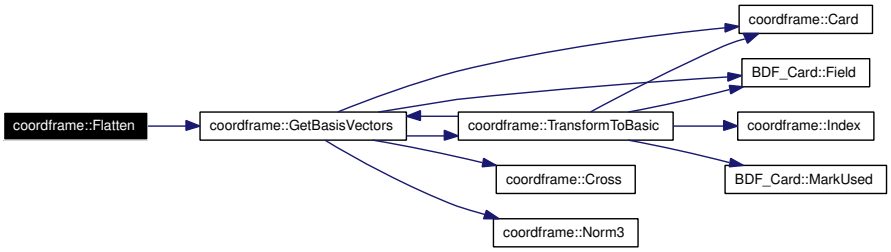
store the basis vectors, e1, e2, e3 and origin for each frame.

Definition at line 363 of file coordframe.C.

References GetBasisVectors(), and xcid.

Referenced by main().

Here is the call graph for this function:



6.3.3.4 int coordframe::GetBasisVectors (double *e1*[3], double *e2*[3], double *e3*[3], double *origin*[3], int *idx*, int *depth*) const

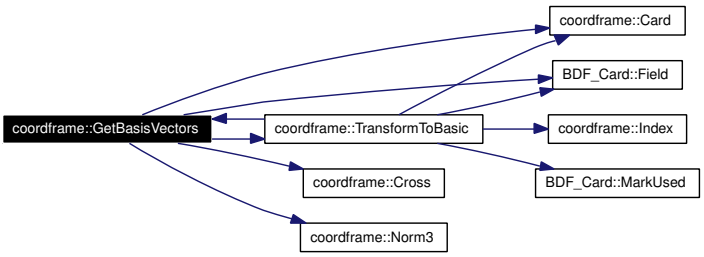
compute the normalized basis vectors and origin

Definition at line 382 of file coordframe.C.

References Card(), Cross(), BDF_Card::Field(), Norm3(), TransformToBasic(), and xcid.

Referenced by Flatten(), and TransformToBasic().

Here is the call graph for this function:



6.3.3.5 int coordframe::GetTransMatrix (double ** *R*, int *cid*, const double *location*[3]) const

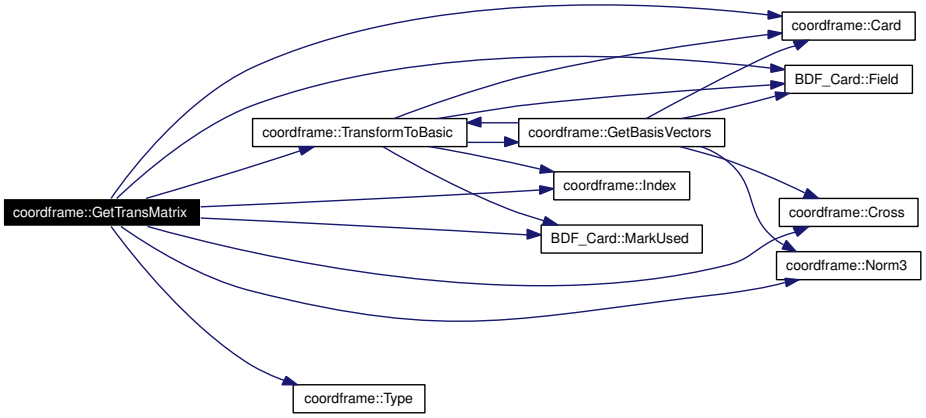
Computes the transformation matrix, R, for a coordinate frame.

Definition at line 165 of file coordframe.C.

References Card(), CORD_CYL, CORD_RECT, CORD_SPH, Cross(), BDF_Card::Field(), Index(), BDF_Card::MarkUsed(), Norm3(), TransformToBasic(), Type(), and xcid.

Referenced by coords::ConvertMassMoment().

Here is the call graph for this function:



6.3.3.6 int coordframe::Index (int *cid*) const

returns the index given the frame id.

Definition at line 352 of file coordframe.C.

Referenced by coords::coords(), GetTransMatrix(), and TransformToBasic().

6.3.3.7 void coordframe::Norm3 (double *x*[3]) const

normalizes a 3 vector

Definition at line 314 of file coordframe.C.

References xcid.

Referenced by GetBasisVectors(), and GetTransMatrix().

6.3.3.8 int coordframe::NumFrames () const [inline]

Definition at line 33 of file coordframe.H.

6.3.3.9 int coordframe::TransformToBasic (Real *x*[3], int *cid*, int *depth* = 0, bool *no_origin* = 0) const

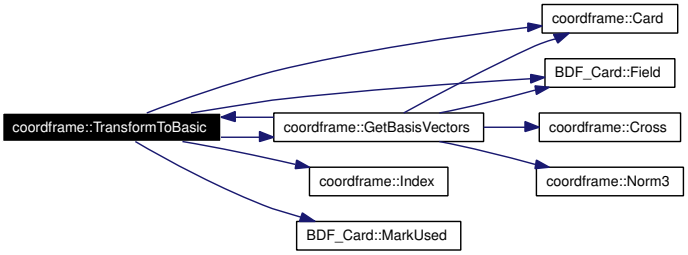
transforms a vector to basic coordinate frame

Definition at line 95 of file coordframe.C.

References Card(), CORD_CYL, CORD_RECT, CORD_SPH, BDF_Card::Field(), GetBasisVectors(), Index(), M_PI, BDF_Card::MarkUsed(), and xcid.

Referenced by GetBasisVectors(), GetTransMatrix(), coords::TransformToBasic(), and coords::TransformToBasicFromCID().

Here is the call graph for this function:



6.3.3.10 void coordframe::Transpose (double ** *M*) const

transpose matrix

Definition at line 336 of file coordframe.C.

Referenced by coords::ConvertMassMoment().

6.3.3.11 enum Cord_Type coordframe::Type (int *index*) const [inline]

Definition at line 39 of file coordframe.H.

Referenced by GetTransMatrix().

The documentation for this class was generated from the following files:

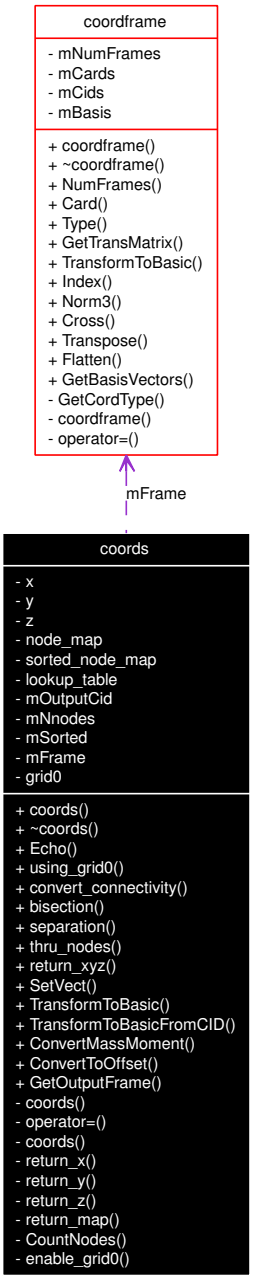
- coordframe.H
- coordframe.C

6.4 coords Class Reference

class to store a single coordinate frame

`#include <coords.H>`

Collaboration diagram for coords:



Public Member Functions

- coords (const NasCards *cards, const coordframe *cframe)

- ~coords ()
- void Echo ()
- Int using_grid0 () const
- void convert_connectivity (Int *con, Int nelem) const
- Int bisection (Int a) const
- Int separation (Int a, Int b) const
- int thru_nodes (Int *node_list, Int a, Int b) const
- void return_xyz (double v[3], int i) const
- void SetVect (Int GA, Int G0, double *x, double *y, double *z) const
- void TransformToBasic (double *x, Int GA) const
- void TransformToBasicFromCID (double *x, Int cid) const
- Int ConvertMassMoment (Int Grid, double *MassMoment) const
- Int ConvertToOffset (double *vec, Int GA) const
- Int GetOutputFrame (Int Grid) const

Friends

- class exodus_file

6.4.1 Detailed Description

class to store a single coordinate frame

Definition at line 19 of file coords.H.

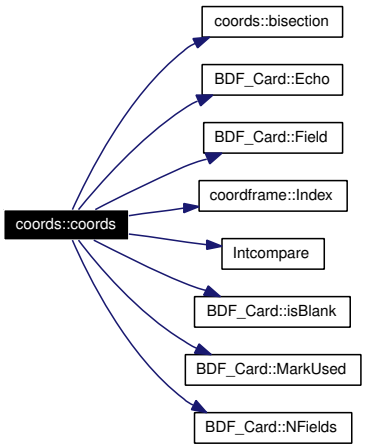
6.4.2 Constructor & Destructor Documentation

6.4.2.1 coords::coords (const NasCards * *cards*, const coordframe * *cframe*)

Definition at line 25 of file coords.C.

References bisection(), BDF_Card::Echo(), BDF_Card::Field(), coordframe::Index(), Intcompare(), BDF_Card::isBlank(), BDF_Card::MarkUsed(), and BDF_Card::NFields().

Here is the call graph for this function:



6.4.2.2 coords::~coords ()

Definition at line 444 of file coords.C.

6.4.3 Member Function Documentation

6.4.3.1 Int coords::bisection (Int *a*) const

Definition at line 210 of file coords.C.

Referenced by ConvertMassMoment(), ConvertToOffset(), coords(), GetOutputFrame(), separation(), SetVect(), and thru_nodes().

6.4.3.2 void coords::convert_connectivity (Int * *con*, Int *nelem*) const

Definition at line 159 of file coords.C.

References bisection().

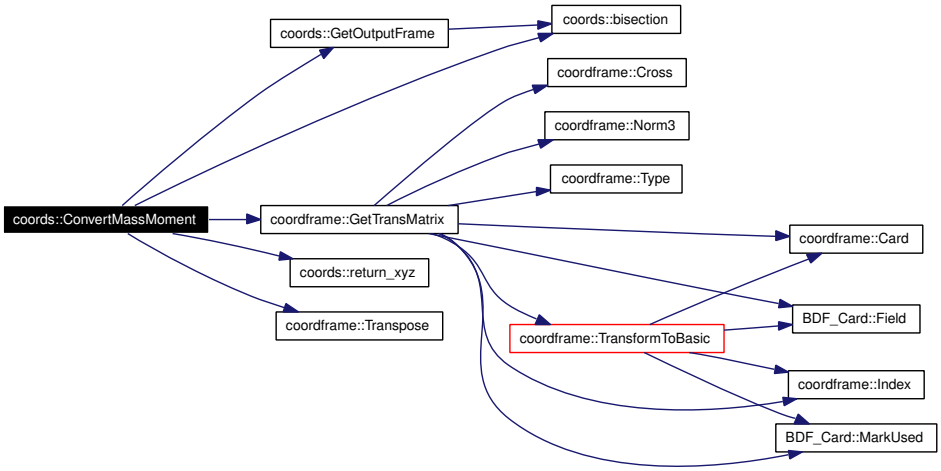
Here is the call graph for this function:

6.4.3.3 Int coords::ConvertMassMoment (Int *Grid*, double * *MassMoment*) const

Definition at line 340 of file coords.C.

References bisection(), GetOutputFrame(), coordframe::GetTransMatrix(), return_xyz(), and coordframe::Transpose().

Here is the call graph for this function:



6.4.3.4 Int coords::ConvertToOffset (double * *vec*, Int *GA*) const

Definition at line 414 of file coords.C.

References bisection().

Here is the call graph for this function:

6.4.3.5 void coords::Echo ()

Definition at line 145 of file coords.C.

6.4.3.6 Int coords::GetOutputFrame (Int *Grid*) const

Definition at line 325 of file coords.C.

References bisection().

Referenced by ConvertMassMoment(), and TransformToBasic().

Here is the call graph for this function:

6.4.3.7 void coords::return_xyz (double *v*[3], int *i*) const

Definition at line 229 of file coords.C.

Referenced by ConvertMassMoment().

6.4.3.8 Int coords::separation (Int *a*, Int *b*) const

Definition at line 261 of file coords.C.

References bisection().

Referenced by thru_nodes().

Here is the call graph for this function:

6.4.3.9 void coords::SetVect (Int *GA*, Int *G0*, double * *x*, double * *y*, double * *z*) const

Definition at line 285 of file coords.C.

References bisection().

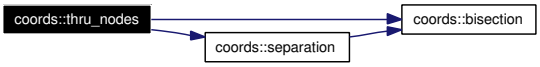
Here is the call graph for this function:

6.4.3.10 int coords::thru_nodes (Int * *node_list*, Int *a*, Int *b*) const

Definition at line 272 of file coords.C.

References bisection(), and separation().

Here is the call graph for this function:

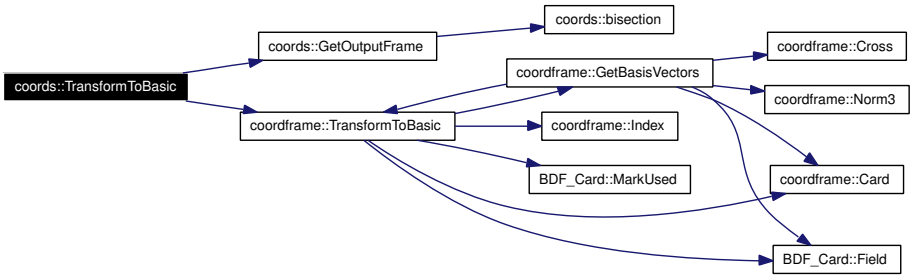


6.4.3.11 void coords::TransformToBasic (double * *x*, Int *GA*) const

Definition at line 302 of file coords.C.

References GetOutputFrame(), and coordframe::TransformToBasic().

Here is the call graph for this function:

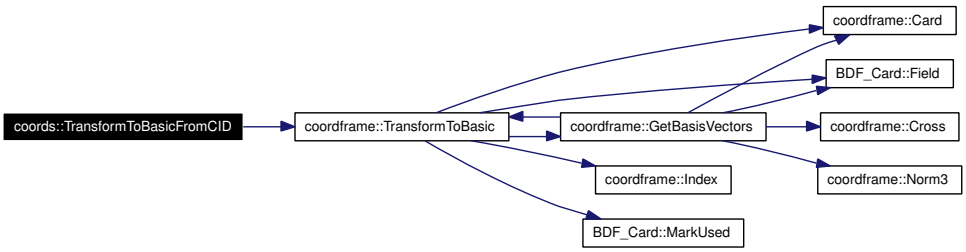


6.4.3.12 void coords::TransformToBasicFromCID (double * *x*, Int *cid*) const

Definition at line 316 of file coords.C.

References coordframe::TransformToBasic().

Here is the call graph for this function:



6.4.3.13 Int coords::using_grid0 () const [inline]

Definition at line 50 of file coords.H.

Referenced by main().

6.4.4 Friends And Related Function Documentation

6.4.4.1 friend class exodus_file [friend]

Definition at line 48 of file coords.H.

The documentation for this class was generated from the following files:

- coords.H
- coords.C

6.5 elem_block Class Reference

data for a single exodus element block

`#include <elem_block.H>`

Collaboration diagram for elem_block:


BDF_Card
- field
- numfields
- mUsed
+ BDF_Card()
+ BDF_Card()
+ ~BDF_Card()
+ Field()
+ Field()
+ Field()
+ NFields()
+ ElemType()
+ LoadType()
+ PCardType()
+ NsetType()
+ chktypes()
+ Echo()
+ Write()
+ isBlank()
+ isInt()
+ isReal()
+ isContinue()
+ MarkUsed()
+ MarkNotUsed()
+ isUsed()
+ PutField()
+ StrMatch()
+ StrMatch()
+ Concat()
+ AppendField()
+ operator=()
- ChkCont()
- BDF_Card()
SuperList
- mSupers
- mNSupers
- mNumSuperBlocks
+ SuperList()
+ ~SuperList()
+ WriteExo()
+ WriteSalinas()
+ NumSuperElems()
+ NumNodes()
+ get_connectivity()
+ WriteBlock()
- GetSuper()
- writemap()
- writematrices()
- ConcatMap()
- get_concat_connectivity()
- CountEig()
- WriteCBmap()
- SuperList()
- operator=()
- SuperList()
mSuperElemList
mPcardPtr
mFirstElemCard
elem_block
- num_elem
- number_pids
- number_properties
- elem_id_map
- elem_pid_map
- node_pointer_list
- property_pointer_list
- pid_list
- elements_per_pid
- properties_per_pid
- elements_to_pid
- properties_to_pid
- nodes_per_pid
- exnamep
- mFirstElemCard
- mNum_rigid_blks
- mRpid_list
- mRblk_elem_cnt
- mRigid_Conn
- mUseOffset
- mMin_elemid
- mMax_elemid
- mRevElemId
- mSuperElemList
- mPcardPtr
+ elem_block()
+ ~elem_block()
+ num_pids()
+ pidlist()
+ ElemName()
+ num_elems()
+ NumElem()
+ num_nodes()
+ NumNode()
+ NumNodesPerElem()
+ num_attributes()
+ convert_nodes()
+ SetDefaultAttributes()
+ ReformAttr()
+ ElemType()
+ FirstElemCard()
+ ElemIds()
+ NumStdElem()
+ CountRBlks()
+ RblkId()
+ RblkNumElems()
+ RblkNumElems()
+ RblkType()
+ BldRevMap()
+ GetExoId()
+ GetNodes()
+ GetBlk()
+ GetSide()
+ NSuper()
+ Super()
+ SetPcardPtr()
+ FindMatchPcard()
+ DimensCheck()
+ GetPcard()
- convert_pid()
- store_elem_connectivity()
- store_elem_attributes()
- elem_block()
- operator=()
- elem_block()
- return_attributes()
- return_elem_block()

Public Member Functions

- elem_block (const NasCards *cards, const pid_cards *pcards, const coords *Coords, SuperList *selems)

  construct the element blocks from nastran cards

- ~elem_block ()
- Int num_pids () const

  return number of property ids

- const Int * pidlist () const

  return an array of property ids

- const char * ElemName (int blk) const
- const Int * num_elems () const

  return the number of elements in an element block

- Int NumElem (int blk) const
- const Int * num_nodes () const

  return the number of nodes per element in an element block

- Int NumNode (int blk) const
- Int NumNodesPerElem (int blknum) const
- const Int * num_attributes () const
- void convert_nodes (const coords *input_coords)
- void SetDefaultAttributes (int blk, const BDF_Card *pcard, int pid)

  sets default element attributes

- void ReformAttr (int blk_cntr, int new_nattr, int old_nattr)
- enum EType ElemType (int blk) const
- const BDF_Card * FirstElemCard (int blk_index) const
- const Int * ElemIds () const
- Int NumStdElem () const
- Int CountRBlks () const
- int RblkId (int i) const
- int RblkNumElems (int i) const
- int RblkNumElems () const
- enum EType RblkType (int i) const
- void BldRevMap ()

  builds a reverse element map. required for GetExoId to work

- Int GetExoId (Int bdf_elem_id) const

  returns the exodus ID (0:n-1) or -1

- void GetNodes (Int *nodes, Int exo_elem_id) const

  populates nodes

- Int GetBlk (Int exo_elem_id, Int *blk_elem_id=0) const

  returns the exodus blk offset and the element offset within the blk

- int GetSide (int blknum, int elemoff, int g1, int g2) const
- Int NSuper () const

  return number of superelement blocks

- const SuperList * Super () const
- void SetPcardPtr (const pid_cards *pcards, const NasCards *cards)

  create an array to store a pointer to property card. One entry per block.

- const BDF_Card * FindMatchPcard (const pid_cards *pcards, const NasCards *cards, int pid, enum EType etype) const

  match one pid and its element type with a property card.

- bool DimensCheck (const BDF_Card *c, enum EType etype) const

  compare dimensionality of a property card and an element type

- const BDF_Card * GetPcard (int i) const

  return a pointer to the property card for block idx.

Friends

- class exodus_file

6.5.1 Detailed Description

data for a single exodus element block

Definition at line 17 of file elem_block.H.

6.5.2 Constructor & Destructor Documentation

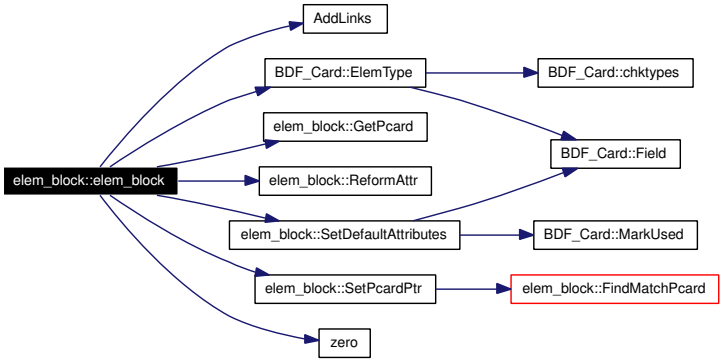
6.5.2.1 elem_block::elem_block (const NasCards * *cards*, const pid_cards * *pcards*, const coords * *Coords*, SuperList * *selems*)

construct the element blocks from nastran cards

Definition at line 127 of file elem_block.C.

