EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH

MAD-9 Class Reference Manual

F. C. Iselin

Abstract

This report describes the classes implemented in the MAD-9 program and their relationship with the CLASSIC classes.

Contents

List of Figures

Chapter 1

Overview

The class diagrams below do not show all classes nor all dependencies, but they do explain the general structure of the program.

1.1 Module Structure

The MAD classes are grouped in several modules. The principal dependencies between the modules are shown in Fig. 1.1.

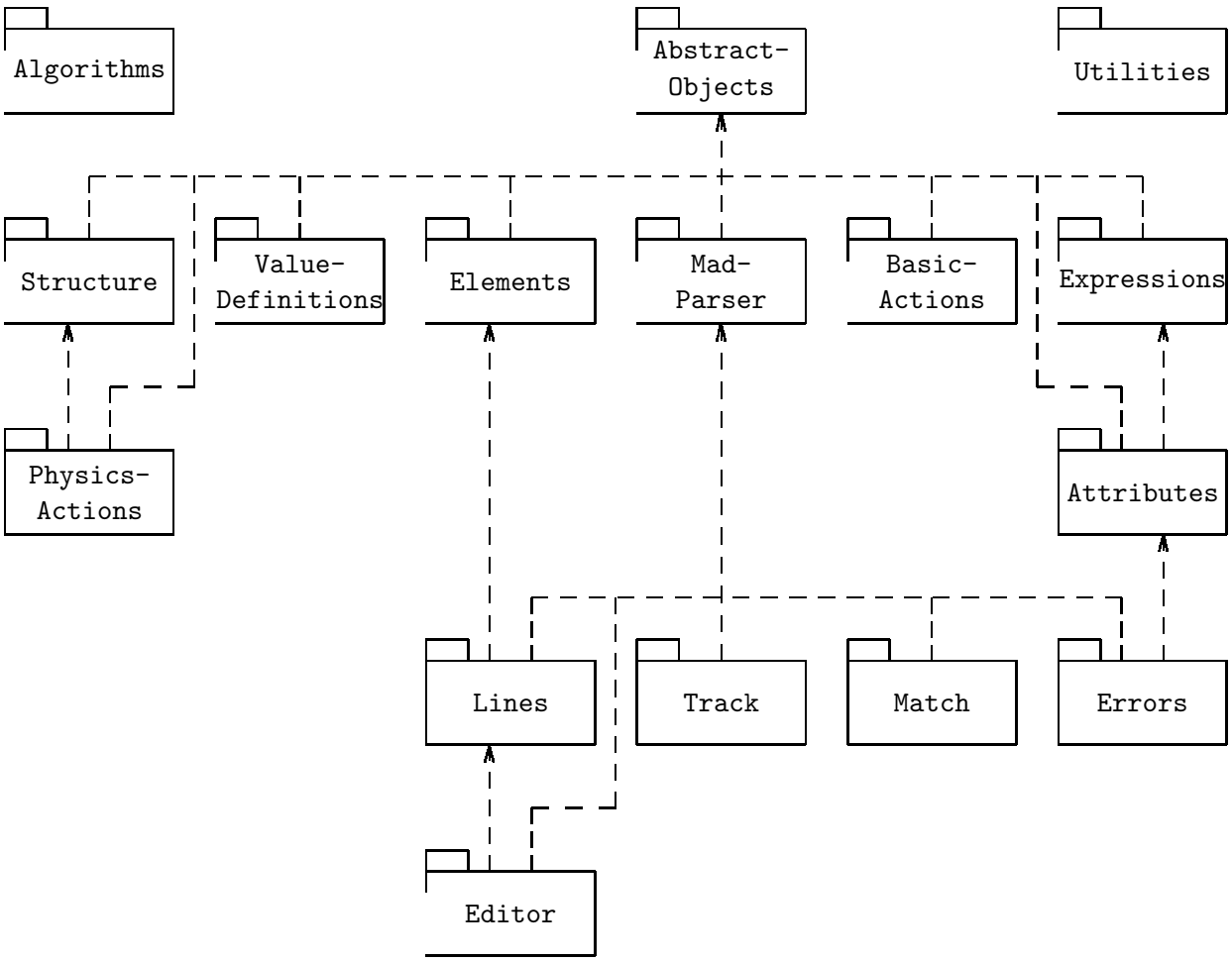


Figure 1.1: CLASSIC Modules and their Dependences

1.2 Hierarchy of Command Objects

All MAD commands are represented by object classes. At start-up time MAD-9 creates a "model" object of each class, known as an "exemplar". When a parser reads a command, it clones the named object, reads its attributes, and executes the clone object by calling its execute() method. For details refer to the description of the parsers. The top level of the "Object" hierarchy is shown in Fig. 1.2.