References AddLinks(), element_nnodes_lookup, BDF_Card::ElemType(), exoname, exoname2, GetPcard(), MAX_ETYPE, noffset_attr_lookup, nproperties_lookup, RBAR, RBE1, RBE2, RBE3, ReformAttr(), RROD, RTRPLT, SetDefaultAttributes(), SetPcardPtr(), SUPER_ELEMENT, use_short_names, and zero().

Here is the call graph for this function:



6.5.2.2 elem_block::~elem_block ()

Definition at line 97 of file elem_block.C.

6.5.3 Member Function Documentation

6.5.3.1 void elem_block::BldRevMap ()

builds a reverse element map. required for GetExoId to work

Definition at line 918 of file elem_block.C.

6.5.3.2 void elem_block::convert_nodes (const coords * *input_coords*)

Definition at line 407 of file elem_block.C.

References NumNodesPerElem().

Referenced by main().

Here is the call graph for this function:

6.5.3.3 Int elem_block::CountRBlks () const [inline]

Definition at line 182 of file elem_block.H.

6.5.3.4 bool elem_block::DimensCheck (const BDF_Card * *c*, enum EType *etype*) const

compare dimensionality of a property card and an element type

Definition at line 1424 of file elem_block.C.

References BDF_Card::Field().

Referenced by FindMatchPcard().

Here is the call graph for this function:

6.5.3.5 const Int* elem_block::ElemIds () const [inline]

Definition at line 176 of file elem_block.H.

6.5.3.6 const char* elem_block::ElemName (int *blk*) const [inline]

Definition at line 160 of file elem_block.H.

6.5.3.7 enum EType elem_block::ElemType (int *blk*) const [inline]

Definition at line 172 of file elem_block.H.

Referenced by GetSide().

6.5.3.8 const BDF_Card * elem_block::FindMatchPcard (const pid_cards * *pcards*, const NasCards * *cards*, int *pid*, enum EType *etype*) const

match one pid and its element type with a property card.

Definition at line 1403 of file elem_block.C.

References DimensCheck().

Referenced by SetPcardPtr().

Here is the call graph for this function:

6.5.3.9 const BDF_Card* elem_block::FirstElemCard (int *blk_index*) const [inline]

Definition at line 174 of file elem_block.H.

6.5.3.10 Int elem_block::GetBlk (Int *exo_elem_id*, Int * *blk_elem_id* = 0) const

returns the exodus blk offset and the element offset within the blk

Definition at line 977 of file elem_block.C.

Referenced by GetNodes().

6.5.3.11 Int elem_block::GetExoId (Int *bdf_elem_id*) const

returns the exodus ID (0:n-1) or -1

Definition at line 952 of file elem_block.C.

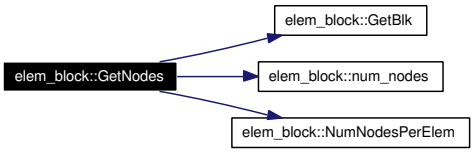
6.5.3.12 void elem_block::GetNodes (Int * *nodes*, Int *exo_elem_id*) const

populates nodes

Definition at line 963 of file elem_block.C.

References GetBlk(), num_nodes(), and NumNodesPerElem().

Here is the call graph for this function:



6.5.3.13 const BDF_Card * elem_block::GetPcard (int *i*) const

return a pointer to the property card for block idx.

Definition at line 1458 of file elem_block.C.

Referenced by elem_block().

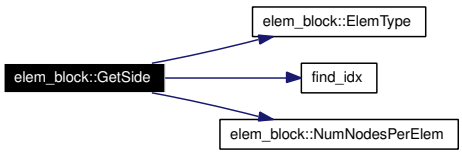
6.5.3.14 int elem_block::GetSide (int *blknum*, int *elemoff*, int *g1*, int *g2*) const

returns the sideid (1:n) of element # elemoff in blk=blknum with reference nodes, global ids=g1 and g2

Definition at line 995 of file elem_block.C.

References CHEX20, CHEX8, CPENTA15, CPENTA6, CTET10, CTET4, ElemType(), find_idx(), and NumNodesPerElem().

Here is the call graph for this function:



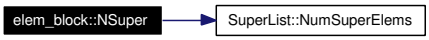
6.5.3.15 Int elem_block::NSuper () const

return number of superelement blocks

Definition at line 1379 of file elem_block.C.

References SuperList::NumSuperElems().

Here is the call graph for this function:



6.5.3.16 const Int * elem_block::num_attributes () const

Definition at line 801 of file elem_block.C.

6.5.3.17 const Int * elem_block::num_elems () const

return the number of elements in an element block

Definition at line 786 of file elem_block.C.

Referenced by NumElem().

6.5.3.18 const Int * elem_block::num_nodes () const

return the number of nodes per element in an element block

Definition at line 794 of file elem_block.C.

Referenced by GetNodes(), and NumNode().

6.5.3.19 Int elem_block::num_pids () const

return number of property ids

Definition at line 762 of file elem_block.C.

6.5.3.20 Int elem_block::NumElem (int *blk*) const [inline]

Definition at line 162 of file elem_block.H.

References num_elems().

Here is the call graph for this function:

6.5.3.21 Int elem_block::NumNode (int *blk*) const [inline]

Definition at line 164 of file elem_block.H.

References num_nodes().

Here is the call graph for this function:

6.5.3.22 Int elem_block::NumNodesPerElem (int *blknum*) const [inline]

Definition at line 165 of file elem_block.H.

Referenced by convert_nodes(), GetNodes(), and GetSide().

6.5.3.23 Int elem_block::NumStdElem () const [inline]

Definition at line 179 of file elem_block.H.

6.5.3.24 const Int * elem_block::pidlist () const

return an array of property ids

Definition at line 770 of file elem_block.C.

6.5.3.25 int elem_block::RblkId (int *i*) const [inline]

Definition at line 185 of file elem_block.H.

6.5.3.26 int elem_block::RblkNumElems () const [inline]

Definition at line 193 of file elem_block.H.

6.5.3.27 int elem_block::RblkNumElems (int *i*) const [inline]

Definition at line 189 of file elem_block.H.

6.5.3.28 enum EType elem_block::RblkType (int *i*) const [inline]

Definition at line 201 of file elem_block.H.

6.5.3.29 void elem_block::ReformAttr (int *blk_cntr*, int *new_nattr*, int *old_nattr*)

removes the last block of attributes. Written to handle the element offsets.

Definition at line 1078 of file elem_block.C.

Referenced by elem_block().

6.5.3.30 void elem_block::SetDefaultAttributes (int *blk*, const BDF_Card * *pcard*, int *pid*)

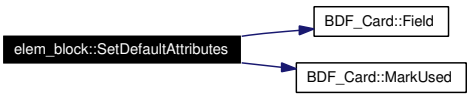
sets default element attributes

Definition at line 809 of file elem_block.C.

References CBAR, CBEAM, CELAS, CONM, CQUAD4, CQUAD8, CROD, CTRIA3, CTRIA6, BDF_-Card::Field(), and BDF_Card::MarkUsed().

Referenced by elem_block().

Here is the call graph for this function:



6.5.3.31 void elem_block::SetPcardPtr (const pid_cards * *pcards*, const NasCards * *cards*)

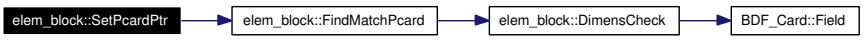
create an array to store a pointer to property card. One entry per block.

Definition at line 1389 of file elem_block.C.

References FindMatchPcard().

Referenced by elem_block().

Here is the call graph for this function:



6.5.3.32 const SuperList* elem_block::Super () const [inline]

Definition at line 223 of file elem_block.H.

6.5.4 Friends And Related Function Documentation

6.5.4.1 friend class exodus_file [friend]

Definition at line 156 of file elem_block.H.

The documentation for this class was generated from the following files:

- elem_block.H
- elem_block.C

6.6 exodus_file Class Reference

data and methods to write an exodus file

`#include <exodus_file.H>`

Public Member Functions

- exodus_file (const char *filename, int io_ws=0)

  construct exodus file from filename

- void initialize (const NasCards *Cards, const coords *input_coords, const elem_block *exodus_elem_block, const nodeset *nodesets, const sideset *sidesets)

  Initialize exodus file.

- void write_nodes (const coords *input_coords)

  write nodal coordinates to exodus file

- void write_elem_blocks (const elem_block *exodus_elem_block)

  write element blocks to exodus file

- void write_nodesets (const nodeset *nodeset_list)

  write nodesets to exodus file

- void write_sidesets (const sideset *sidesets)

  write sidesets to exodus file

- void write_cframe (const coordframe *cframe) const

  write coordinate frame to exodus file

- void close ()

  close exodus file

6.6.1 Detailed Description

data and methods to write an exodus file

Definition at line 20 of file exodus_file.H.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 exodus_file::exodus_file (const char * *filename*, int *io_ws* = 0)

construct exodus file from filename

Definition at line 32 of file exodus_file.C.

6.6.3 Member Function Documentation

6.6.3.1 void exodus_file::close ()

close exodus file

Definition at line 292 of file exodus_file.C.

Referenced by main().

6.6.3.2 void exodus_file::initialize (const NasCards * *Cards*, const coords * *input_coords*, const elem_block * *exodus_elem_block*, const nodeset * *nodesets*, const sideset * *sidesets*)

Initialize exodus file.

Definition at line 49 of file exodus_file.C.

Referenced by main().

6.6.3.3 void exodus_file::write_cframe (const coordframe * *cframe*) const

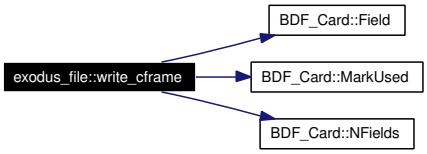
write coordinate frame to exodus file

Definition at line 302 of file exodus_file.C.

References CORD_CYL, CORD_RECT, CORD_SPH, BDF_Card::Field(), BDF_Card::MarkUsed(), and BDF_Card::NFields().

Referenced by main().

Here is the call graph for this function:



6.6.3.4 void exodus_file::write_elem_blocks (const elem_block * *exodus_elem_block*)

write element blocks to exodus file

Definition at line 100 of file exodus_file.C.

Referenced by main().

6.6.3.5 void exodus_file::write_nodes (const coords * *input_coords*)

write nodal coordinates to exodus file

Definition at line 83 of file exodus_file.C.

Referenced by main().

6.6.3.6 void exodus_file::write_nodesets (const nodeset * *nodeset_list*)

write nodesets to exodus file

Definition at line 221 of file exodus_file.C.

Referenced by main().

6.6.3.7 void exodus_file::write_sidesets (const sideset * *sidesets*)

write sidesets to exodus file

Definition at line 258 of file exodus_file.C.

Referenced by main().

The documentation for this class was generated from the following files:

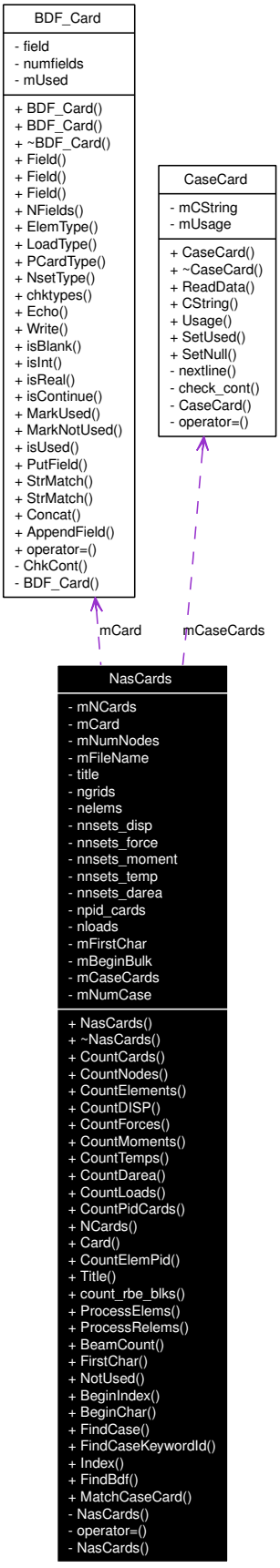
- exodus_file.H
- exodus_file.C

6.7 NasCards Class Reference

A class to hold all bulk data and case control cards.

`#include <NasCards.H>`

Collaboration diagram for NasCards:


BDF_Card
- field
- numfields
- mUsed
+ BDF_Card()
+ BDF_Card()
+ ~BDF_Card()
+ Field()
+ Field()
+ Field()
+ NFields()
+ ElemType()
+ LoadType()
+ PCardType()
+ NsetType()
+ chktypes()
+ Echo()
+ Write()
+ isBlank()
+ isInt()
+ isReal()
+ isContinue()
+ MarkUsed()
+ MarkNotUsed()
+ isUsed()
+ PutField()
+ StrMatch()
+ StrMatch()
+ Concat()
+ AppendField()
+ operator=()
- ChkCont()
- BDF_Card()
CaseCard
- mCString
- mUsage
+ CaseCard()
+ ~CaseCard()
+ ReadData()
+ CString()
+ Usage()
+ SetUsed()
+ SetNull()
- nextline()
- check_cont()
- CaseCard()
- operator=()
mCard
mCaseCards
NasCards
- mNCards
- mCard
- mNumNodes
- mFileName
- title
- ngrids
- nelems
- nnsets_disp
- nnsets_force
- nnsets_moment
- nnsets_temp
- nnsets_darea
- npid_cards
- nloads
- mFirstChar
- mBeginBulk
- mCaseCards
- mNumCase
+ NasCards()
+ ~NasCards()
+ CountCards()
+ CountNodes()
+ CountElements()
+ CountDISP()
+ CountForces()
+ CountMoments()
+ CountTemps()
+ CountDarea()
+ CountLoads()
+ CountPidCards()
+ NCards()
+ Card()
+ CountElemPid()
+ Title()
+ count_rbe_blks()
+ ProcessElems()
+ ProcessRelems()
+ BeamCount()
+ FirstChar()
+ NotUsed()
+ BeginIndex()
+ BeginChar()
+ FindCase()
+ FindCaseKeywordId()
+ Index()
+ FindBdf()
+ MatchCaseCard()
- NasCards()
- operator=()
- NasCards()

Public Member Functions

- NasCards (const char *filen)
  default constructor
- ~NasCards ()
- void CountCards ()
  counts BDF cards only
- int CountNodes () const
  count the number of nodes
- int CountElements () const
  count the number of elements in nastran file
- int CountDISP () const
  count the number of displacement type nodesets
- int CountForces () const
  count the number of force type nodesets
- int CountMoments () const
  count the number of moment type nodesets
- int CountTemps () const
  count the number of temperature type nodesets
- int CountDarea () const
  count the number of DAREA cards
- int CountLoads () const
  count the number of load cards
- int CountPidCards () const
  count the number of property cards
- int NCards () const
- const BDF_Card * Card (int i) const
- void CountElemPid (int &numelems, int &npids)
- const char * Title () const
  return the title from nastran input
- int count_rbe_blks (int npids, long *mpcoff, int *blkmap, int &numrelem)
- void ProcessElems (int ex_id, int numelems, int npids, int nrpids, int *node_map, int nnodes, int numrelem, const double x[ ], const double y[ ], const double z[ ], int *blkcnt, int *blknodes, int *blkmap, int *blktyp)
- void ProcessRelems (int ex_id, int npids, int nrpids, int *blkmap, int *node_map, int nnodes)
- int BeamCount (const BDF_Card *c, int idx) const
- const char * FirstChar () const
- void NotUsed (const char *, int) const

echo cards to a file

- Int BeginIndex () const
- const char * BeginChar () const
- const CaseCard * FindCase (const char *, Int &idx) const

  find a card in the case control section that begins with str.

- bool FindCaseKeywordId (const char *keyword, const char *id) const

  find a keyword=id combination in case control.

- const BDF_Card * Index (const char *p) const
- const BDF_Card * FindBdf (Int &idx, const char *str, Int id=0) const

  find a card with matching field and ID in the BDF cards

- bool MatchCaseCard (Int icase, const char *str) const

  check if the first nonblank case control card matches the string.

6.7.1 Detailed Description

A class to hold all bulk data and case control cards.

Definition at line 11 of file NasCards.H.

6.7.2 Constructor & Destructor Documentation

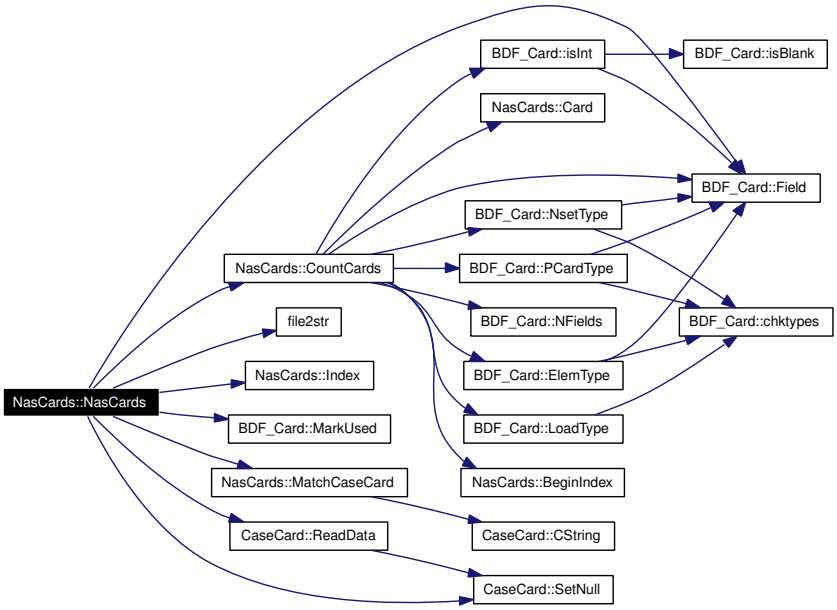
6.7.2.1 NasCards::NasCards (const char * *filen*)

default constructor

Definition at line 72 of file NasCards.C.

References CountCards(), BDF_Card::Field(), file2str(), Index(), BDF_Card::MarkUsed(), MatchCaseCard(), CaseCard::ReadData(), and CaseCard::SetNull().

Here is the call graph for this function:



6.7.2.2 NasCards::~NasCards () [inline]

Definition at line 40 of file NasCards.H.

6.7.3 Member Function Documentation

6.7.3.1 int NasCards::BeamCount (const BDF_Card * *c*, int *idx*) const

6.7.3.2 const char* NasCards::BeginChar () const [inline]

Definition at line 80 of file NasCards.H.

Referenced by FindBdf().

6.7.3.3 Int NasCards::BeginIndex () const [inline]

Definition at line 78 of file NasCards.H.

Referenced by CountCards(), and NotUsed().

6.7.3.4 const BDF_Card* NasCards::Card (int *i*) const [inline]

Definition at line 61 of file NasCards.H.

Referenced by CountCards(), and NotUsed().

6.7.3.5 int NasCards::count_rbe_blks (int *npids*, long * *mpcoff*, int * *blkmap*, int & *numrelem*)

6.7.3.6 void NasCards::CountCards ()

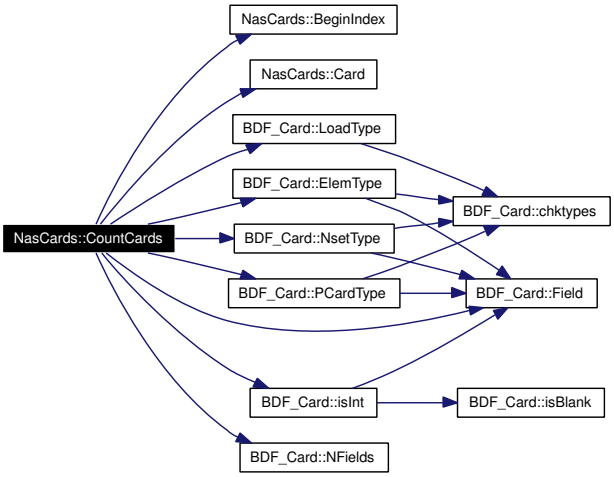
counts BDF cards only

Definition at line 189 of file NasCards.C.

References BeginIndex(), Card(), CORD, BDF_Card::ElemType(), BDF_Card::Field(), BDF_Card::isInt(), BDF_Card::LoadType(), BDF_Card::NFields(), NOT_ELEM, BDF_Card::NsetType(), BDF_Card::PCardType(), RBAR, RBE1, RBE2, RBE3, RROD, and RTRPLT.

Referenced by NasCards().

Here is the call graph for this function:



6.7.3.7 int NasCards::CountDarea () const

count the number of DAREA cards

Definition at line 291 of file NasCards.C.

6.7.3.8 int NasCards::CountDISP () const

count the number of displacement type nodesets

Definition at line 263 of file NasCards.C.

6.7.3.9 int NasCards::CountElements () const

count the number of elements in nastran file

Definition at line 256 of file NasCards.C.

6.7.3.10 void NasCards::CountElemPid (int & *numelems*, int & *npids*)

6.7.3.11 int NasCards::CountForces () const

count the number of force type nodesets

Definition at line 270 of file NasCards.C.

6.7.3.12 int NasCards::CountLoads () const

count the number of load cards

Definition at line 305 of file NasCards.C.

6.7.3.13 int NasCards::CountMoments () const

count the number of moment type nodesets

Definition at line 277 of file NasCards.C.

6.7.3.14 int NasCards::CountNodes () const

count the number of nodes

Definition at line 249 of file NasCards.C.

6.7.3.15 int NasCards::CountPidCards () const

count the number of property cards

Definition at line 298 of file NasCards.C.

6.7.3.16 int NasCards::CountTemps () const

count the number of temperature type nodesets

Definition at line 284 of file NasCards.C.

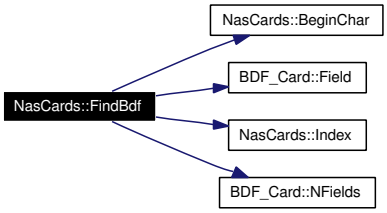
6.7.3.17 const BDF_Card * NasCards::FindBdf (Int & *idx*, const char * *str*, Int *id* = 0) const

find a card with matching field and ID in the BDF cards

Definition at line 453 of file NasCards.C.

References BeginChar(), BDF_Card::Field(), Index(), and BDF_Card::NFields().

Here is the call graph for this function:



6.7.3.18 const CaseCard * NasCards::FindCase (const char *, Int & *idx*) const

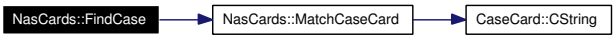
find a card in the case control section that begins with str.

Definition at line 391 of file NasCards.C.

References MatchCaseCard().

Referenced by FindCaseKeywordId().

Here is the call graph for this function:



6.7.3.19 bool NasCards::FindCaseKeywordId (const char * *keyword*, const char * *id*) const

find a keyword=id combination in case control.

Definition at line 423 of file NasCards.C.

References CaseCard::CString(), and FindCase().

Here is the call graph for this function:

6.7.3.20 const char* NasCards::FirstChar () const [inline]

Definition at line 74 of file NasCards.H.

6.7.3.21 const BDF_Card* NasCards::Index (const char * *p*) const [inline]

Definition at line 86 of file NasCards.H.

Referenced by FindBdf(), and NasCards().

6.7.3.22 bool NasCards::MatchCaseCard (Int *icase*, const char * *str*) const

check if the first nonblank case control card matches the string.

Definition at line 405 of file NasCards.C.

References CaseCard::CString().

Referenced by FindCase(), and NasCards().

Here is the call graph for this function:

6.7.3.23 int NasCards::NCards () const [inline]

Definition at line 60 of file NasCards.H.

6.7.3.24 void NasCards::NotUsed (const char *, int) const

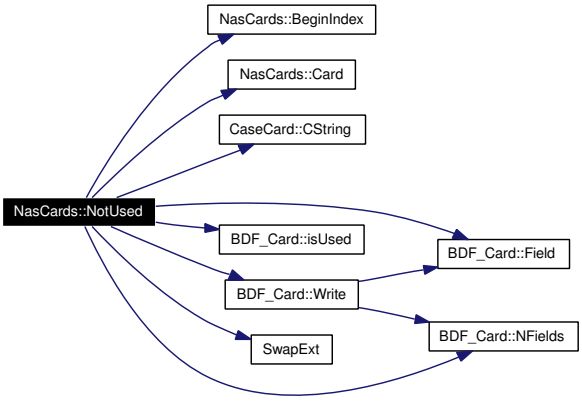
echo cards to a file

Definition at line 326 of file NasCards.C.

References BeginIndex(), Card(), CaseCard::CString(), BDF_Card::Field(), BDF_Card::isUsed(), BDF_-Card::NFields(), SwapExt(), and BDF_Card::Write().

Referenced by main().

Here is the call graph for this function:



6.7.3.25 void NasCards::ProcessElems (int *ex_id*, int *numelems*, int *npids*, int *nrpids*, int * *node_map*, int *nnodes*, int *numrelem*, const double *x*[ ], const double *y*[ ], const double *z*[ ], int * *blkcnt*, int * *blknodes*, int * *blkmap*, int * *blktyp*)

6.7.3.26 void NasCards::ProcessRelems (int *ex_id*, int *npids*, int *nrpids*, int * *blkmap*, int * *node_map*, int *nnodes*)

6.7.3.27 const char * NasCards::Title () const

return the title from nastran input

Definition at line 312 of file NasCards.C.

The documentation for this class was generated from the following files:

- NasCards.H
- NasCards.C

6.8 nodeset Class Reference

data and methods for a single exodus nodeset

`#include <nodeset.H>`

Public Member Functions

- nodeset (const NasCards *cards, const coords *input_coords)

  default nodeset constructor

- ~nodeset ()
- Int nnodesets () const

  return the number of nodesets

- const Int * sid () const
- const Int * num_nodes () const
- void convert_nodes (const coords *input_coords)

  converts nasgen node numbering to Exodus node numbering

- int num_disp () const
- int num_force () const
- int num_darea () const
- const Int * Disp_Sid () const
- const Int * Force_Sid () const
- const Int * Darea_Sid () const
- void enable_nodeset1 (Int grid0)

  used to enable nodeset 1, which is a substitue for nastran's ground

Friends

- class exodus_file

6.8.1 Detailed Description

data and methods for a single exodus nodeset

Definition at line 23 of file nodeset.H.

6.8.2 Constructor & Destructor Documentation

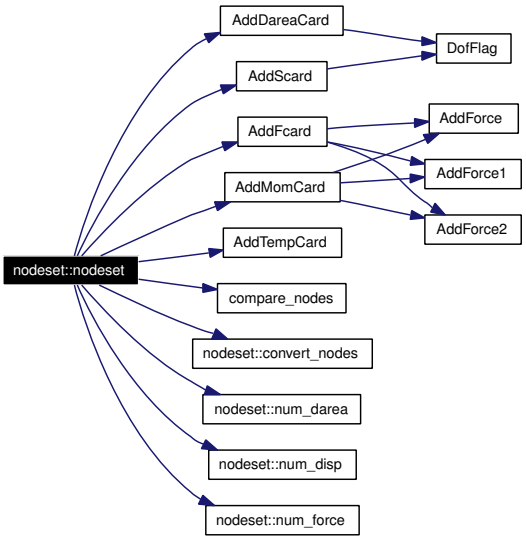
6.8.2.1 nodeset::nodeset (const NasCards * *cards*, const coords * *input_coords*)

default nodeset constructor

Definition at line 43 of file nodeset.C.

References AddDareaCard(), AddFcard(), AddMomCard(), AddScard(), AddTempCard(), compare_nodes(), convert_nodes(), nodeset_node::displacement, nodeset_node::node, num_darea(), num_disp(), num_force(), and nodeset_node::position.

Here is the call graph for this function:



6.8.2.2 nodeset::~nodeset () [inline]

Definition at line 111 of file nodeset.H.

6.8.3 Member Function Documentation

6.8.3.1 void nodeset::convert_nodes (const coords * *input_coords*)

converts nasgen node numbering to Exodus node numbering

Definition at line 450 of file nodeset.C.

Referenced by nodeset().

6.8.3.2 const Int* nodeset::Darea_Sid () const [inline]

Definition at line 135 of file nodeset.H.

6.8.3.3 const Int* nodeset::Disp_Sid () const [inline]

Definition at line 133 of file nodeset.H.

6.8.3.4 void nodeset::enable_nodeset1 (Int *grid0*)

used to enable nodeset 1, which is a substitue for nastran's ground

Definition at line 458 of file nodeset.C.

Referenced by main().

6.8.3.5 const Int* nodeset::Force_Sid () const [inline]

Definition at line 134 of file nodeset.H.

6.8.3.6 Int nodeset::nnodesets () const

return the number of nodesets

Definition at line 474 of file nodeset.C.

6.8.3.7 int nodeset::num_darea () const [inline]

Definition at line 132 of file nodeset.H.

Referenced by nodeset().

6.8.3.8 int nodeset::num_disp () const [inline]

Definition at line 130 of file nodeset.H.

Referenced by nodeset().

6.8.3.9 int nodeset::num_force () const [inline]

Definition at line 131 of file nodeset.H.

Referenced by nodeset().

6.8.3.10 const Int * nodeset::num_nodes () const

Definition at line 485 of file nodeset.C.

6.8.3.11 const Int * nodeset::sid () const

Definition at line 479 of file nodeset.C.

6.8.4 Friends And Related Function Documentation

6.8.4.1 friend class exodus_file [friend]

Definition at line 124 of file nodeset.H.

The documentation for this class was generated from the following files:

- nodeset.H
- nodeset.C

6.9 nodeset_node Class Reference

`#include <nodeset.H>`

Public Attributes

- Int node
- Int position
- double displacement

6.9.1 Member Data Documentation

6.9.1.1 double nodeset_node::displacement

Definition at line 19 of file nodeset.H.

Referenced by nodeset::nodeset().

6.9.1.2 Int nodeset_node::node

Definition at line 17 of file nodeset.H.

Referenced by compare_nodes(), and nodeset::nodeset().

6.9.1.3 Int nodeset_node::position

Definition at line 18 of file nodeset.H.

Referenced by compare_nodes(), and nodeset::nodeset().

The documentation for this class was generated from the following file:

- nodeset.H

6.10 pid_cards Class Reference

data and methods for property cards

`#include <pid_cards.H>`

Public Member Functions

- pid_cards (const NasCards *cards)
- ~pid_cards ()
- Int Num () const
- Int Pid (Int idx) const
- Int BdfCardIndex (Int idx) const

Classes

- class sortset

6.10.1 Detailed Description

data and methods for property cards

Definition at line 16 of file pid_cards.H.

6.10.2 Constructor & Destructor Documentation

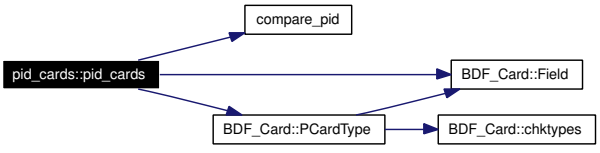
6.10.2.1 pid_cards::pid_cards (const NasCards * *cards*)

This class simply stores an array with pid card ids and a second array that corrisponds to the first that contains the NasCards card number of the pid card. This class gets around the use of a lookup table as used in coords as the pid_map array is never used externally, while in coords it is exported to Exodus

Definition at line 25 of file pid_cards.C.

References compare_pid(), BDF_Card::Field(), pid_cards::sortset::nasidx, BDF_Card::PCardType(), and pid_cards::sortset::pid.

Here is the call graph for this function:



6.10.2.2 pid_cards::~pid_cards () [inline]

Definition at line 32 of file pid_cards.H.

6.10.3 Member Function Documentation

6.10.3.1 Int pid_cards::BdfCardIndex (Int *idx*) const

Definition at line 112 of file pid_cards.C.

6.10.3.2 Int pid_cards::Num () const [inline]

Definition at line 37 of file pid_cards.H.

6.10.3.3 Int pid_cards::Pid (Int *idx*) const [inline]

Definition at line 40 of file pid_cards.H.

The documentation for this class was generated from the following files:

- pid_cards.H
- pid_cards.C

6.11 pid_cards::sortset Class Reference

`#include <pid_cards.H>`

Public Attributes

- Int pid
- Int nasidx

6.11.1 Member Data Documentation

6.11.1.1 Int pid_cards::sortset::nasidx

Definition at line 50 of file pid_cards.H.

Referenced by pid_cards::pid_cards().

6.11.1.2 Int pid_cards::sortset::pid

Definition at line 49 of file pid_cards.H.

Referenced by compare_pid(), and pid_cards::pid_cards().

The documentation for this class was generated from the following file:

- pid_cards.H

6.12 sideset Class Reference

data and methods for an exodus sideset

`#include <sideset.H>`

Public Member Functions

- sideset (const NasCards *cards, elem_block *eblock, const coords *Coords)
- int nsidesets () const
- const int * sideset_ids () const
- const int * num_sides () const
- const int * num_dist () const
- ~sideset ()
- int NumSets () const
- const int * SetIDs () const
- int AddPLoad1 (const BDF_Card *c, const elem_block *, int *ndist, const coords *Coords=0, int *sides=0, int *elem_ids=0, double *dist_fact=0)
- int AddPLoad2 (const BDF_Card *c, const elem_block *, int *ndist, int *sides=0, int *elem_ids=0, double *dist_fact=0)
- int AddPLoad4 (const BDF_Card *c, const elem_block *, int *ndist, int *sides=0, int *elem_ids=0, double *dist_fact=0)

Friends

- class exodus_file

6.12.1 Detailed Description

data and methods for an exodus sideset

Definition at line 16 of file sideset.H.

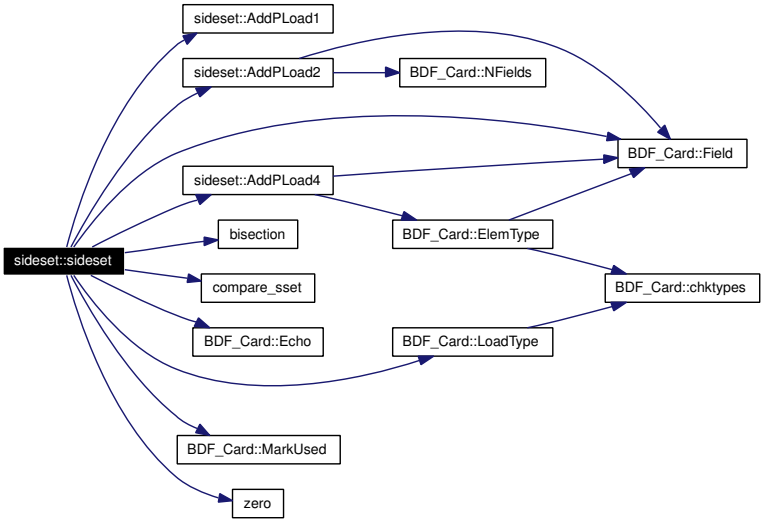
6.12.2 Constructor & Destructor Documentation

6.12.2.1 sideset::sideset (const NasCards * *cards*, elem_block * *eblock*, const coords * *Coords*)

Definition at line 20 of file sideset.C.

References AddPLoad1(), AddPLoad2(), AddPLoad4(), bisection(), compare_sset(), BDF_Card::Echo(), BDF_Card::Field(), BDF_Card::LoadType(), BDF_Card::MarkUsed(), and zero().

Here is the call graph for this function:



6.12.2.2 sideset::~sideset () [inline]

Definition at line 51 of file sideset.H.

6.12.3 Member Function Documentation

6.12.3.1 int sideset::AddPLoad1 (const BDF_Card * *c*, const elem_block * *blk*, int * *ndist*, const coords * *Coords* = 0, int * *sides* = 0, int * *elems* = 0, double * *dist_fact* = 0)

double NormX[3];

Definition at line 257 of file sideset.C.

Referenced by sideset().

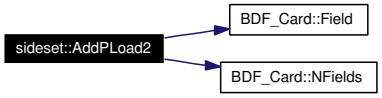
6.12.3.2 int sideset::AddPLoad2 (const BDF_Card * *c*, const elem_block *, int * *ndist*, int * *sides* = 0, int * *elem_ids* = 0, double * *dist_fact* = 0)

Definition at line 400 of file sideset.C.

References BDF_Card::Field(), and BDF_Card::NFields().

Referenced by sideset().

Here is the call graph for this function:



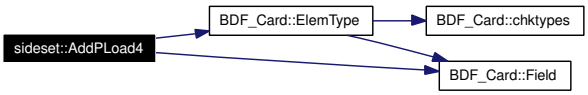
6.12.3.3 int sideset::AddPLoad4 (const BDF_Card * *c*, const elem_block *, int * *ndist*, int * *sides* = 0, int * *elem_ids* = 0, double * *dist_fact* = 0)

Definition at line 468 of file sideset.C.

References CHEX20, CHEX8, CPENTA15, CPENTA6, CQUAD4, CQUAD8, CTET10, CTET4, CTRIA3, CTRIA6, BDF_Card::ElemType(), and BDF_Card::Field().

Referenced by sideset().

Here is the call graph for this function:



6.12.3.4 int sideset::nsidesets () const

Definition at line 234 of file sideset.C.

6.12.3.5 const int * sideset::num_dist () const

Definition at line 249 of file sideset.C.

6.12.3.6 const int * sideset::num_sides () const

Definition at line 244 of file sideset.C.

6.12.3.7 int sideset::NumSets () const [inline]

Definition at line 64 of file sideset.H.

6.12.3.8 const int* sideset::SetIDs () const [inline]

Definition at line 65 of file sideset.H.

6.12.3.9 const int * sideset::sideset_ids () const

Definition at line 239 of file sideset.C.

6.12.4 Friends And Related Function Documentation

6.12.4.1 friend class exodus_file [friend]

Definition at line 49 of file sideset.H.

The documentation for this class was generated from the following files:

- sideset.H

- sideset.C

6.13 SuperElem Class Reference

data and methods for a single super element.

`#include <superelem.H>`

Public Member Functions

- SuperElem (const NasCards *Cards, const char *name, char ctype)
  construct a single, mega superelement from the bulk data
- ~SuperElem ()
- void Echo () const
- char CType () const
- Int Node (Int i) const
- short Cid (Int i) const
- Int NumNodes ()
  return the number of nodes in the superelement
- Int get_connectivity (Int *conn)
  set the nodal connectivity in 'conn'
- Int Dim () const
- const char * Name () const
  return the matrix name (or "unknown")
- Int WriteMatrixExo (int ex_id, const vector< long > &cmap) const

Classes

- struct ireal

6.13.1 Detailed Description

data and methods for a single super element.

Definition at line 30 of file superelem.H.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 SuperElem::SuperElem (const NasCards * *Cards*, const char * *name*, char *ctype*)

construct a single, mega superelement from the bulk data

Definition at line 12 of file superelem.C.

6.13.2.2 SuperElem::~SuperElem ()

Definition at line 39 of file superelem.C.

6.13.3 Member Function Documentation

6.13.3.1 short SuperElem::Cid (Int *i*) const [inline]

Definition at line 61 of file superelem.H.

Referenced by Echo().

6.13.3.2 char SuperElem::CType () const [inline]

Definition at line 58 of file superelem.H.

6.13.3.3 Int SuperElem::Dim () const [inline]

Definition at line 65 of file superelem.H.

Referenced by SuperList::WriteBlock().

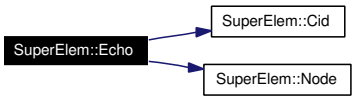
6.13.3.4 void SuperElem::Echo () const

Definition at line 174 of file superelem.C.

References Cid(), and Node().

Referenced by SuperList::SuperList().

Here is the call graph for this function:



6.13.3.5 Int SuperElem::get_connectivity (Int * *conn*)

set the nodal connectivity in 'conn'

Definition at line 267 of file superelem.C.

Referenced by SuperList::get_connectivity(), SuperList::NumNodes(), and NumNodes().

6.13.3.6 const char * SuperElem::Name () const

return the matrix name (or "unknown")

Definition at line 294 of file superelem.C.

Referenced by SuperList::WriteBlock().

6.13.3.7 Int SuperElem::Node (Int *i*) const [inline]

Definition at line 59 of file superelem.H.

Referenced by Echo().

6.13.3.8 Int SuperElem::NumNodes ()

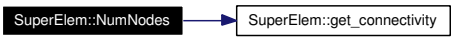
return the number of nodes in the superelement

Definition at line 258 of file superelem.C.

References get_connectivity().

Referenced by SuperList::NumNodes().

Here is the call graph for this function:



6.13.3.9 Int SuperElem::WriteMatrixExo (int *ex_id*, const vector< long > & *cmap*) const

Definition at line 306 of file superelem.C.

References zero().

Here is the call graph for this function:

The documentation for this class was generated from the following files:

- superelem.H
- superelem.C

6.14 SuperElem::ireal Struct Reference

`#include <superelem.H>`

Public Attributes

- Int i
- Real v

6.14.1 Member Data Documentation

6.14.1.1 Int SuperElem::ireal::i

Definition at line 74 of file superelem.H.

Referenced by irealcmp().

6.14.1.2 Real SuperElem::ireal::v

Definition at line 75 of file superelem.H.

The documentation for this struct was generated from the following file:

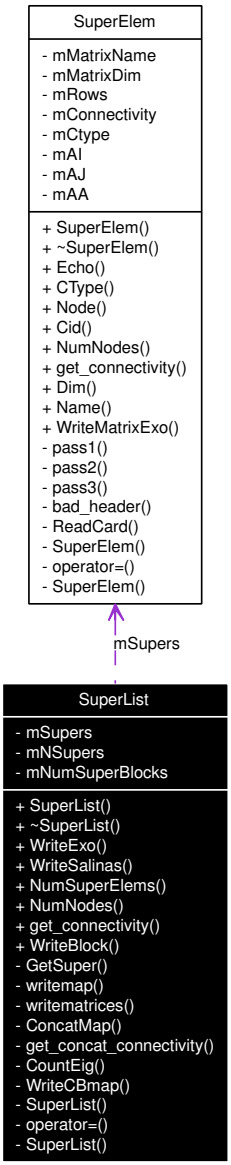
- superelem.H

6.15 SuperList Class Reference

data for a list of superelements

`#include <superlist.H>`

Collaboration diagram for SuperList:



Public Member Functions

- SuperList (const NasCards *Cards)

  construct the list of super elements

- ~SuperList ()
- void WriteExo ()

- void WriteSalinas ()
- Int NumSuperElems () const
- Int NumNodes (Int i) const

  returns the number of nodes in the connectivity for superelement block i

- Int get_connectivity (Int &blk_index, Int nodes_per_blk[ ], Int elems_per_blk[ ], Int ∗conn[ ]) const

  set the values of the elements per block, and the connectivity

- void WriteBlock (FILE ∗out, Int blkid_start, const coords ∗input_coord, const char ∗exoname) const

  write block data to a salinas file.