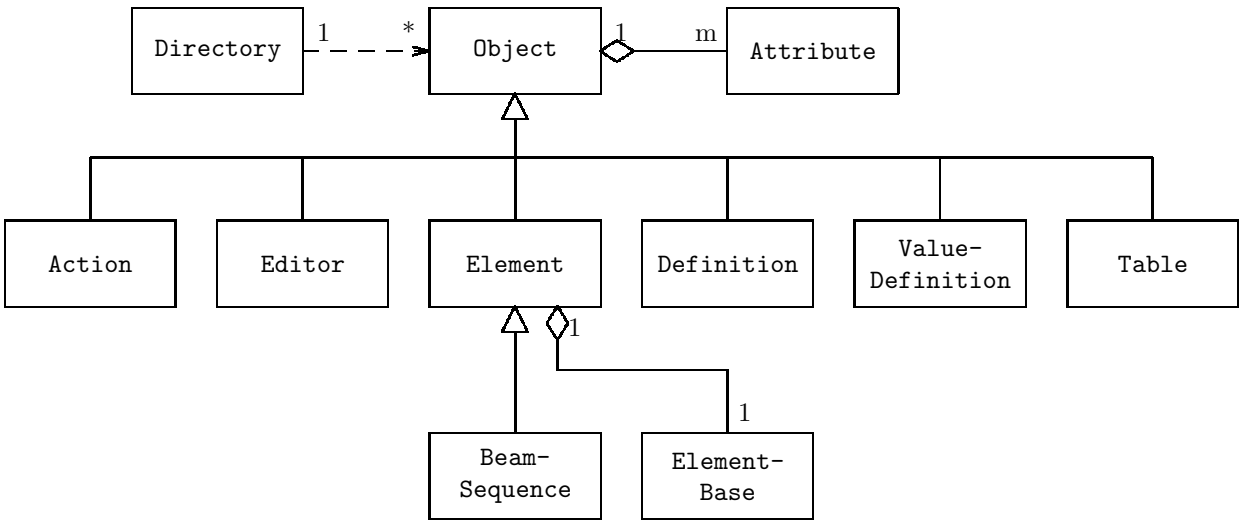


Figure 1.2: Hierarchy for Command Objects

1.3 Command Attributes

Each attribute is represented by an object of class "Attribute", arranged in an array contained in the command object. Each attribute points to

1. An AttributeBase. This contains the attribute value, and is normally set to a default value in the exemplar object. The clone process for the object creates a new reference to the value in the exemplar, and any new value overrides this value. The values are templates, and may return several data types (`double, string, PlaceRep, RangeRep, vector<>`, etc.).

2. An AttributeHandler. This contains the name and the help text for the attribute. It takes care of parsing and printing the attribute. The attribute handlers for the same attribute of all object of a given class are shared.

The relationship of attribute objects is shown in Fig. 1.3, the hierarchy for attribute values in Fig. 1.4, and the hierarchy for attribute values in Fig. 1.5.
Expressions entered by the user are kept in tree form. Expression trees are built from the classes in the hierarchy shown in Fig. 1.6. The expression classes are templates, and may return several data types (`double, string, PlaceRep, RangeRep, vector<>`, etc.).