6.15.1 Detailed Description

data for a list of superelements

Definition at line 16 of file superlist.H.

6.15.2 Constructor & Destructor Documentation

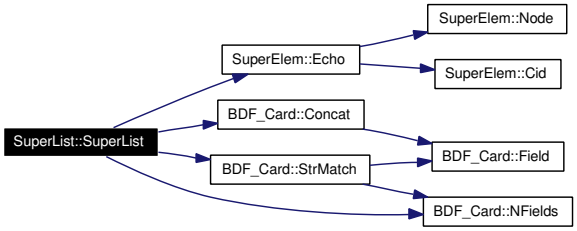
6.15.2.1 SuperList::SuperList (const NasCards ∗ *Cards*)

construct the list of super elements

Definition at line 24 of file superlist.C.

References BDF_Card::Concat(), SuperElem::Echo(), BDF_Card::NFields(), and BDF_Card::StrMatch().

Here is the call graph for this function:



6.15.2.2 SuperList::∼SuperList ()

Definition at line 88 of file superlist.C.

6.15.3 Member Function Documentation

6.15.3.1 Int SuperList::get_connectivity (Int & *blk_index*, Int *nodes_per_blk*[ ], Int *elems_per_blk*[ ], Int ∗ *conn*[ ]) const

set the values of the elements per block, and the connectivity

Definition at line 151 of file superlist.C.

References SuperElem::get_connectivity().

Here is the call graph for this function:

6.15.3.2 Int SuperList::NumNodes (Int *i*) const

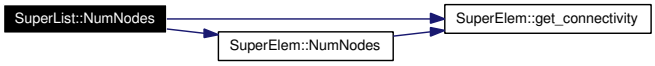
returns the number of nodes in the connectivity for superelement block i

Definition at line 102 of file superlist.C.

References SuperElem::get_connectivity(), and SuperElem::NumNodes().

Referenced by WriteBlock().

Here is the call graph for this function:



6.15.3.3 Int SuperList::NumSuperElems () const [inline]

Definition at line 29 of file superlist.H.

Referenced by elem_block::NSuper(), write_salinas::wrt_blks(), and write_presto::wrt_blks().

6.15.3.4 void SuperList::WriteBlock (FILE * *out*, Int *blkid_start*, const coords * *input_coord*, const char * *exoname*) const

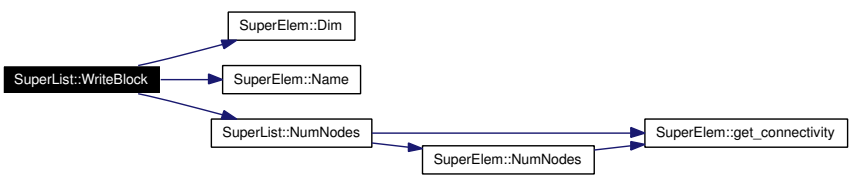
write block data to a salinas file.

Definition at line 175 of file superlist.C.

References SuperElem::Dim(), SuperElem::Name(), and NumNodes().

Referenced by write_salinas::wrt_blks(), and write_presto::wrt_blks().

Here is the call graph for this function:



6.15.3.5 void SuperList::WriteExo ()

6.15.3.6 void SuperList::WriteSalinas ()

The documentation for this class was generated from the following files:

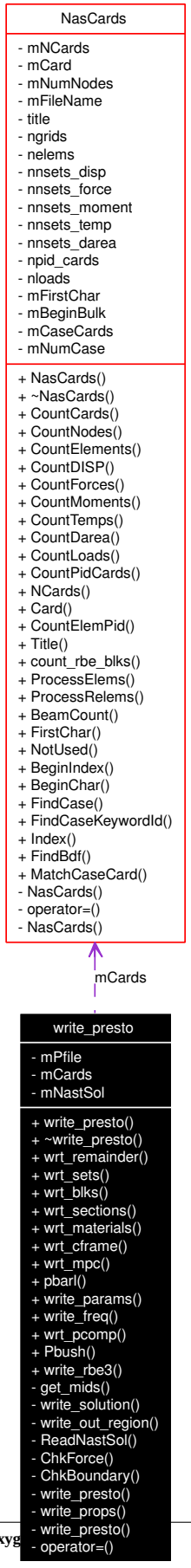
- superlist.H
- superlist.C

6.16 write_presto Class Reference

methods to write presto input file

`#include <write_presto.H>`

Collaboration diagram for write_presto:


NasCards
- mNCards
- mCard
- mNumNodes
- mFileName
- title
- ngrids
- nelems
- nnsets_disp
- nnsets_force
- nnsets_moment
- nnsets_temp
- nnsets_darea
- npid_cards
- nloads
- mFirstChar
- mBeginBulk
- mCaseCards
- mNumCase
+ NasCards()
+ ~NasCards()
+ CountCards()
+ CountNodes()
+ CountElements()
+ CountDISP()
+ CountForces()
+ CountMoments()
+ CountTemps()
+ CountDarea()
+ CountLoads()
+ CountPidCards()
+ NCards()
+ Card()
+ CountElemPid()
+ Title()
+ count_rbe_blks()
+ ProcessElems()
+ ProcessRelems()
+ BeamCount()
+ FirstChar()
+ NotUsed()
+ BeginIndex()
+ BeginChar()
+ FindCase()
+ FindCaseKeywordId()
+ Index()
+ FindBdf()
+ MatchCaseCard()
- NasCards()
- operator=()
- NasCards()
mCards
write_presto
- mPfile
- mCards
- mNastSol
+ write_presto()
+ ~write_presto()
+ wrt_remainder()
+ wrt_sets()
+ wrt_blks()
+ wrt_sections()
+ wrt_materials()
+ wrt_cframe()
+ wrt_mpc()
+ pbarl()
+ write_params()
+ write_freq()
+ wrt_pcomp()
+ Pbush()
+ write_rbe3()
- get_mids()
- write_solution()
- write_out_region()
- ReadNastSol()
- ChkForce()
- ChkBoundary()
- write_presto()
- write_props()
- write_presto()
- operator=()

Public Member Functions

- write_presto (const char *nasfile, const char *title, const char *exofile, const NasCards *Cards)

  the constructor

- ~write_presto ()
- void wrt_remainder (const nodeset *, const sideset *, const NasCards *) const
- void wrt_sets (const nodeset *, const sideset *, const NasCards *) const
- void wrt_blks (const NasCards *Cards, const pid_cards *pcard, const elem_block *blk, const coords *input_coords, const char *exofile) const
- void wrt_sections (const NasCards *Cards, const pid_cards *pcard, const elem_block *blk, const coords *input_coords, const char *exofile) const
- void wrt_materials (const NasCards *Cards) const
- void wrt_cframe (const coordframe *cframe) const
- int wrt_mpc (const NasCards *Cards, const coords *input_coord) const
- int pbarl (const BDF_Card *p) const
- void write_params (const NasCards *Cards) const
- void write_freq (const NasCards *Cards) const
- void wrt_pcomp (const BDF_Card *p) const

  write out most of the PCOMP generated layered information.

- int Pbush (const BDF_Card *p) const

  process a PBUSH card.

- void write_rbe3 (const elem_block *blk, int blk_index) const

  rbe3 are complicated animalis. This writes the data for an

6.16.1 Detailed Description

methods to write presto input file

Definition at line 23 of file write_presto.H.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 write_presto::write_presto (const char * *nasfile*, const char * *title*, const char * *exofile*, const NasCards * *Cards*)

the constructor

Definition at line 45 of file write_presto.C.

References SBSIZE, and SwapExt().

Here is the call graph for this function:

6.16.2.2 write_presto::~write_presto () [inline]

Definition at line 32 of file write_presto.H.

6.16.3 Member Function Documentation

6.16.3.1 int write_presto::pbarl (const BDF_Card * *p*) const

Definition at line 1045 of file write_presto.C.

References BDF_Card::Field(), and M_PI.

Referenced by wrt_sections().

Here is the call graph for this function:

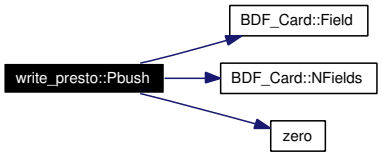
6.16.3.2 int write_presto::Pbush (const BDF_Card * *p*) const

process a PBUSH card.

Definition at line 1451 of file write_presto.C.

References BDF_Card::Field(), BDF_Card::NFields(), and zero().

Here is the call graph for this function:

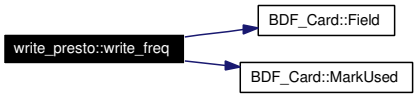


6.16.3.3 void write_presto::write_freq (const NasCards * *Cards*) const

Definition at line 1297 of file write_presto.C.

References BDF_Card::Field(), and BDF_Card::MarkUsed().

Here is the call graph for this function:

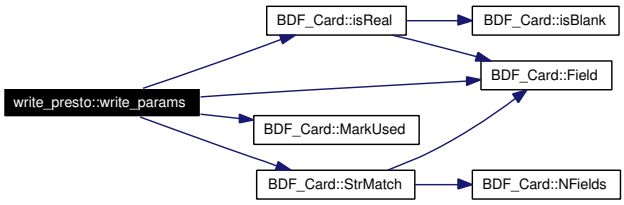


6.16.3.4 void write_presto::write_params (const NasCards * *Cards*) const

Definition at line 1249 of file write_presto.C.

References BDF_Card::Field(), BDF_Card::isReal(), BDF_Card::MarkUsed(), and BDF_Card::StrMatch().

Here is the call graph for this function:



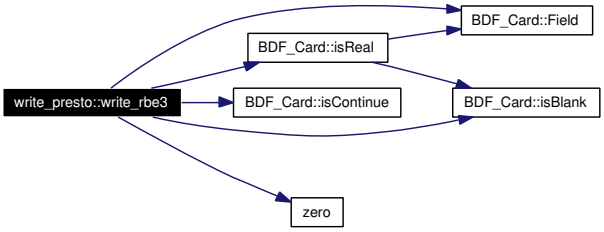
6.16.3.5 void write_presto::write_rbe3 (const elem_block * *blk*, int *blk_index*) const

rbe3 are complicated animalis. This writes the data for an

Definition at line 1555 of file write_presto.C.

References BDF_Card::Field(), BDF_Card::isBlank(), BDF_Card::isContinue(), BDF_Card::isReal(), and zero().

Here is the call graph for this function:



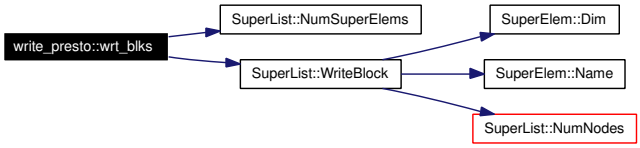
6.16.3.6 void write_presto::wrt_blks (const NasCards * *Cards*, const pid_cards * *pcard*, const elem_block * *blk*, const coords * *input_coords*, const char * *exofile*) const

Definition at line 439 of file write_presto.C.

References SuperList::NumSuperElems(), RBE3, and SuperList::WriteBlock().

Referenced by main().

Here is the call graph for this function:

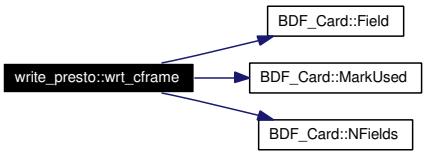


6.16.3.7 void write_presto::wrt_cframe (const coordframe * *cframe*) const

Definition at line 897 of file write_presto.C.

References CORD_CYL, CORD_RECT, CORD_SPH, BDF_Card::Field(), BDF_Card::MarkUsed(), and BDF_Card::NFields().

Here is the call graph for this function:



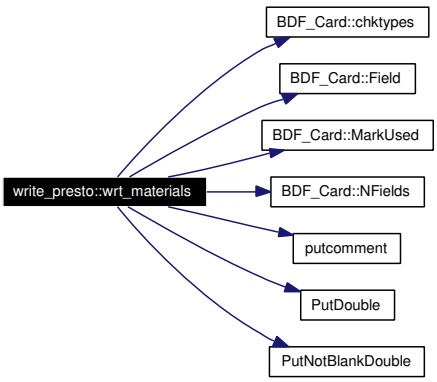
6.16.3.8 void write_presto::wrt_materials (const NasCards * *Cards*) const

Definition at line 665 of file write_presto.C.

References BDF_Card::chktypes(), BDF_Card::Field(), BDF_Card::MarkUsed(), BDF_Card::NFields(), NUM_MAT_TYPES, putcomment(), PutDouble(), and PutNotBlankDouble().

Referenced by main().

Here is the call graph for this function:

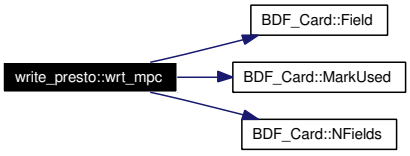


6.16.3.9 int write_presto::wrt_mpc (const NasCards * *Cards*, const coords * *input_coord*) const

Definition at line 957 of file write_presto.C.

References BDF_Card::Field(), BDF_Card::MarkUsed(), and BDF_Card::NFields().

Here is the call graph for this function:



6.16.3.10 void write_presto::wrt_pcomp (const BDF_Card * *p*) const

write out most of the PCOMP generated layered information.

Definition at line 1386 of file write_presto.C.

References BDF_Card::Field().

Here is the call graph for this function:

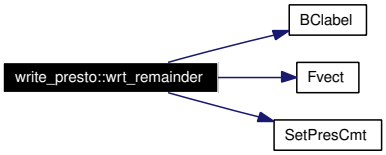
6.16.3.11 void write_presto::wrt_remainder (const nodeset *, const sideset *, const NasCards *) const

Definition at line 133 of file write_presto.C.

References BClabel(), Fvect(), and SetPresCmt().

Referenced by main().

Here is the call graph for this function:



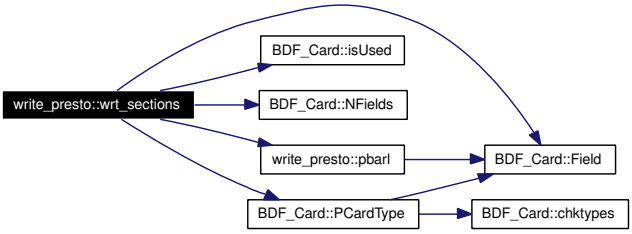
6.16.3.12 void write_presto::wrt_sections (const NasCards * *Cards*, const pid_cards * *pcard*, const elem_block * *blk*, const coords * *input_coords*, const char * *exofile*) const

Definition at line 510 of file write_presto.C.

References CELAS, CONM, BDF_Card::Field(), BDF_Card::isUsed(), BDF_Card::NFields(), pbarl(), BDF_Card::PCardType(), and RBE3.

Referenced by main().

Here is the call graph for this function:

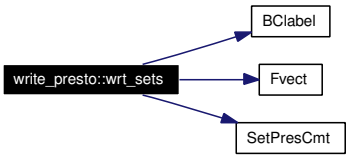


6.16.3.13 void write_presto::wrt_sets (const nodeset *, const sideset *, const NasCards *) const

Definition at line 253 of file write_presto.C.

References BClabel(), Fvect(), and SetPresCmt().

Here is the call graph for this function:



The documentation for this class was generated from the following files:

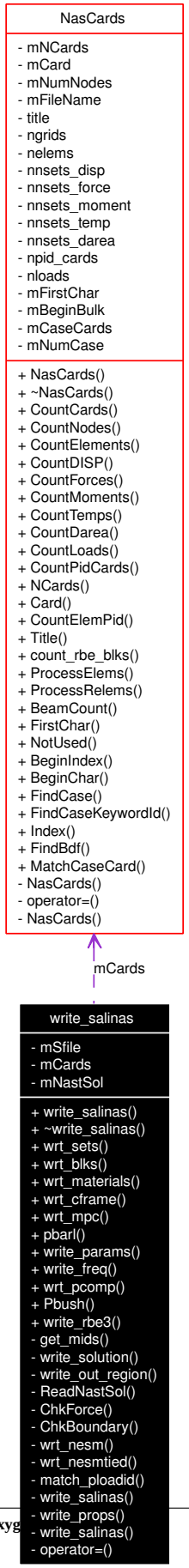
- write_presto.H
- write_presto.C

6.17 write_salinas Class Reference

methods to write the Salinas ∗.inp file

`#include <write_salinas.H>`

Collaboration diagram for write_salinas:


NasCards
- mNCards
- mCard
- mNumNodes
- mFileName
- title
- ngrids
- nelems
- nnsets_disp
- nnsets_force
- nnsets_moment
- nnsets_temp
- nnsets_darea
- npid_cards
- nloads
- mFirstChar
- mBeginBulk
- mCaseCards
- mNumCase
+ NasCards()
+ ~NasCards()
+ CountCards()
+ CountNodes()
+ CountElements()
+ CountDISP()
+ CountForces()
+ CountMoments()
+ CountTemps()
+ CountDarea()
+ CountLoads()
+ CountPidCards()
+ NCards()
+ Card()
+ CountElemPid()
+ Title()
+ count_rbe_blks()
+ ProcessElems()
+ ProcessRelems()
+ BeamCount()
+ FirstChar()
+ NotUsed()
+ BeginIndex()
+ BeginChar()
+ FindCase()
+ FindCaseKeywordId()
+ Index()
+ FindBdf()
+ MatchCaseCard()
- NasCards()
- operator=()
- NasCards()
mCards
write_salinas
- mSfile
- mCards
- mNastSol
+ write_salinas()
+ ~write_salinas()
+ wrt_sets()
+ wrt_blks()
+ wrt_materials()
+ wrt_cframe()
+ wrt_mpc()
+ pbarl()
+ write_params()
+ write_freq()
+ wrt_pcomp()
+ Pbush()
+ write_rbe3()
- get_mids()
- write_solution()
- write_out_region()
- ReadNastSol()
- ChkForce()
- ChkBoundary()
- wrt_nesm()
- wrt_nesmtied()
- match_ploadid()
- write_salinas()
- write_props()
- write_salinas()
- operator=()

Public Member Functions

- write_salinas (const char *nasfile, const char *title, const char *exofile, const NasCards *Cards)

  the constructor

- ~write_salinas ()
- void wrt_sets (const nodeset *, const sideset *, const NasCards *) const
- void wrt_blks (const NasCards *Cards, const pid_cards *pcard, const elem_block *blk, const coords *input_coords, const char *exofile) const
- void wrt_materials (const NasCards *Cards) const
- void wrt_cframe (const coordframe *cframe) const
- int wrt_mpc (const NasCards *Cards, const coords *input_coord) const
- int pbarl (const BDF_Card *p) const
- void write_params (const NasCards *Cards) const
- void write_freq (const NasCards *Cards) const
- void wrt_pcomp (const BDF_Card *p) const

  write out most of the PCOMP generated layered information.

- int Pbush (const BDF_Card *p) const

  process a PBUSH card.

- void write_rbe3 (const elem_block *blk, int blk_index) const

  rbe3 are complicated animalis. This writes the data for an

6.17.1 Detailed Description

methods to write the Salinas *.inp file

Definition at line 23 of file write_salinas.H.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 write_salinas::write_salinas (const char * *nasfile*, const char * *title*, const char * *exofile*, const NasCards * *Cards*)

the constructor

Definition at line 41 of file write_salinas.C.

References SBSIZE, and SwapExt().

Here is the call graph for this function:

6.17.2.2 write_salinas::~write_salinas () [inline]

Definition at line 32 of file write_salinas.H.

6.17.3 Member Function Documentation

6.17.3.1 int write_salinas::pbarl (const BDF_Card * *p*) const

Definition at line 900 of file write_salinas.C.

References BDF_Card::Field(), and M_PI.

Here is the call graph for this function:

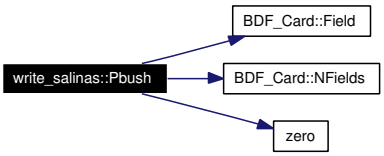
6.17.3.2 int write_salinas::Pbush (const BDF_Card * *p*) const

process a PBUSH card.

Definition at line 1260 of file write_salinas.C.

References BDF_Card::Field(), BDF_Card::NFields(), and zero().

Here is the call graph for this function:



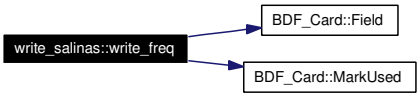
6.17.3.3 void write_salinas::write_freq (const NasCards * *Cards*) const

Definition at line 1106 of file write_salinas.C.

References BDF_Card::Field(), and BDF_Card::MarkUsed().

Referenced by main().

Here is the call graph for this function:



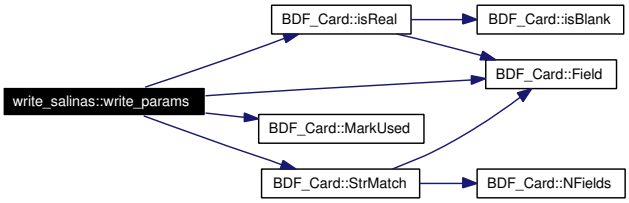
6.17.3.4 void write_salinas::write_params (const NasCards * *Cards*) const

Definition at line 1057 of file write_salinas.C.

References BDF_Card::Field(), BDF_Card::isReal(), BDF_Card::MarkUsed(), and BDF_Card::StrMatch().

Referenced by main().

Here is the call graph for this function:



6.17.3.5 void write_salinas::write_rbe3 (const elem_block * *blk*, int *blk_index*) const

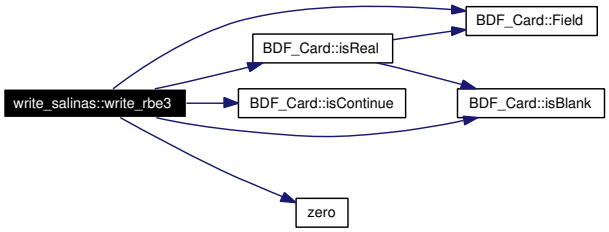
rbe3 are complicated animalis. This writes the data for an

Definition at line 1364 of file write_salinas.C.

References BDF_Card::Field(), BDF_Card::isBlank(), BDF_Card::isContinue(), BDF_Card::isReal(), and zero().

Referenced by wrt_blks().

Here is the call graph for this function:



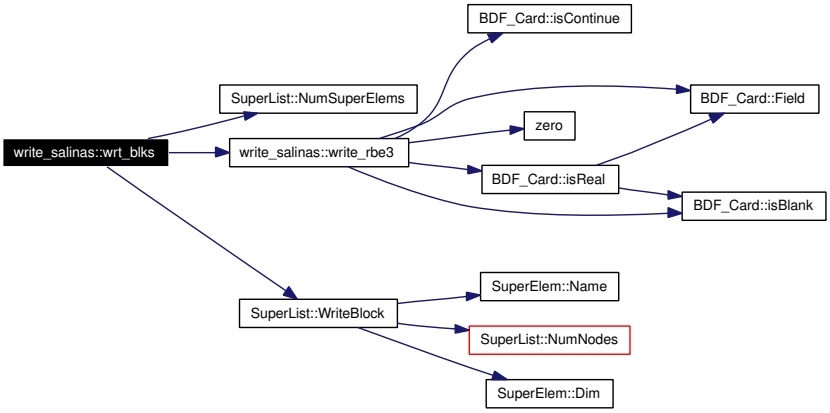
6.17.3.6 void write_salinas::wrt_blks (const NasCards * *Cards*, const pid_cards * *pcard*, const elem_block * *blk*, const coords * *input_coords*, const char * *exofile*) const

Definition at line 347 of file write_salinas.C.

References SuperList::NumSuperElems(), RBE3, write_rbe3(), and SuperList::WriteBlock().

Referenced by main().

Here is the call graph for this function:



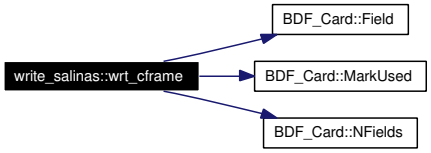
6.17.3.7 void write_salinas::wrt_cframe (const coordframe * *cframe*) const

Definition at line 752 of file write_salinas.C.

References CORD_CYL, CORD_RECT, CORD_SPH, BDF_Card::Field(), BDF_Card::MarkUsed(), and BDF_Card::NFields().

Referenced by main().

Here is the call graph for this function:



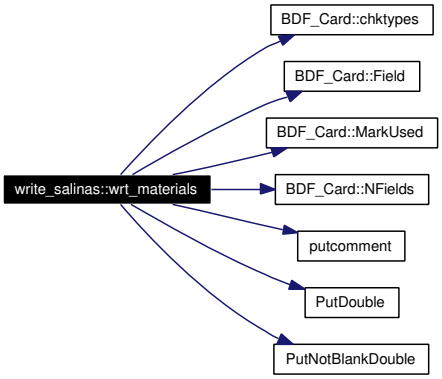
6.17.3.8 void write_salinas::wrt_materials (const NasCards * *Cards*) const

Definition at line 422 of file write_salinas.C.

References BDF_Card::chktypes(), BDF_Card::Field(), BDF_Card::MarkUsed(), BDF_Card::NFields(), NUM_MAT_TYPES, putcomment(), PutDouble(), and PutNotBlankDouble().

Referenced by main().

Here is the call graph for this function:



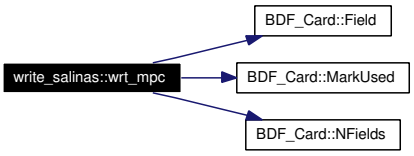
6.17.3.9 int write_salinas::wrt_mpc (const NasCards * *Cards*, const coords * *input_coord*) const

Definition at line 812 of file write_salinas.C.

References BDF_Card::Field(), BDF_Card::MarkUsed(), and BDF_Card::NFields().

Referenced by main().

Here is the call graph for this function:



6.17.3.10 void write_salinas::wrt_pcomp (const BDF_Card * *p*) const

write out most of the PCOMP generated layered information.

Definition at line 1195 of file write_salinas.C.

References BDF_Card::Field().

Here is the call graph for this function:

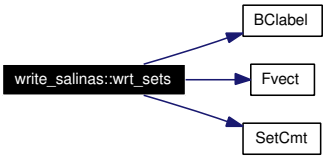
6.17.3.11 void write_salinas::wrt_sets (const nodeset *, const sideset *, const NasCards *) const

Definition at line 122 of file write_salinas.C.

References BClabel(), Fvect(), and SetCmt().

Referenced by main().

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- write_salinas.H
- write_salinas.C

Chapter 7

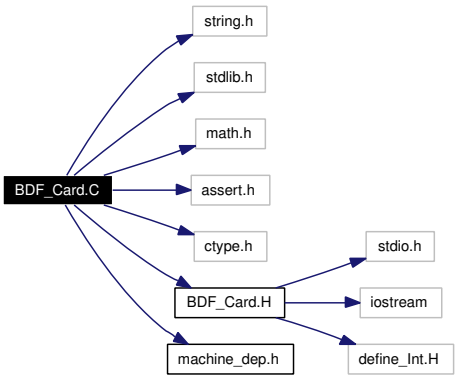
Nasgen File Documentation

7.1 BDF_Card.C File Reference

In memory representation of a nastran BDF card and logical continuation.

```
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <ctype.h>
#include "BDF_Card.H"
#include "machine_dep.h"
```

Include dependency graph for BDF_Card.C:



Defines

- #define BSZ 133
- #define NELEMTYPES 21
- #define NDLOADTYPES 6
- #define PCARDTYPES 13

- #define TLEN 7
- #define NNSETTYPES 7
- #define TLEN 5

Functions

- void AddField (const char *, int)

  adds data to temporary card

- double mypow (double in, int exp)

  integer exponentiation

- void strip_dollar ()

  inline function to remove nastran comment '$' and following text

- void chktab (char buf2[ ])

  check for tabs in nastran input. abort if found.

- int getlinetobuff (const char *str, long int *p)

  copy characters from a part of str, into buff

- int getlinetobuff (FILE *in)

  read a line of data from the file into buffer. strip comments.

- void rewindline (FILE *in, long oldloc, const char *str, long int *pstr)

  rewind file to old location

- int ChkContComma (const char *temp, FILE *in, const char *str, long int *pstr)

  check for contuation of line using commas.

Variables

- static char buff [BSZ+1]
- static const char * blank8 = " "
- static char tmpcard [CARD_LIST_SIZE][9]
- static long prevloc
- static long newloc = 0

7.1.1 Detailed Description

In memory representation of a nastran BDF card and logical continuation.

Definition in file BDF_Card.C.

7.1.2 Define Documentation

7.1.2.1 #define BSZ 133

Definition at line 19 of file BDF_Card.C.

Referenced by chktab(), getlinetobuff(), and strip_dollar().

7.1.2.2 #define NDLOADTYPES 6

Referenced by BDF_Card::LoadType().

7.1.2.3 #define NELEMTYPES 21

Referenced by BDF_Card::ElemType().

7.1.2.4 #define NNSETTYPES 7

Referenced by BDF_Card::NsetType().

7.1.2.5 #define PCARDTYPES 13

Referenced by BDF_Card::PCardType(), and PropHasMat().

7.1.2.6 #define TLEN 5

7.1.2.7 #define TLEN 7

Referenced by BDF_Card::NsetType(), and BDF_Card::PCardType().

7.1.3 Function Documentation

7.1.3.1 void AddField (const char *, int)

adds data to temporary card

Definition at line 441 of file BDF_Card.C.

References CARD_LIST_SIZE, and tmpcard.

Referenced by BDF_Card::BDF_Card().

7.1.3.2 int ChkContComma (const char * *temp*, FILE * *in*, const char * *str*, long int * *pstr*) [inline]

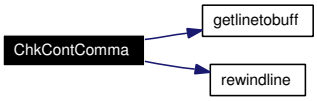
check for contuation of line using commas.

Definition at line 136 of file BDF_Card.C.

References blank8, buff, getlinetobuff(), newloc, and rewindline().

Referenced by BDF_Card::BDF_Card().

Here is the call graph for this function:



7.1.3.3 void chktab (char *buf2*[ ])

check for tabs in nastran input. abort if found.

Definition at line 41 of file BDF_Card.C.

References BSZ.

Referenced by getlinetobuff().

7.1.3.4 int getlinetobuff (FILE * *in*)

read a line of data from the file into buffer. strip comments.

Definition at line 105 of file BDF_Card.C.

7.1.3.5 int getlinetobuff (const char * *str*, long int * *p*)

copy characters from a part of str, into buff

Definition at line 80 of file BDF_Card.C.

Referenced by BDF_Card::BDF_Card(), ChkContComma(), and getlinetobuff().

7.1.3.6 double mypow (double *in*, int *exp*)

integer exponentiation

Definition at line 499 of file BDF_Card.C.

Referenced by BDF_Card::Field().

7.1.3.7 void rewindline (FILE * *in*, long *oldloc*, const char * *str*, long int * *pstr*)

rewind file to old location

Definition at line 126 of file BDF_Card.C.

References prevloc, and SEEK_SET.

Referenced by BDF_Card::BDF_Card(), and ChkContComma().

7.1.3.8 void strip_dollar () [inline]

inline function to remove nastran comment '$' and following text

Definition at line 33 of file BDF_Card.C.

References BSZ, and buff.

Referenced by getlinetobuff().

7.1.4 Variable Documentation

7.1.4.1 const char* blank8 = " " [static]

Definition at line 21 of file BDF_Card.C.

Referenced by BDF_Card::BDF_Card(), ChkContComma(), and BDF_Card::Write().

7.1.4.2 char buff[BSZ+1] [static]

Definition at line 20 of file BDF_Card.C.

Referenced by BDF_Card::BDF_Card(), ChkContComma(), getlinetobuff(), and strip_dollar().

7.1.4.3 long newloc = 0 [static]

Definition at line 27 of file BDF_Card.C.

Referenced by ChkContComma(), and getlinetobuff().

7.1.4.4 long prevloc [static]

Definition at line 26 of file BDF_Card.C.

Referenced by BDF_Card::BDF_Card(), getlinetobuff(), and rewindline().

7.1.4.5 char tmpcard[CARD_LIST_SIZE][9] [static]

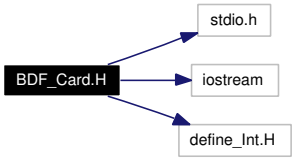
Definition at line 25 of file BDF_Card.C.

Referenced by AddField(), and BDF_Card::BDF_Card().

7.2 BDF_Card.H File Reference

```
#include <stdio.h>
#include <iostream>
#include "define_Int.H"
```

Include dependency graph for BDF_Card.H:



Namespaces

- namespace std

Defines

- #define CARD_LIST_SIZE 15000
- #define MAX_ETYPE 20

Enumerations

- enum EType {
  NOT_ELEM = -1, CBAR = 0, CBEAM = 1, CDAMP = 2,
  CELAS = 3, CGAP = 4, CHEX8 = 5, CHEX20 = 6,
  CONM = 7, CORD = 8, CPENTA6 = 9, CPENTA15 = 10,
  CROD = 11, CQUAD4 = 12, CQUAD8 = 13, CTET4 = 14,
  CTET10 = 15, CTRIA3 = 16, CTRIA6 = 17, CBUSH = 18,
  CVISC = 19, RBAR = 20, RBE1, RBE2,
  RBE3, RROD, RTRPLT, SUPER_ELEMENT }

Functions

- int getlinetobuff (FILE *in)
  read a line of data from the file into buffer. strip comments.
- int getlinetobuff (const char *str, long int *p)
  copy characters from a part of str, into buff
- int getlinetobuff (FILE *in, const char *str, long *p)

7.2.1 Define Documentation

7.2.1.1 #define CARD_LIST_SIZE 15000

Definition at line 9 of file BDF_Card.H.

Referenced by AddField(), and BDF_Card::BDF_Card().

7.2.1.2 #define MAX_ETYPE 20

Definition at line 44 of file BDF_Card.H.

Referenced by elem_block::elem_block().

7.2.2 Enumeration Type Documentation

7.2.2.1 enum EType

Enumeration values:

NOT_ELEM
CBAR
CBEAM
CDAMP
CELAS
CGAP
CHEX8
CHEX20
CONM
CORD
CPENTA6
CPENTA15
CROD
CQUAD4
CQUAD8
CTET4
CTET10
CTRIA3
CTRIA6
CBUSH
CVISC
RBAR
RBE1
RBE2
RBE3
RROD
RTRPLT
SUPER_ELEMENT

Definition at line 11 of file BDF_Card.H.

7.2.3 Function Documentation

7.2.3.1 int getlinetobuff (FILE * *in*, const char * *str*, long * *p*) [inline]

Definition at line 118 of file BDF_Card.H.

References getlinetobuff().

Here is the call graph for this function:

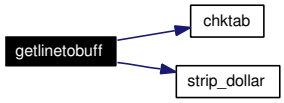
7.2.3.2 int getlinetobuff (const char * *str*, long int * *p*)

copy characters from a part of str, into buff

Definition at line 80 of file BDF_Card.C.

References BSZ, buff, chktab(), newloc, prevloc, and strip_dollar().

Here is the call graph for this function:



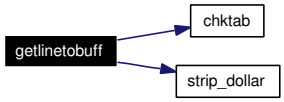
7.2.3.3 int getlinetobuff (FILE * *in*)

read a line of data from the file into buffer. strip comments.

Definition at line 105 of file BDF_Card.C.

References BSZ, buff, chktab(), newloc, prevloc, and strip_dollar().

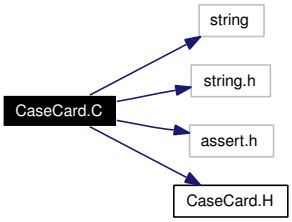
Here is the call graph for this function:



7.3 CaseCard.C File Reference

```
#include <string>
#include <string.h>
#include <assert.h>
#include "CaseCard.H"
```

Include dependency graph for CaseCard.C:



Defines

- #define COMMA ’,’

7.3.1 Define Documentation

7.3.1.1 #define COMMA ’,’

7.4 CaseCard.H File Reference

a container to store a single case control card as a long string

7.4.1 Detailed Description

a container to store a single case control card as a long string

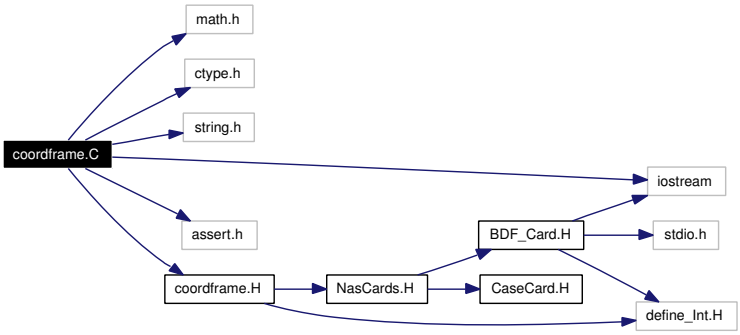
This class provides a container to store the data for a single case control card. This card is logically a long string. We store it that way.

Definition in file CaseCard.H.

7.5 coordframe.C File Reference

```
#include <math.h>
#include <ctype.h>
#include <string.h>
#include <iostream>
#include <assert.h>
#include "coordframe.H"
```

Include dependency graph for coordframe.C:



Defines

- #define M_PI 3.14159265358979

Variables

- static int xcid = -1

7.5.1 Define Documentation

7.5.1.1 #define M_PI 3.14159265358979

Definition at line 14 of file coordframe.C.

Referenced by write_salinas::pbarl(), write_presto::pbarl(), and coordframe::TransformToBasic().

7.5.2 Variable Documentation

7.5.2.1 int xcid = -1 [static]

Definition at line 17 of file coordframe.C.

Referenced by coordframe::Flatten(), coordframe::GetBasisVectors(), coordframe::GetTransMatrix(), coordframe::Norm3(), and coordframe::TransformToBasic().

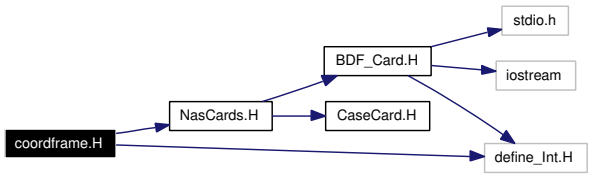
7.6 coordframe.H File Reference

A class to hold the array of all coordinate frames.

`#include "NasCards.H"`

`#include "define_Int.H"`

Include dependency graph for coordframe.H:



Enumerations

- enum Cord_Type {
  CORD_INVALID = -1, CORD_RECT = 0, CORD_CYL = 1, CORD_SPH = 2,
  CORD_GENERAL = 3 }

7.6.1 Detailed Description

A class to hold the array of all coordinate frames.

Definition in file coordframe.H.

7.6.2 Enumeration Type Documentation

7.6.2.1 enum Cord_Type

Enumeration values:

CORD_INVALID

CORD_RECT

CORD_CYL

CORD_SPH

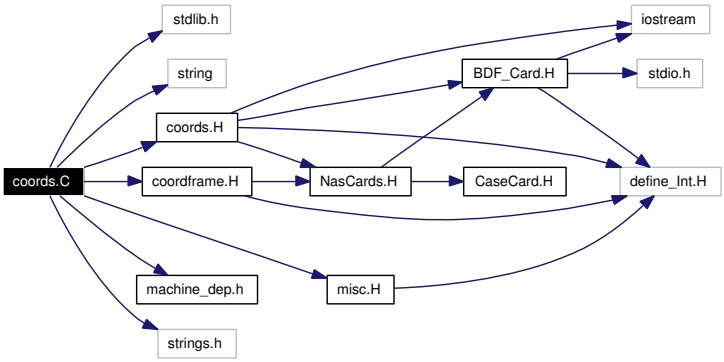
CORD_GENERAL

Definition at line 10 of file coordframe.H.

7.7 coords.C File Reference

```
#include <stdlib.h>
#include <string>
#include "coords.H"
#include "coordframe.H"
#include "machine_dep.h"
#include "misc.H"
#include <strings.h>
```

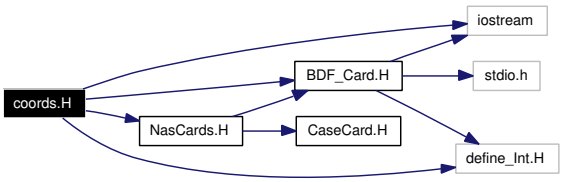
Include dependency graph for coords.C:



7.8 coords.H File Reference

```
#include <iostream>
#include "NasCards.H"
#include "BDF_Card.H"
#include "define_Int.H"
```

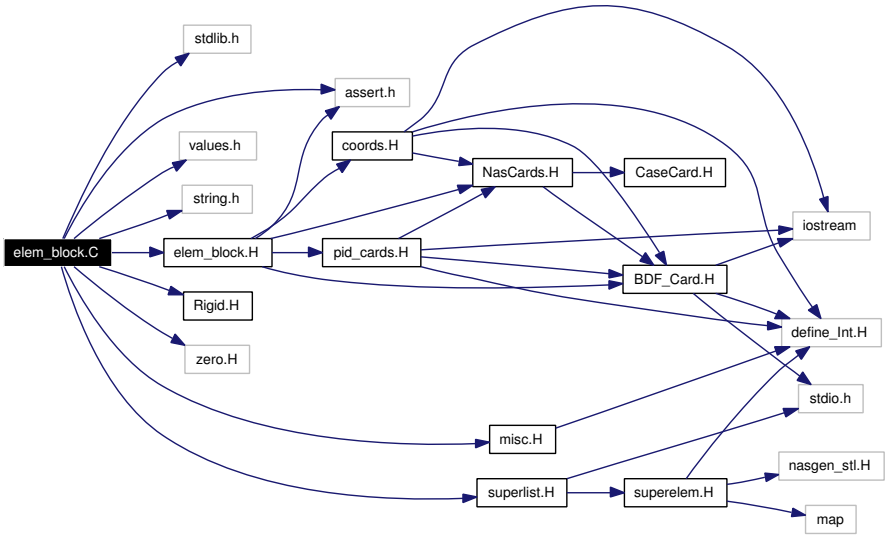
Include dependency graph for coords.H:



7.9 elem_block.C File Reference

```
#include <stdlib.h>
#include <assert.h>
#include <values.h>
#include <string.h>
#include "elem_block.H"
#include "Rigid.H"
#include "zero.H"
#include "misc.H"
#include "superlist.H"
```

Include dependency graph for elem_block.C:



Defines

- #define MAXINT 2147483648

Functions

- void zero (double *x, int n)
- int compare_pid_etype (const void *p1, const void *p2)

  comparison sort routine for pid and etype

- Int max (const Int *a, Int n)

Variables

- Int element_nnodes_lookup [MAX_ETYPE]

- Int nproperties_lookup [MAX_ETYPE]
- int noffset_attr_lookup [MAX_ETYPE]
- int use_short_names = 1
- static const char * exoname [MAX_ETYPE]
- static const char * exoname2 [MAX_ETYPE]
- static Int pid_convert_lookup [MAX_ETYPE]

7.9.1 Define Documentation

7.9.1.1 #define MAXINT 2147483648

Definition at line 25 of file elem_block.C.