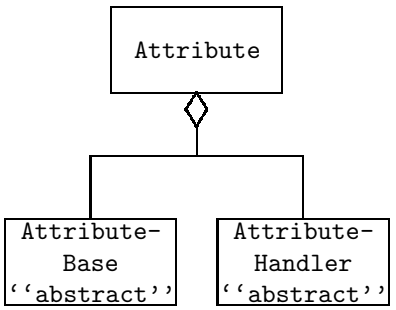


Figure 1.3: Relationships between Attribute Objects

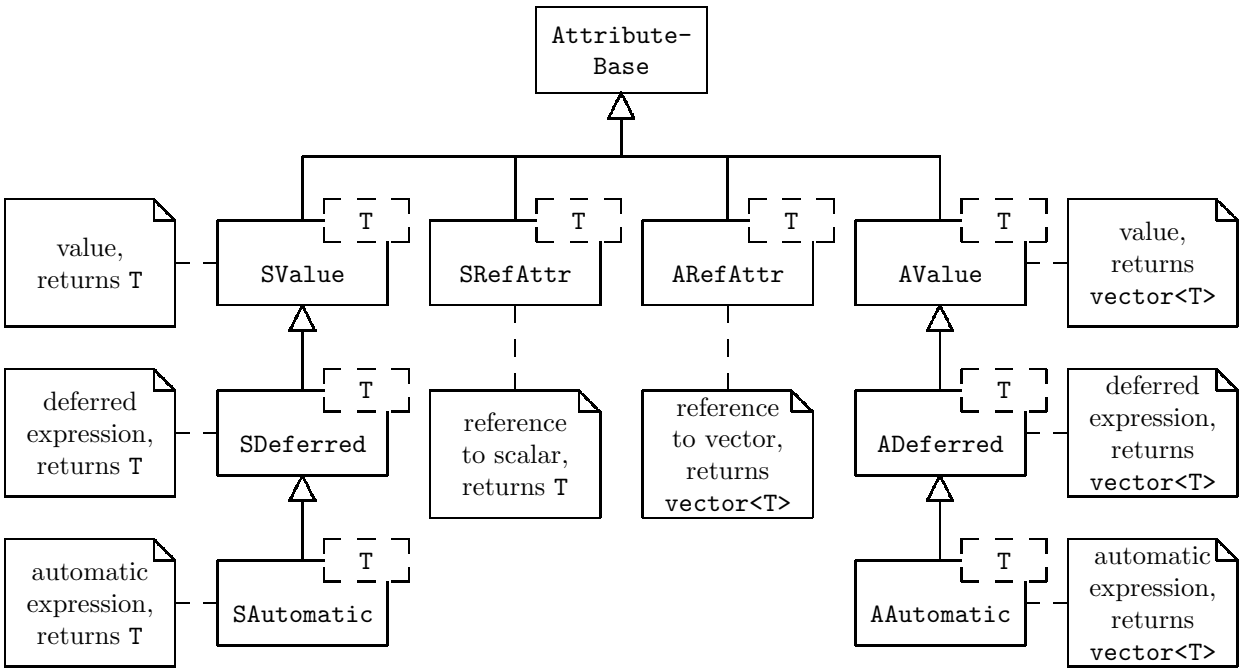


Figure 1.4: Hierarchy for Attribute Values

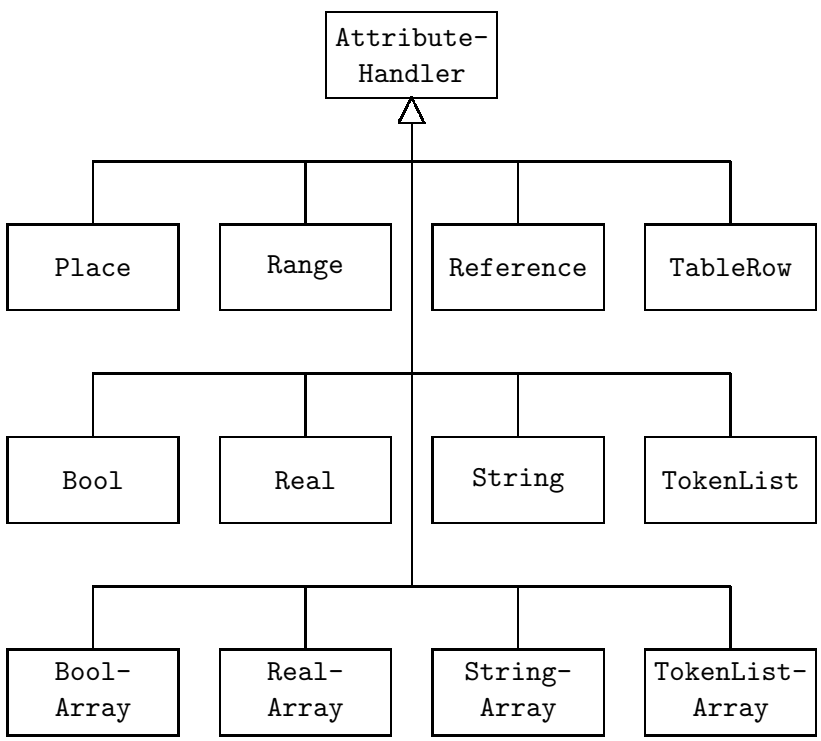


Figure 1.5: Hierarchy for Attribute Handlers

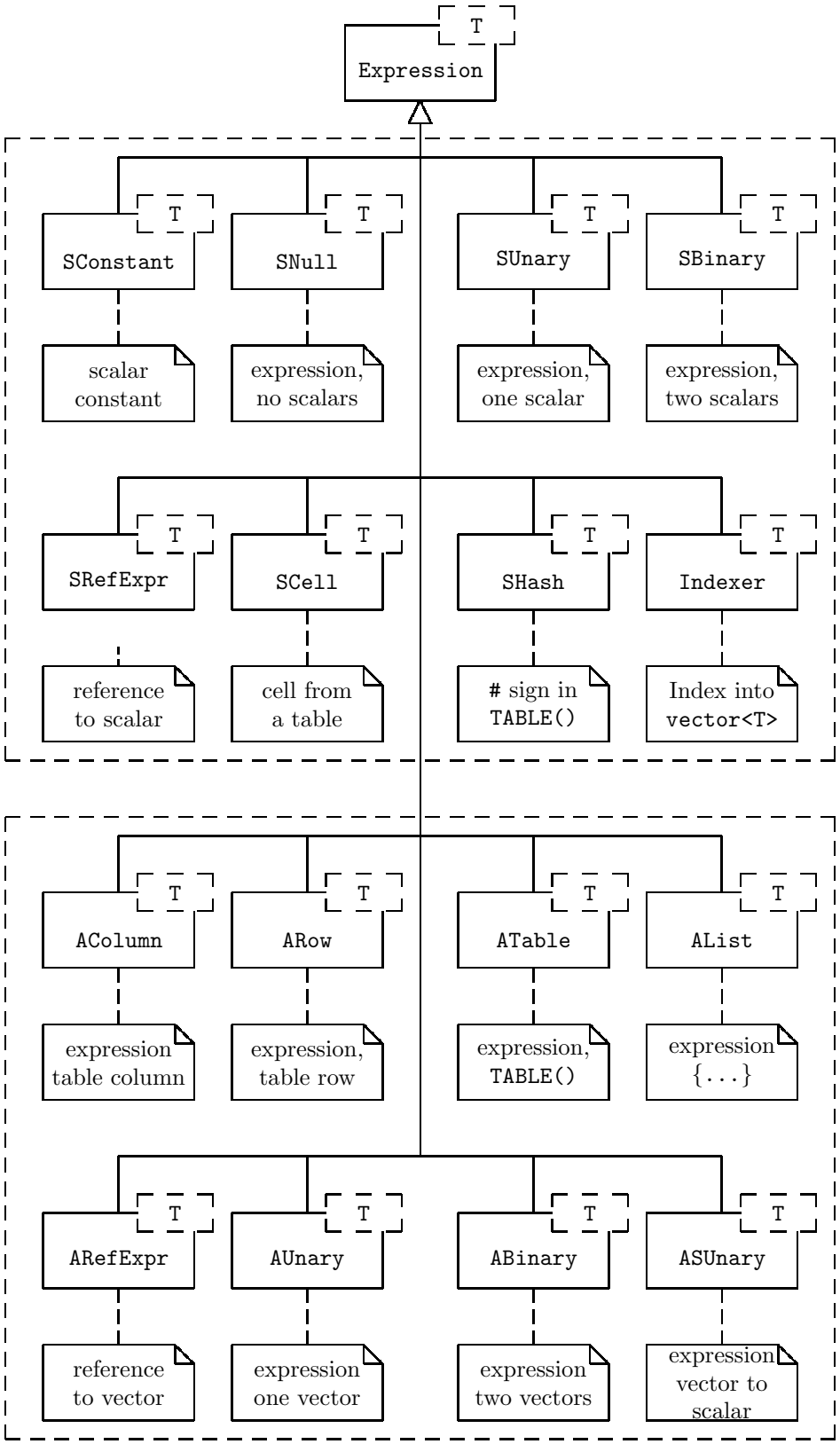


Figure 1.6: Hierarchy for scalar (top) and vector (bottom) Expressions

1.4 Active Elements