7.9.2 Function Documentation

7.9.2.1 int compare_pid_etype (const void * *p1*, const void * *p2*)

comparison sort routine for pid and etype

Definition at line 72 of file elem_block.C.

7.9.2.2 Int max (const Int * *a*, Int *n*) [inline]

Definition at line 89 of file elem_block.C.

7.9.2.3 void zero (double * *x*, int *n*) [inline]

Definition at line 44 of file elem_block.C.

Referenced by elem_block::elem_block(), write_salinas::Pbush(), write_presto::Pbush(), sideset::sideset(), write_salinas::write_rbe3(), write_presto::write_rbe3(), and SuperElem::WriteMatrixExo().

7.9.3 Variable Documentation

7.9.3.1 Int element_nnodes_lookup[MAX_ETYPE]

Initial value:

```
{ 2, 2, 2, 2, 2 , 8, 20, 1, 0,
                        6, 15, 2, 4, 8, 4, 10, 3, 6, 2, 2}
```

Definition at line 29 of file elem_block.C.

Referenced by elem_block::elem_block().

7.9.3.2 const char* exoname[MAX_ETYPE] [static]

Initial value:

```
{ "BEAM", "BEAM", "BEAM", "TRUSS-SPR", "TRUSS-GAP",
  "HEX8", "HEX20", "SPHERE-MASS", "CORD", "WEDGE6",
  "WEDGE15", "TRUSS","SHELL4", "SHELL8","TETRA",
  "TETRA", "TRIANGLE", "TRIANGLE", "TRUSS-BUS", "TRUSS-VIS"
}
```

Definition at line 49 of file elem_block.C.

Referenced by elem_block::elem_block().

7.9.3.3 const char* exoname2[MAX_ETYPE] [static]

Initial value:

```
{ "BEAM", "BEAM", "BEAM", "TRUSS", "TRUSS",
  "HEX", "HEX", "SPHERE-MASS", "CORD", "WEDGE",
  "WEDGE", "TRUSS","SHELL", "SHELL","TETRA",
  "TETRA", "TRIANGLE", "TRIANGLE", "TRUSS", "TRUSS"
}
```

Definition at line 56 of file elem_block.C.

Referenced by elem_block::elem_block().

7.9.3.4 int noffset_attr_lookup[MAX_ETYPE]

Initial value:

```
{ 3, 3, 0, 0, 0,
                                        0, 0, 0, 0, 0,
                                        0, 0, 1, 1, 0,
                                        0, 1, 1, 0, 0}
```

Definition at line 38 of file elem_block.C.

Referenced by elem_block::elem_block().

7.9.3.5 Int nproperties_lookup[MAX_ETYPE]

Initial value:

```
{ 7, 7, 0, 3, 0,
                                      0, 0, 10, 0, 0,
                                      0, 1, 1, 1, 0,
                                      0, 1, 1, 0, 0}
```

Definition at line 33 of file elem_block.C.

Referenced by elem_block::elem_block().

7.9.3.6 Int pid_convert_lookup[MAX_ETYPE] [static]

Initial value:

```
{ 0, 0, 0, 6, 0,
                                                0, 1, 7, 0, 2,
                                                3, 2, 3, 0, 4,
                                                5, 0, 1, 0, 0}
```

Definition at line 65 of file elem_block.C.

7.9.3.7 int use_short_names = 1

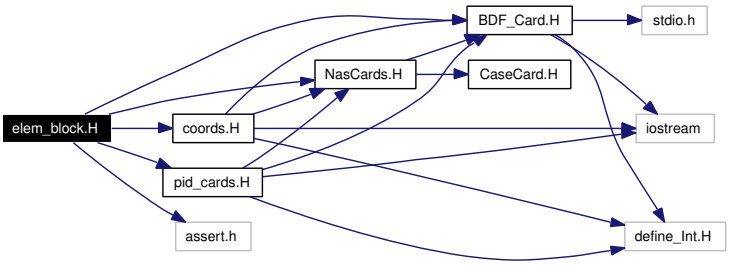
Definition at line 48 of file elem_block.C.

Referenced by elem_block::elem_block().

7.10 elem_block.H File Reference

```
#include "NasCards.H"
#include "BDF_Card.H"
#include "coords.H"
#include "pid_cards.H"
#include <assert.h>
```

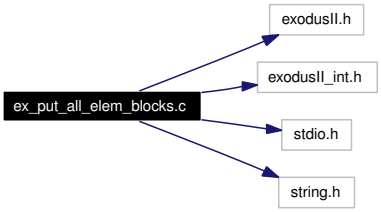
Include dependency graph for elem_block.H:



7.11 ex_put_all_elem_blocks.c File Reference

```
#include "exodusII.h"
#include "exodusII_int.h"
#include <stdio.h>
#include <string.h>
```

Include dependency graph for ex_put_all_elem_blocks.c:



Defines

- #define VOID_PTR (void∗)
- #define PRINT_ERROR (void)printf

Functions

- int ex_put_all_elem_blocks (int exo_id, int nblks, const int ∗blockid, const char ∗const ∗elem_type, const int ∗num_elem_in_blk, const int ∗num_nodes_per_element, const int ∗num_attributes)

7.11.1 Define Documentation

7.11.1.1 #define PRINT_ERROR (void)printf

Definition at line 10 of file ex_put_all_elem_blocks.c.

Referenced by ex_put_all_elem_blocks().

7.11.1.2 #define VOID_PTR (void∗)

Definition at line 7 of file ex_put_all_elem_blocks.c.

7.11.2 Function Documentation

7.11.2.1 int ex_put_all_elem_blocks (int *exo_id*, int *nblks*, const int ∗ *blockid*, const char ∗const ∗ *elem_type*, const int ∗ *num_elem_in_blk*, const int ∗ *num_nodes_per_element*, const int ∗ *num_attributes*)

Definition at line 12 of file ex_put_all_elem_blocks.c.

References PRINT_ERROR.

7.12 ex_put_all_elem_blocks.h File Reference

Functions

- int ex_put_all_elem_blocks (int exo_id, int nblks, const int *blockid, const char *const *elem_type, const int *num_elem_in_blk, const int *num_nodes_per_element, const int *num_attributes)

7.12.1 Function Documentation

7.12.1.1 int ex_put_all_elem_blocks (int *exo_id*, int *nblks*, const int * *blockid*, const char *const * *elem_type*, const int * *num_elem_in_blk*, const int * *num_nodes_per_element*, const int * *num_attributes*)

Definition at line 12 of file ex_put_all_elem_blocks.c.

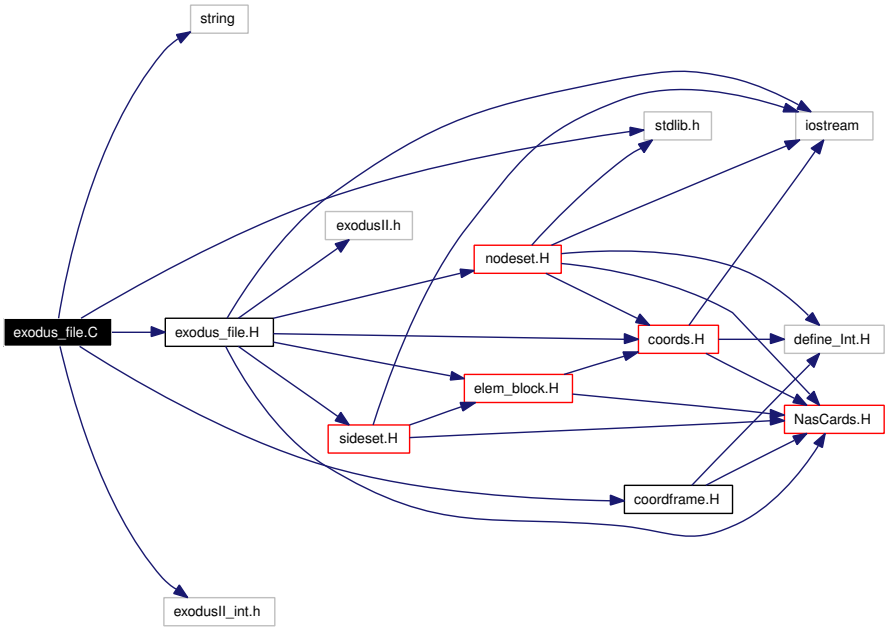
References PRINT_ERROR.

7.13 exodus_file.C File Reference

a wrapper around calls to the Exodus file API

```
#include <string>
#include <stdlib.h>
#include "exodus_file.H"
#include "coordframe.H"
#include "exodusII_int.h"
```

Include dependency graph for exodus_file.C:



7.13.1 Detailed Description

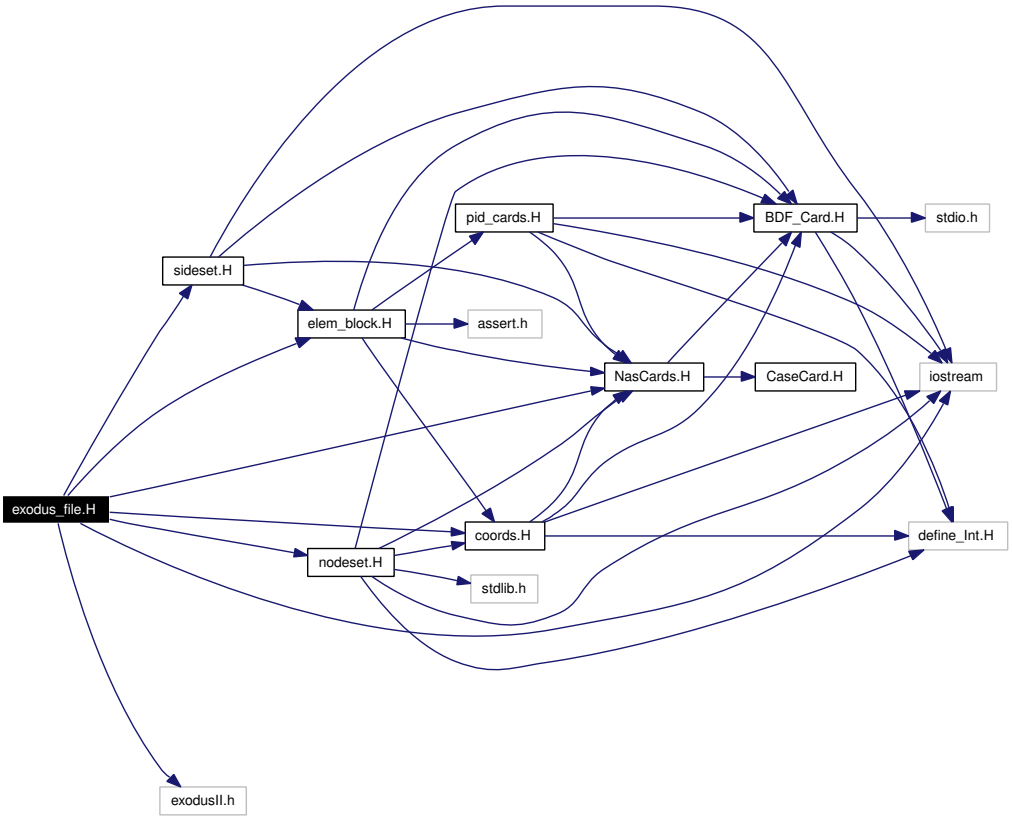
a wrapper around calls to the Exodus file API

Definition in file exodus_file.C.

7.14 exodus_file.H File Reference

```
#include <iostream>
#include "exodusII.h"
#include "NasCards.H"
#include "coords.H"
#include "elem_block.H"
#include "nodeset.H"
#include "sideset.H"
```

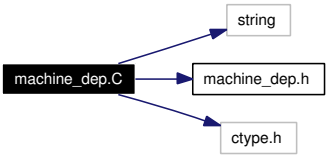
Include dependency graph for exodus_file.H:



7.15 machine_dep.C File Reference

```
#include <string>
#include "machine_dep.h"
#include <ctype.h>
```

Include dependency graph for machine_dep.C:



7.16 machine_dep.h File Reference

Defines

- #define SEEK_SET 0
- #define SEEK_CUR 1
- #define SEEK_END 2

7.16.1 Define Documentation

7.16.1.1 #define SEEK_CUR 1

Definition at line 133 of file machine_dep.h.

7.16.1.2 #define SEEK_END 2

Definition at line 137 of file machine_dep.h.

7.16.1.3 #define SEEK_SET 0

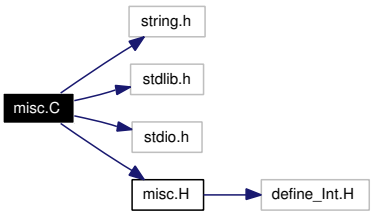
Definition at line 129 of file machine_dep.h.

Referenced by rewindline().

7.17 misc.C File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "misc.H"
```

Include dependency graph for misc.C:



Functions

- unsigned int DofFlag (const char *str)

  returns an integer representing the dofs active.

- int CountDof (unsigned int a)

  counts the number of dofs in a dof_flag

- int isSet (unsigned int a, int j)

  returns 1 if dof "j" is set in a dof_flag

- void SortAndEliminate (int &n, Int *ids)

  sorts an array of integers and eliminates redundancy

- Int Intcompare (const void *i, const void *j)

  comparison routine for qsort. taken from man page for qsort

- int bisection (int a, const int map[ ], int n)

  binary search, should be global

7.17.1 Function Documentation

7.17.1.1 int bisection (int *a*, const int *map*[ ], int *n*)

binary search, should be global

Definition at line 82 of file misc.C.

Referenced by coords::convert_connectivity(), and sideset::sideset().

7.17.1.2 int CountDof (unsigned int *a*)

counts the number of dofs in a dof_flag

Definition at line 28 of file misc.C.

7.17.1.3 unsigned int DofFlag (const char * *str*)

returns an integer representing the dofs active.

Definition at line 13 of file misc.C.

Referenced by AddDareaCard(), and AddScard().

7.17.1.4 Int Intcompare (const void * *i*, const void * *j*)

comparison routine for qsort. taken from man page for qsort

Definition at line 71 of file misc.C.

Referenced by coords::coords(), and SortAndEliminate().

7.17.1.5 int isSet (unsigned int *a*, int *j*)

returns 1 if dof "j" is set in a dof_flag

Definition at line 40 of file misc.C.

7.17.1.6 void SortAndEliminate (int & *n*, Int * *ids*)

sorts an array of integers and eliminates redundancy

Definition at line 49 of file misc.C.

References Intcompare().

Here is the call graph for this function:

7.18 misc.H File Reference

`#include "define_Int.H"`

Include dependency graph for misc.H:

Functions

- unsigned int DofFlag (const char *str)

  returns an integer representing the dofs active.

- int CountDof (unsigned int a)

  counts the number of dofs in a dof_flag

- int isSet (unsigned int a, int j)

  returns 1 if dof "j" is set in a dof_flag

- void SortAndEliminate (int &n, Int *ids)

  sorts an array of integers and eliminates redundancy

- int Intcompare (const void *i, const void *j)

  comparison routine for qsort. taken from man page for qsort

- int find_idx (int i1, const int *list, int len)
- int bisection (int a, const int map[ ], int n)

  binary search, should be global

7.18.1 Function Documentation

7.18.1.1 int bisection (int *a*, const int *map*[ ], int *n*)

binary search, should be global

Definition at line 82 of file misc.C.

Referenced by coords::convert_connectivity(), and sideset::sideset().

7.18.1.2 int CountDof (unsigned int *a*)

counts the number of dofs in a dof_flag

Definition at line 28 of file misc.C.

7.18.1.3 unsigned int DofFlag (const char * *str*)

returns an integer representing the dofs active.

Definition at line 13 of file misc.C.

Referenced by AddDareaCard(), and AddScard().

7.18.1.4 int find_idx (int *i1*, const int * *list*, int *len*) [inline]

Definition at line 21 of file misc.H.

Referenced by elem_block::GetSide().

7.18.1.5 int Intcompare (const void * *i*, const void * *j*)

comparison routine for qsort. taken from man page for qsort

Definition at line 71 of file misc.C.

Referenced by coords::coords(), and SortAndEliminate().

7.18.1.6 int isSet (unsigned int *a*, int *j*)

returns 1 if dof "j" is set in a dof_flag

Definition at line 40 of file misc.C.

7.18.1.7 void SortAndEliminate (int & *n*, Int * *ids*)

sorts an array of integers and eliminates redundancy

Definition at line 49 of file misc.C.

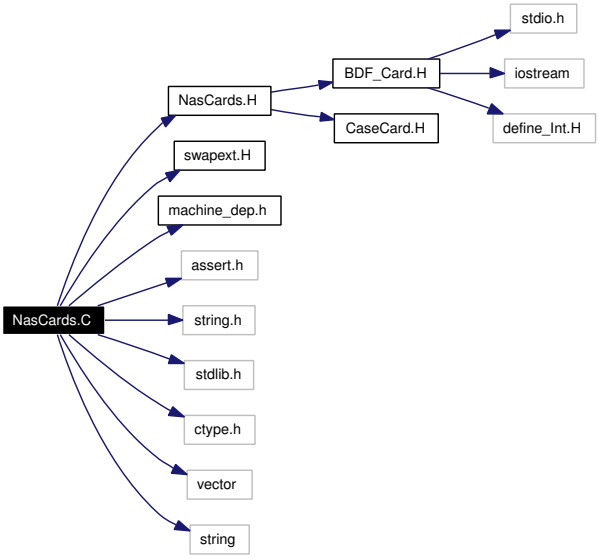
References Intcompare().

Here is the call graph for this function:

7.19 NasCards.C File Reference

```
#include "NasCards.H"
#include "swapext.H"
#include "machine_dep.h"
#include <assert.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <vector>
#include <string>
```

Include dependency graph for NasCards.C:



Defines

- #define BSIZE 511

Functions

- long int file2str (const char *filen, std::string &strv)

  transfer characters from file to a vector string

Variables

- static char buff [BSIZE+1]

7.19.1 Define Documentation

7.19.1.1 #define BSIZE 511

Definition at line 16 of file NasCards.C.

Referenced by file2str().

7.19.2 Function Documentation

7.19.2.1 long int file2str (const char * *filen*, std::string & *strv*)

transfer characters from file to a vector string

Definition at line 24 of file NasCards.C.

References BSIZE, and buff.

Referenced by NasCards::NasCards().

7.19.3 Variable Documentation

7.19.3.1 char buff[BSIZE+1] [static]

Definition at line 17 of file NasCards.C.

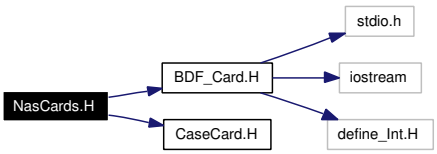
Referenced by file2str().

7.20 NasCards.H File Reference

A class to hold all the bulk data cards.

```
#include "BDF_Card.H"
#include "CaseCard.H"
```

Include dependency graph for NasCards.H:



7.20.1 Detailed Description

A class to hold all the bulk data cards.

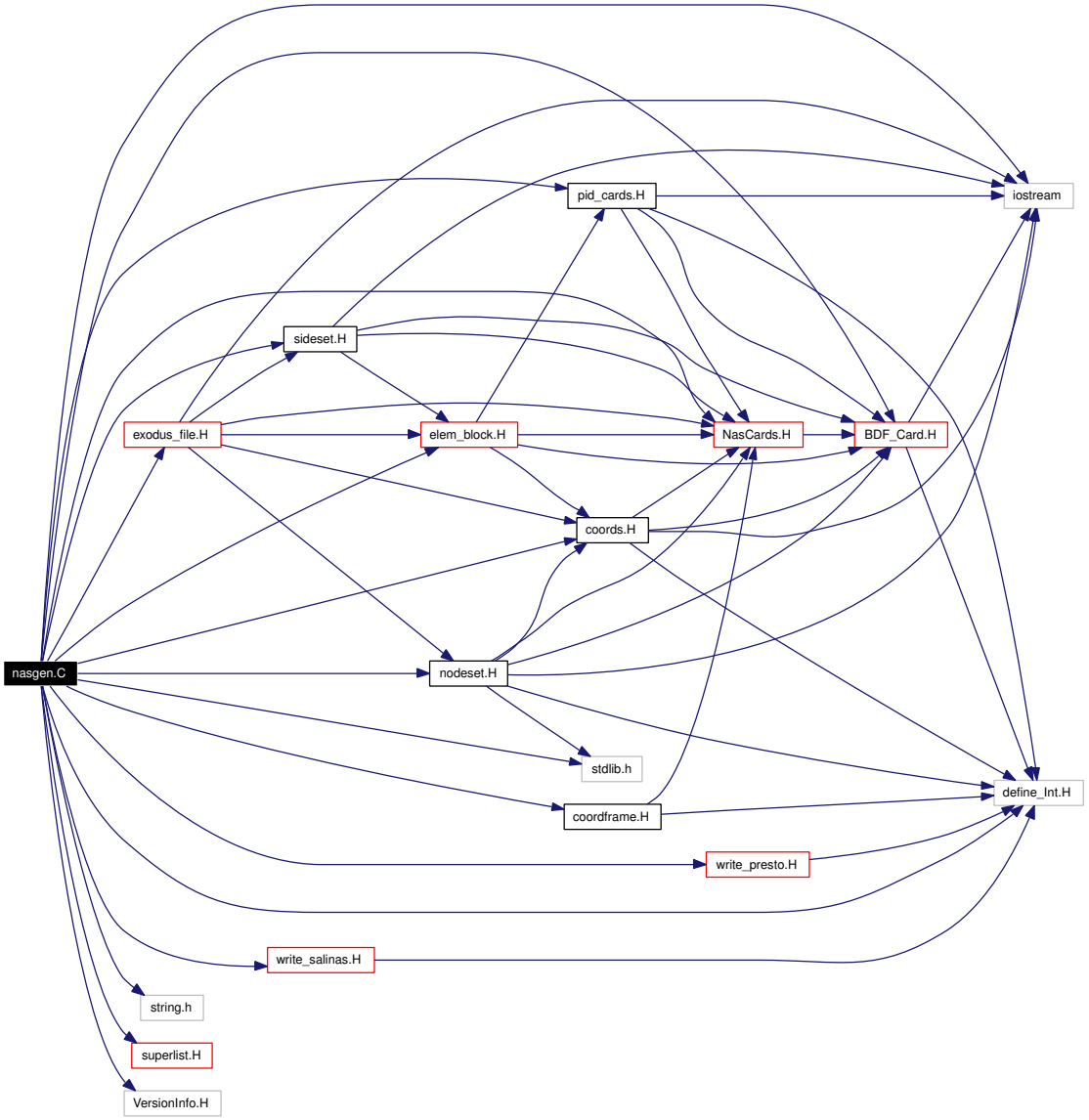
Definition in file NasCards.H.

7.21 nasgen.C File Reference

Converts a standard nastran file into exodus II format.

```
#include <iostream>
#include <string.h>
#include <stdlib.h>
#include "BDF_Card.H"
#include "NasCards.H"
#include "coords.H"
#include "exodus_file.H"
#include "elem_block.H"
#include "pid_cards.H"
#include "nodeset.H"
#include "sideset.H"
#include "define_Int.H"
#include "write_salinas.H"
#include "write_presto.H"
#include "coordframe.H"
#include "superlist.H"
#include "VersionInfo.H"
```

Include dependency graph for nasgen.C:



Functions

- void print_usage ()

  print nasgen usage

- int main (int argc, char ∗argv[ ])

Variables

- const char ∗ tag_str = NULL
- const char ∗ bld_date = SAL_BUILD_DATE
- const char ∗ bld_mach = SAL_BUILD_MACH
- const char ∗ bld_os = SAL_BUILD_OS1 " " SAL_BUILD_OS2

- const char * bld_cc = SAL_COMPILER

- const char * version_date = NULL

7.21.1 Detailed Description

Converts a standard nastran file into exodus II format.

Definition in file nasgen.C.

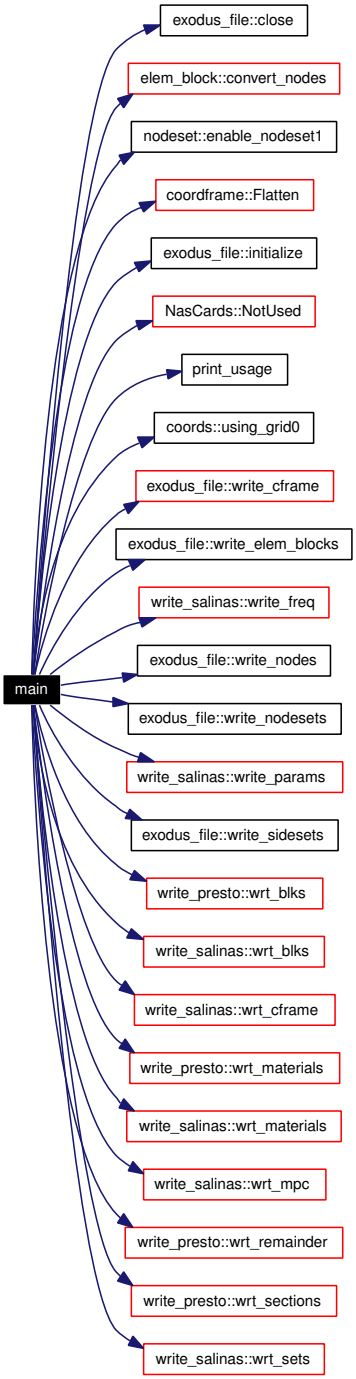
7.21.2 Function Documentation

7.21.2.1 int main (int *argc*, char * *argv*[ ])

Definition at line 112 of file nasgen.C.

References bld_cc, bld_date, bld_mach, bld_os, exodus_file::close(), elem_block::convert_nodes(), nodeset::enable_nodeset1(), coordframe::Flatten(), exodus_file::initialize(), NasCards::NotUsed(), print_usage(), tag_str, coords::using_grid0(), version_date, exodus_file::write_cframe(), exodus_file::write_elem_blocks(), write_salinas::write_freq(), exodus_file::write_nodes(), exodus_file::write_nodesets(), write_salinas::write_params(), exodus_file::write_sidesets(), write_presto::wrt_blks(), write_salinas::wrt_blks(), write_salinas::wrt_cframe(), write_presto::wrt_materials(), write_salinas::wrt_materials(), write_salinas::wrt_mpc(), write_presto::wrt_remainder(), write_presto::wrt_sections(), and write_salinas::wrt_sets().

Here is the call graph for this function:



7.21.2.2 void print_usage ()

print nasgen usage

Definition at line 277 of file nasgen.C.

Referenced by main().

7.21.3 Variable Documentation

7.21.3.1 const char* bld_cc = SAL_COMPILER

Definition at line 106 of file nasgen.C.

Referenced by main().

7.21.3.2 const char* bld_date = SAL_BUILD_DATE

Definition at line 103 of file nasgen.C.

Referenced by main().

7.21.3.3 const char* bld_mach = SAL_BUILD_MACH

Definition at line 104 of file nasgen.C.

Referenced by main().

7.21.3.4 const char* bld_os = SAL_BUILD_OS1 " " SAL_BUILD_OS2

Definition at line 105 of file nasgen.C.

Referenced by main().

7.21.3.5 const char* tag_str = NULL

Definition at line 102 of file nasgen.C.

Referenced by main().

7.21.3.6 const char* version_date = NULL

Definition at line 107 of file nasgen.C.

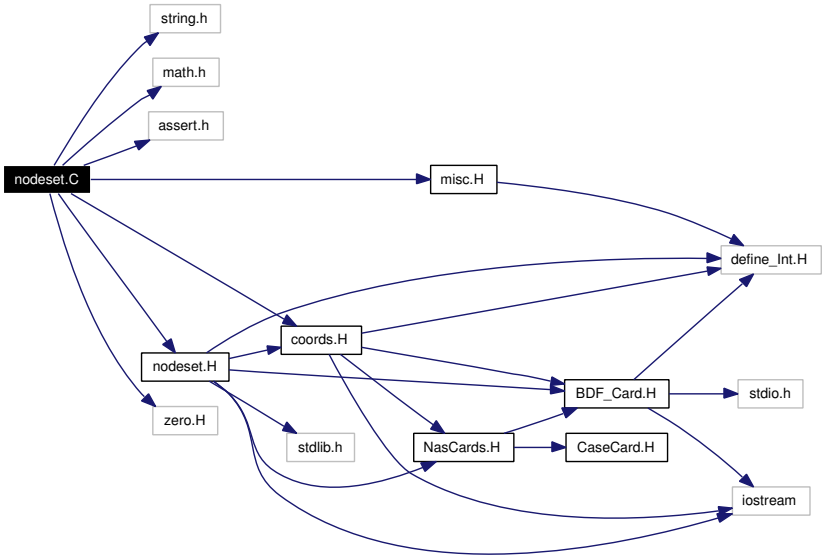
Referenced by main().

7.22 nodeset.C File Reference

All information about a single nodeset. May span multiple BDF.

```
#include <string.h>
#include <math.h>
#include <assert.h>
#include "coords.H"
#include "nodeset.H"
#include "misc.H"
#include "zero.H"
```

Include dependency graph for nodeset.C:



Typedefs

- typedef nodeset_node * nodeset_node_p

Functions

- Int compare_nodes (const void *i, const void *j)

7.22.1 Detailed Description

All information about a single nodeset. May span multiple BDF.

Definition in file nodeset.C.

7.22.2 Typedef Documentation

7.22.2.1 typedef nodeset_node* nodeset_node_p

Definition at line 28 of file nodeset.C.

7.22.3 Function Documentation

7.22.3.1 Int compare_nodes (const void * *i*, const void * *j*)

Definition at line 29 of file nodeset.C.

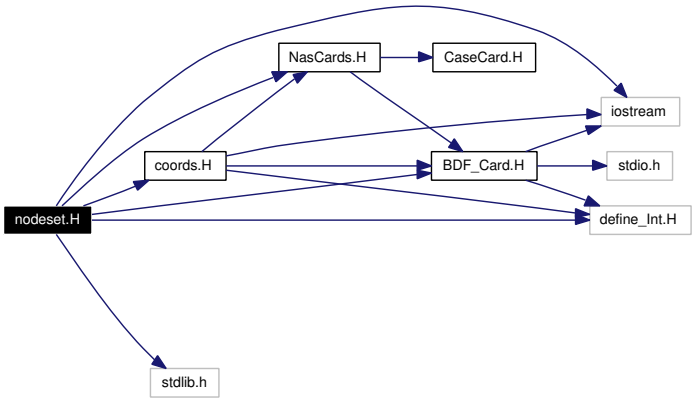
References nodeset_node::node, and nodeset_node::position.

Referenced by nodeset::nodeset().

7.23 nodeset.H File Reference

```
#include <iostream>
#include <stdlib.h>
#include "NasCards.H"
#include "BDF_Card.H"
#include "define_Int.H"
#include "coords.H"
```

Include dependency graph for nodeset.H:



Functions

- int compare_nset (const void *p1, const void *p2)
- int AddScard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddFcard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddMomCard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddTempCard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddDareaCard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)

7.23.1 Function Documentation

7.23.1.1 int AddDareaCard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 387 of file nodeset1.C.

References DofFlag().

Referenced by nodeset::nodeset().

Here is the call graph for this function:

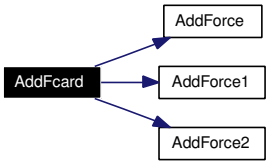
7.23.1.2 int AddFcard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 253 of file nodeset1.C.

References AddForce(), AddForce1(), and AddForce2().

Referenced by nodeset::nodeset().

Here is the call graph for this function:



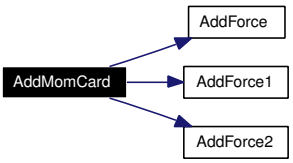
7.23.1.3 int AddMomCard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 289 of file nodeset1.C.

References AddForce(), AddForce1(), and AddForce2().

Referenced by nodeset::nodeset().

Here is the call graph for this function:



7.23.1.4 int AddScard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 28 of file nodeset1.C.

References DofFlag().

Referenced by nodeset::nodeset().

Here is the call graph for this function:

7.23.1.5 int AddTempCard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 330 of file nodeset1.C.

Referenced by nodeset::nodeset().

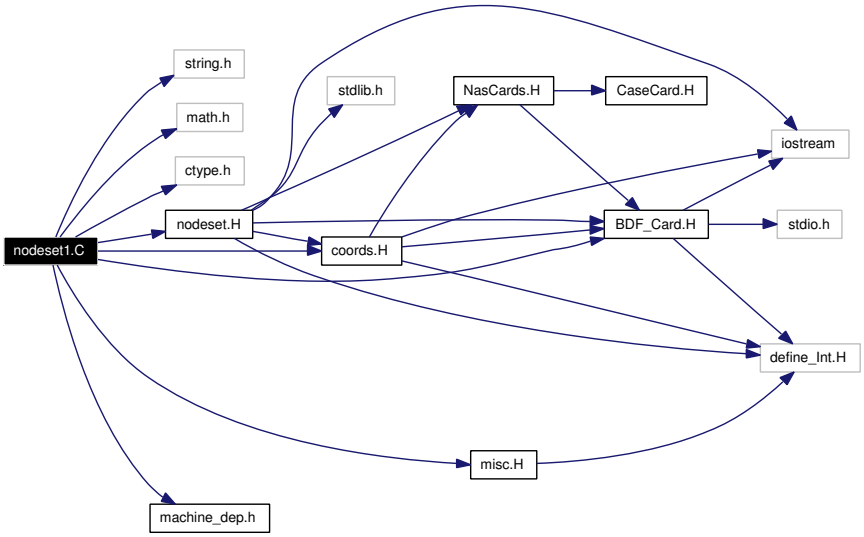
7.23.1.6 int compare_nset (const void * *p1*, const void * *p2*)

7.24 nodeset1.C File Reference

Nodeset information from a single BDF card.

```
#include <string.h>
#include <math.h>
#include <ctype.h>
#include "misc.H"
#include "nodeset.H"
#include "BDF_Card.H"
#include "coords.H"
#include "machine_dep.h"
```

Include dependency graph for nodeset1.C:



Functions

- int AddScard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddForce1 (const BDF_Card *c, const coords *Nodemap, unsigned int dof, int *nodelist, double *wtlist)
- int AddForce (const BDF_Card *c, const coords *Nodemap, unsigned int dof, int *nodelist, double *wtlist)
- int AddForce2 (const BDF_Card *c, const coords *Nodemap, unsigned int dof, int *nodelist, double *wtlist)
- int AddFcard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddMomCard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)

- int AddTempCard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)
- int AddDareaCard (const BDF_Card *c, int &sid, const coords *Nodemap, unsigned int &dof_flag, unsigned int dof, int *nodelist, double *wtlist)

7.24.1 Detailed Description

Nodeset information from a single BDF card.

Each of the following routines processes data from a single BDF card that generates a node set. They all have a similar calling sequence. They all return the number of nodes referenced in the card. They also modify an integer flag that indicates the number of active DOFS referenced. If nodelist is non-null the nodelist will be filled.

Definition in file nodeset1.C.

7.24.2 Function Documentation

7.24.2.1 int AddDareaCard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 387 of file nodeset1.C.

References DofFlag().

Referenced by nodeset::nodeset().

Here is the call graph for this function:

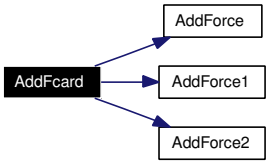
7.24.2.2 int AddFcard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 253 of file nodeset1.C.

References AddForce(), AddForce1(), and AddForce2().

Referenced by nodeset::nodeset().

Here is the call graph for this function:



7.24.2.3 int AddForce (const BDF_Card * *c*, const coords * *Nodemap*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 154 of file nodeset1.C.

Referenced by AddFcard(), and AddMomCard().

7.24.2.4 int AddForce1 (const BDF_Card * *c*, const coords * *Nodemap*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 109 of file nodeset1.C.

Referenced by AddFcard(), and AddMomCard().

7.24.2.5 int AddForce2 (const BDF_Card * *c*, const coords * *Nodemap*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 193 of file nodeset1.C.

Referenced by AddFcard(), and AddMomCard().

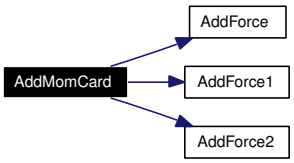
7.24.2.6 int AddMomCard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 289 of file nodeset1.C.

References AddForce(), AddForce1(), and AddForce2().

Referenced by nodeset::nodeset().

Here is the call graph for this function:



7.24.2.7 int AddScard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 28 of file nodeset1.C.

References DofFlag().

Referenced by nodeset::nodeset().

Here is the call graph for this function:

7.24.2.8 int AddTempCard (const BDF_Card * *c*, int & *sid*, const coords * *Nodemap*, unsigned int & *dof_flag*, unsigned int *dof*, int * *nodelist*, double * *wtlist*)

Definition at line 330 of file nodeset1.C.

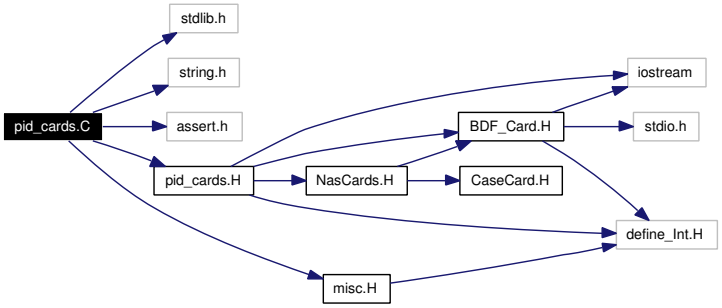
Referenced by nodeset::nodeset().

7.25 pid_cards.C File Reference

property information including nastran and exodus ids

```
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "pid_cards.H"
#include "misc.H"
```

Include dependency graph for pid_cards.C:



Defines

- #define PCARDTYPES 13

Functions

- int compare_pid (const void *p1, const void *p2)
- int PropHasMat (int ptyp)

7.25.1 Detailed Description

property information including nastran and exodus ids

Definition in file pid_cards.C.

7.25.2 Define Documentation

7.25.2.1 #define PCARDTYPES 13

7.25.3 Function Documentation

7.25.3.1 int compare_pid (const void * *p1*, const void * *p2*)

Definition at line 83 of file pid_cards.C.

References pid_cards::sortset::pid.

Referenced by pid_cards::pid_cards().

7.25.3.2 int PropHasMat (int *ptyp*)

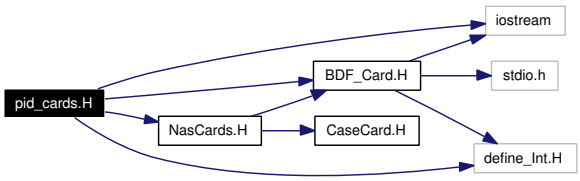
Definition at line 96 of file pid_cards.C.

References PCARDTYPES.

7.26 pid_cards.H File Reference

```
#include <iostream>
#include "NasCards.H"
#include "BDF_Card.H"
#include "define_Int.H"
```

Include dependency graph for pid_cards.H:



Functions

- int compare_pid (const void *p1, const void *p2)
- int PropHasMat (int ptyp)

7.26.1 Function Documentation

7.26.1.1 int compare_pid (const void * *p1*, const void * *p2*)

Definition at line 83 of file pid_cards.C.

References pid_cards::sortset::pid.

Referenced by pid_cards::pid_cards().

7.26.1.2 int PropHasMat (int *ptyp*)

Definition at line 96 of file pid_cards.C.

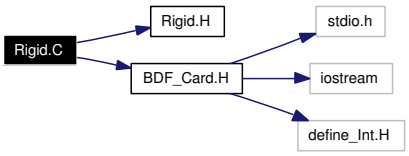
References PCARDTYPES.

7.27 Rigid.C File Reference

functions for handling rigid elements