The MAD element classes serve as an interface to the CLASSIC library. Each MAD element contains a pointer to an instance of a corresponding CLASSIC element. The CLASSIC element handles the interface to the algorithms, whilst the MAD element deals with the attributes in MAD language, and updates the CLASSIC element when its `update()` function is called. This happens before execution of any executable command, so as to ensure that the data structure is up to date.
The MAD element hierarchy is illustrated by two typical examples, namely the MadDrift and MadBend elements in Fig. 1.7.

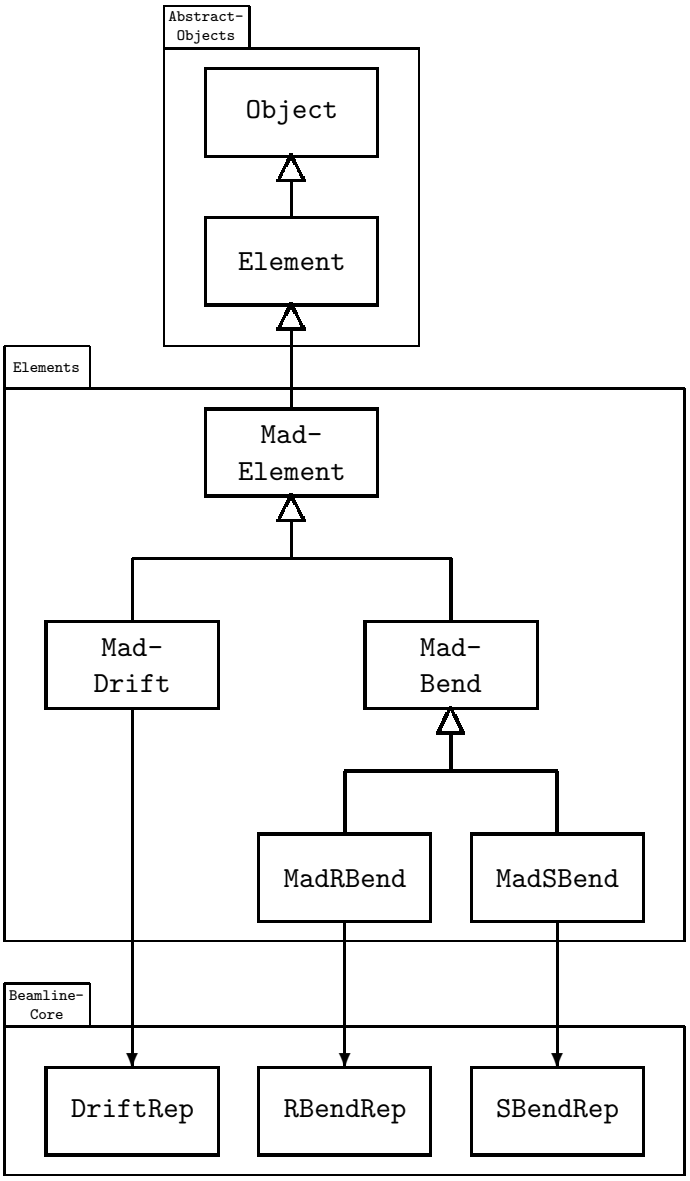


Figure 1.7: Hierarchy for Drift and Bend Elements

1.5 Beam Lines

The MAD beam lines are treated like the elements. Again, a Line or Sequence points to a CLASSIC beam line. This is shown in Fig. 1.8. This same figure also shows the templates used for lines and sequences with arguments.