```
#include "Rigid.H"
#include "BDF_Card.H"
```

Include dependency graph for Rigid.C:



Functions

- int CountLinks (const BDF_Card *c)
  counts the number of rigid links in a RBE card
- int AddLinks (const BDF_Card *c, int *conn)
  adds the nodes in the connectivity of a rigid link to the connectivity

7.27.1 Detailed Description

functions for handling rigid elements

Definition in file Rigid.C.

7.27.2 Function Documentation

7.27.2.1 int AddLinks (const BDF_Card * *c*, int * *conn*)

adds the nodes in the connectivity of a rigid link to the connectivity

Definition at line 67 of file Rigid.C.

References NOT_ELEM, RBAR, RBE1, RBE2, RBE3, RROD, and RTRPLT.

Referenced by elem_block::elem_block().

7.27.2.2 int CountLinks (const BDF_Card * *c*)

counts the number of rigid links in a RBE card

Definition at line 12 of file Rigid.C.

References NOT_ELEM, RBAR, RBE1, RBE2, RBE3, RROD, and RTRPLT.

7.28 Rigid.H File Reference

Functions

- int CountLinks (const BDF_Card ∗c)
  counts the number of rigid links in a RBE card
- int AddLinks (const BDF_Card ∗c, int ∗conn)
  adds the nodes in the connectivity of a rigid link to the connectivity

7.28.1 Function Documentation

7.28.1.1 int AddLinks (const BDF_Card ∗ *c*, int ∗ *conn*)

adds the nodes in the connectivity of a rigid link to the connectivity

Definition at line 67 of file Rigid.C.

References NOT_ELEM, RBAR, RBE1, RBE2, RBE3, RROD, and RTRPLT.

Referenced by elem_block::elem_block().

7.28.1.2 int CountLinks (const BDF_Card ∗ *c*)

counts the number of rigid links in a RBE card

Definition at line 12 of file Rigid.C.

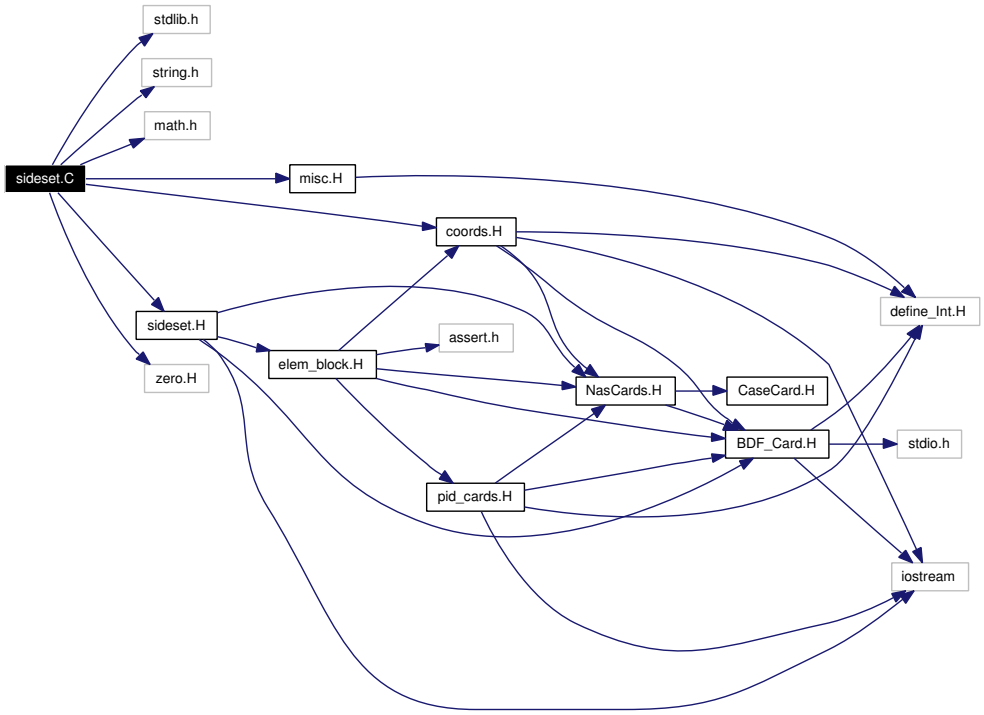
References NOT_ELEM, RBAR, RBE1, RBE2, RBE3, RROD, and RTRPLT.

7.29 sideset.C File Reference

sideset class for nasgen

```
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "coords.H"
#include "sideset.H"
#include "zero.H"
#include "misc.H"
```

Include dependency graph for sideset.C:



Functions

- int compare_sset (const void *p1, const void *p2)

7.29.1 Detailed Description

sideset class for nasgen

Definition in file sideset.C.

7.29.2 Function Documentation

7.29.2.1 int compare_sset (const void * *p1*, const void * *p2*)

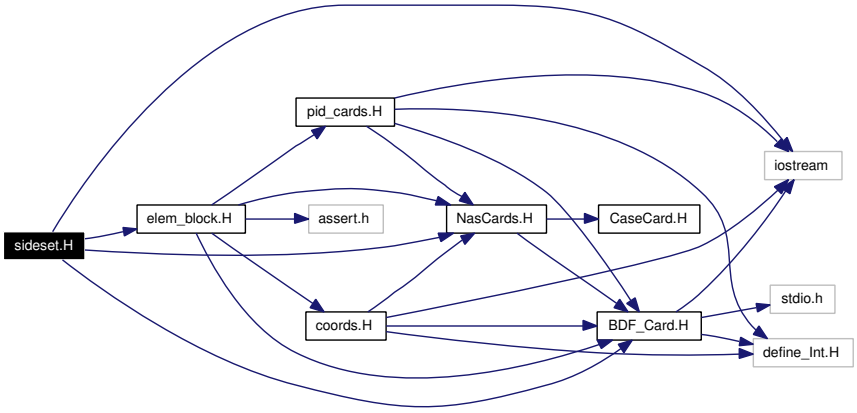
Definition at line 220 of file sideset.C.

Referenced by sideset::sideset().

7.30 sideset.H File Reference

```
#include <iostream>
#include "NasCards.H"
#include "BDF_Card.H"
#include "elem_block.H"
```

Include dependency graph for sideset.H:



Functions

- int compare_sset (const void *p1, const void *p2)

7.30.1 Function Documentation

7.30.1.1 int compare_sset (const void * *p1*, const void * *p2*)

Definition at line 220 of file sideset.C.

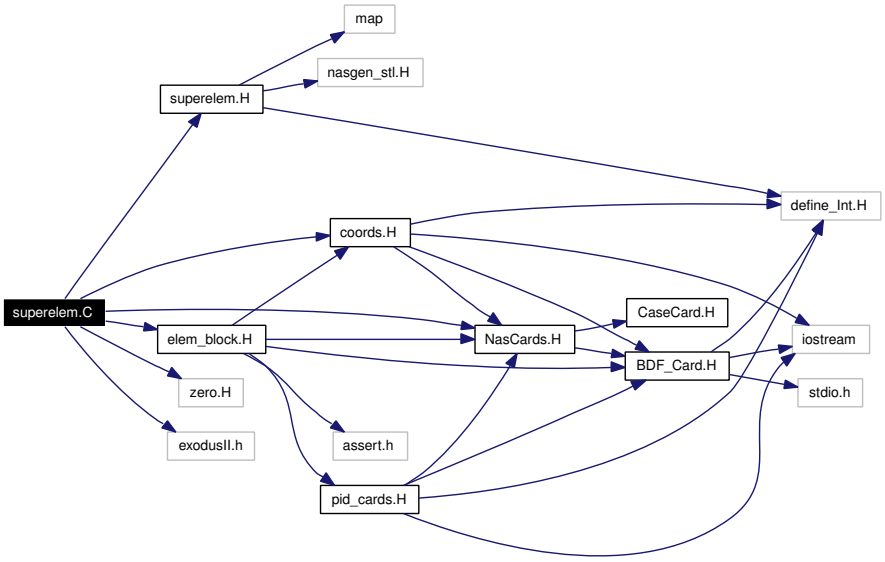
Referenced by sideset::sideset().

7.31 superelem.C File Reference

super element class for nasgen

```
#include "superelem.H"
#include "elem_block.H"
#include "coords.H"
#include "NasCards.H"
#include "zero.H"
#include "exodusII.h"
```

Include dependency graph for superelem.C:



Functions

- int irealcmp (const void *va, const void *vb)

7.31.1 Detailed Description

super element class for nasgen

Definition in file superelem.C.

7.31.2 Function Documentation

7.31.2.1 int irealcmp (const void * *va*, const void * *vb*)

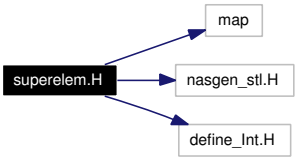
Definition at line 208 of file superelem.C.

References SuperElem::ireal::i.

7.32 superelem.H File Reference

```
#include <map>
#include "nasgen_stl.H"
#include "define_Int.H"
```

Include dependency graph for superelem.H:

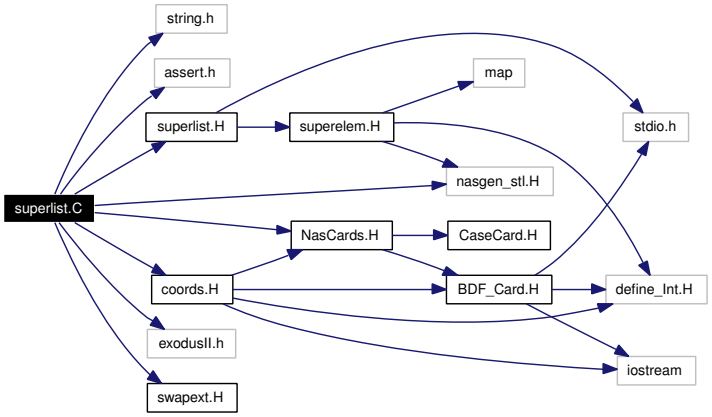


7.33 superlist.C File Reference

handle list of superelements

```
#include <string.h>
#include <assert.h>
#include "nasgen_stl.H"
#include "exodusII.h"
#include "superlist.H"
#include "NasCards.H"
#include "coords.H"
#include "swapext.H"
```

Include dependency graph for superlist.C:



7.33.1 Detailed Description

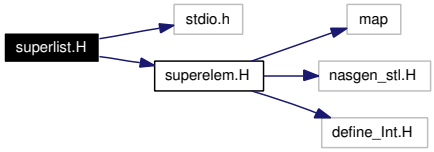
handle list of superelements

Definition in file superlist.C.

7.34 superlist.H File Reference

```
#include <stdio.h>
#include "superelem.H"
```

Include dependency graph for superlist.H:



7.35 swapext.C File Reference

Takes off an old extension (if it exists), adds a new one.

`#include <string.h>`

Include dependency graph for swapext.C:

Functions

- int SwapExt (const char *basefile, const char *ext, char *buff, size_t maxstr)

7.35.1 Detailed Description

Takes off an old extension (if it exists), adds a new one.

Definition in file swapext.C.

7.35.2 Function Documentation

7.35.2.1 int SwapExt (const char * *basefile*, const char * *ext*, char * *buff*, size_t *maxstr*)

Definition at line 15 of file swapext.C.

Referenced by NasCards::NotUsed(), write_presto::write_presto(), and write_salinas::write_salinas().

7.36 swapext.H File Reference

Functions

- int SwapExt (const char *basefile, const char *ext, char *buff, size_t maxstr=256)

7.36.1 Function Documentation

7.36.1.1 int SwapExt (const char * *basefile*, const char * *ext*, char * *buff*, size_t *maxstr* = 256)

Definition at line 15 of file swapext.C.

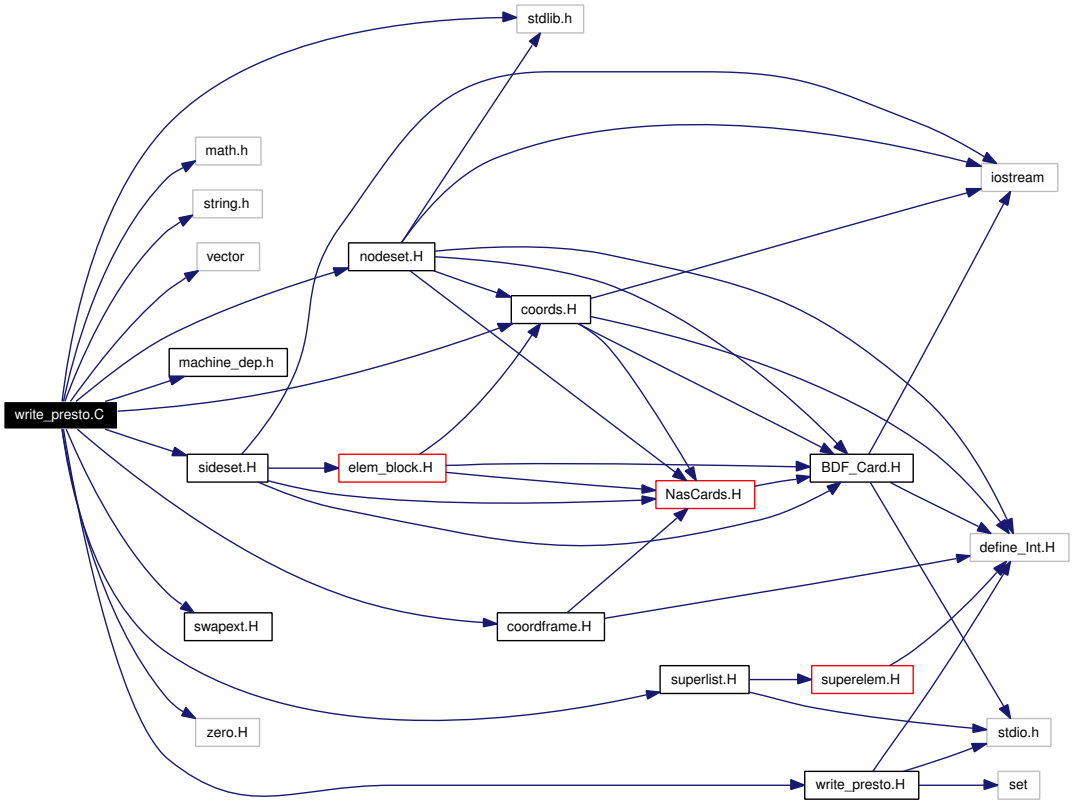
Referenced by NasCards::NotUsed(), write_presto::write_presto(), and write_salinas::write_salinas().

7.37 write_presto.C File Reference

write_presto class

```
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <vector>
#include "write_presto.H"
#include "machine_dep.h"
#include "nodeset.H"
#include "sideset.H"
#include "swapext.H"
#include "coordframe.H"
#include "coords.H"
#include "zero.H"
#include "superlist.H"
```

Include dependency graph for write_presto.C:



Defines

- #define SBSIZE 256
- #define NUM_MAT_TYPES 11
- #define M_PI 3.14159265358979323846

Functions

- const char * SetPresCmt (bool a)
- const char * Fvect (int id)
- const char * BClabel (int id)
- void putcomment (bool doit, FILE *sfile)

Variables

- int verbose

7.37.1 Detailed Description

write_presto class

Definition in file write_presto.C.

7.37.2 Define Documentation

7.37.2.1 #define M_PI 3.14159265358979323846

Definition at line 1039 of file write_presto.C.

7.37.2.2 #define NUM_MAT_TYPES 11

Referenced by write_salinas::wrt_materials(), and write_presto::wrt_materials().

7.37.2.3 #define SBSIZE 256

Referenced by write_presto::write_presto(), and write_salinas::write_salinas().

7.37.3 Function Documentation

7.37.3.1 const char* BClabel (int *id*) [inline]

Definition at line 112 of file write_presto.C.

Referenced by write_presto::wrt_remainder(), write_salinas::wrt_sets(), and write_presto::wrt_sets().

7.37.3.2 const char* Fvect (int *id*) [inline]

Definition at line 96 of file write_presto.C.

Referenced by write_presto::wrt_remainder(), write_salinas::wrt_sets(), and write_presto::wrt_sets().

7.37.3.3 void putcomment (bool *doit*, FILE * *sfile*) [inline]

Definition at line 657 of file write_presto.C.

Referenced by write_salinas::wrt_materials(), and write_presto::wrt_materials().

7.37.3.4 const char* SetPresCmt (bool *a*)

Definition at line 35 of file write_presto.C.

Referenced by write_presto::wrt_remainder(), and write_presto::wrt_sets().

7.37.4 Variable Documentation

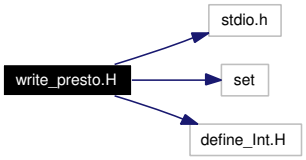
7.37.4.1 int verbose

Definition at line 26 of file write_salinas.C.

7.38 write_presto.H File Reference

```
#include <stdio.h>
#include <set>
#include "define_Int.H"
```

Include dependency graph for write_presto.H:



Functions

- void PutDouble (FILE *sfile, const BDF_Card *c, int field, const char *tag)
- int PutNotBlankDouble (FILE *sfile, const BDF_Card *c, int field, const char *tag)

7.38.1 Function Documentation

7.38.1.1 void PutDouble (FILE * *sfile*, const BDF_Card * *c*, int *field*, const char * *tag*)

Definition at line 731 of file write_salinas.C.

Referenced by PutNotBlankDouble(), write_salinas::wrt_materials(), and write_presto::wrt_materials().

7.38.1.2 int PutNotBlankDouble (FILE * *sfile*, const BDF_Card * *c*, int *field*, const char * *tag*)

Definition at line 742 of file write_salinas.C.

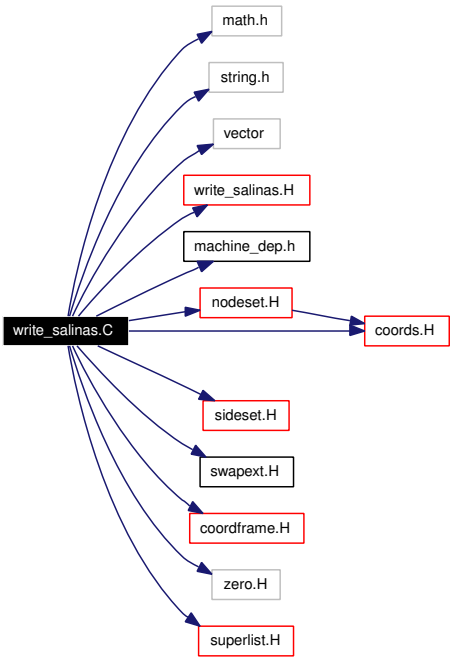
Referenced by write_salinas::wrt_materials(), and write_presto::wrt_materials().

7.39 write_salinas.C File Reference

write_salinas class

```
#include <math.h>
#include <string.h>
#include <vector>
#include "write_salinas.H"
#include "machine_dep.h"
#include "nodeset.H"
#include "sideset.H"
#include "swapext.H"
#include "coordframe.H"
#include "coords.H"
#include "zero.H"
#include "superlist.H"
```

Include dependency graph for write_salinas.C:



Defines

- #define SBSIZE 256
- #define NUM_MAT_TYPES 11
- #define M_PI 3.14159265358979323846

Functions

- const char * SetCmt (bool a)
- const char * Fvect (int id)
- const char * BClabel (int id)
- void putcomment (bool doit, FILE *sfile)
- void PutDouble (FILE *sfile, const BDF_Card *c, int field, const char *tag)
- int PutNotBlankDouble (FILE *sfile, const BDF_Card *c, int field, const char *tag)

Variables

- int verbose = 1

7.39.1 Detailed Description

write_salinas class

Definition in file write_salinas.C.

7.39.2 Define Documentation

7.39.2.1 #define M_PI 3.14159265358979323846

Definition at line 894 of file write_salinas.C.

7.39.2.2 #define NUM_MAT_TYPES 11

7.39.2.3 #define SBSIZE 256

7.39.3 Function Documentation

7.39.3.1 const char* BClabel (int *id*) [inline]

Definition at line 101 of file write_salinas.C.

7.39.3.2 const char* Fvect (int *id*) [inline]

Definition at line 85 of file write_salinas.C.

7.39.3.3 void putcomment (bool *doit*, FILE * *sfile*) [inline]

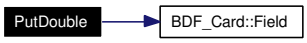
Definition at line 414 of file write_salinas.C.

7.39.3.4 void PutDouble (FILE * *sfile*, const BDF_Card * *c*, int *field*, const char * *tag*)

Definition at line 731 of file write_salinas.C.

References BDF_Card::Field().

Here is the call graph for this function:



7.39.3.5 int PutNotBlankDouble (FILE * *sfile*, const BDF_Card * *c*, int *field*, const char * *tag*)

Definition at line 742 of file write_salinas.C.

References PutDouble().

Here is the call graph for this function:

7.39.3.6 const char* SetCmt (bool *a*)

Definition at line 31 of file write_salinas.C.

Referenced by write_salinas::wrt_sets().

7.39.4 Variable Documentation

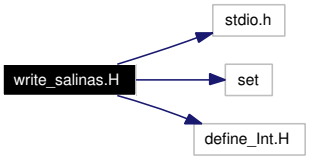
7.39.4.1 int verbose = 1

Definition at line 26 of file write_salinas.C.

7.40 write_salinas.H File Reference

```
#include <stdio.h>
#include <set>
#include "define_Int.H"
```

Include dependency graph for write_salinas.H:



Functions

- void PutDouble (FILE *sfile, const BDF_Card *c, int field, const char *tag)
- int PutNotBlankDouble (FILE *sfile, const BDF_Card *c, int field, const char *tag)

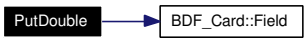
7.40.1 Function Documentation

7.40.1.1 void PutDouble (FILE * *sfile*, const BDF_Card * *c*, int *field*, const char * *tag*)

Definition at line 731 of file write_salinas.C.

References BDF_Card::Field().

Here is the call graph for this function:



7.40.1.2 int PutNotBlankDouble (FILE * *sfile*, const BDF_Card * *c*, int *field*, const char * *tag*)

Definition at line 742 of file write_salinas.C.

References PutDouble().

Here is the call graph for this function:

Index