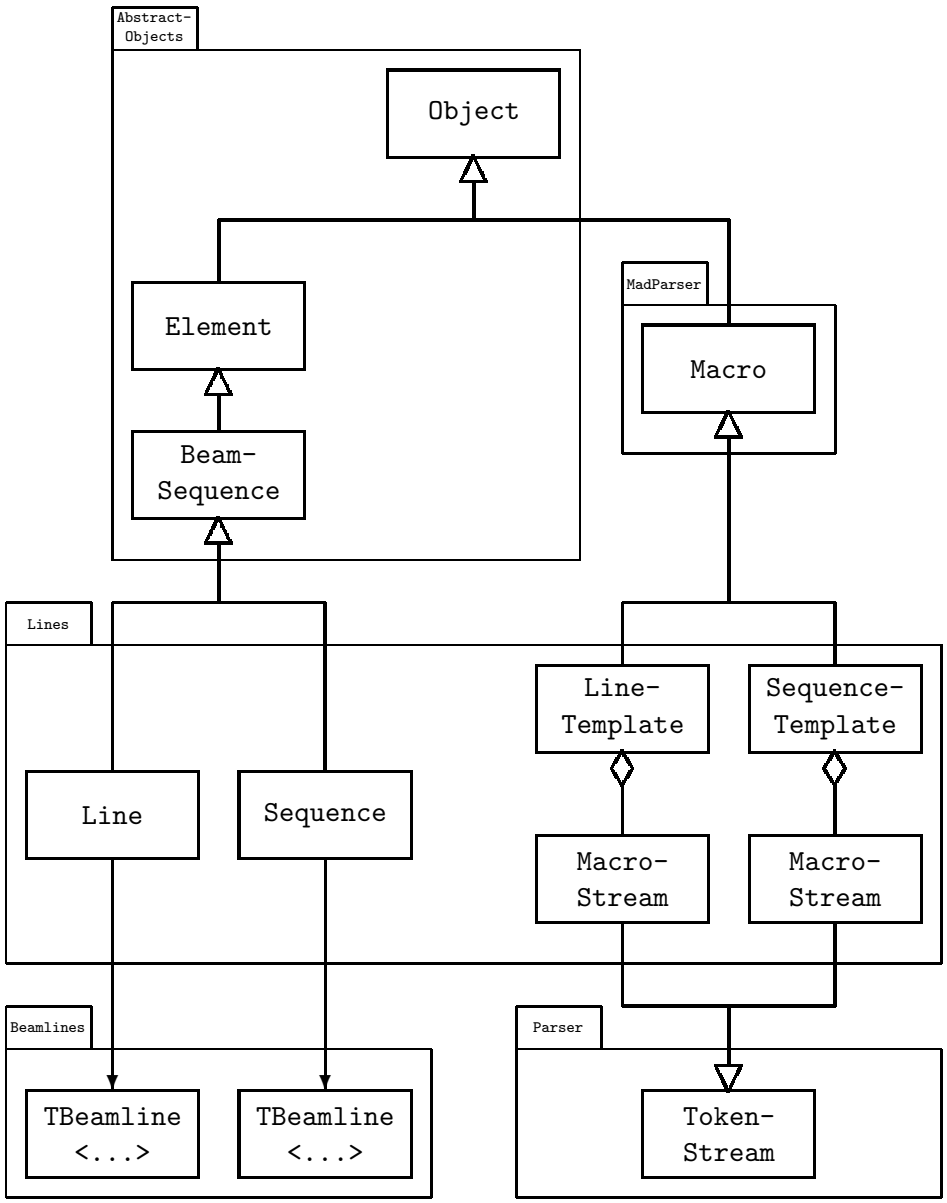


Figure 1.8: Hierarchy for MAD Beam Lines and Sequences

1.6 Line Structure

The structure of a beam line or sequence follows the layout shown in Fig. 1.9. For non-shared elements, all the objects except the concrete representations for **ideal** elemenst (`XXXRep`) are unique. For shared elements, also the wrappers (`XXXWrapper`) are unique.

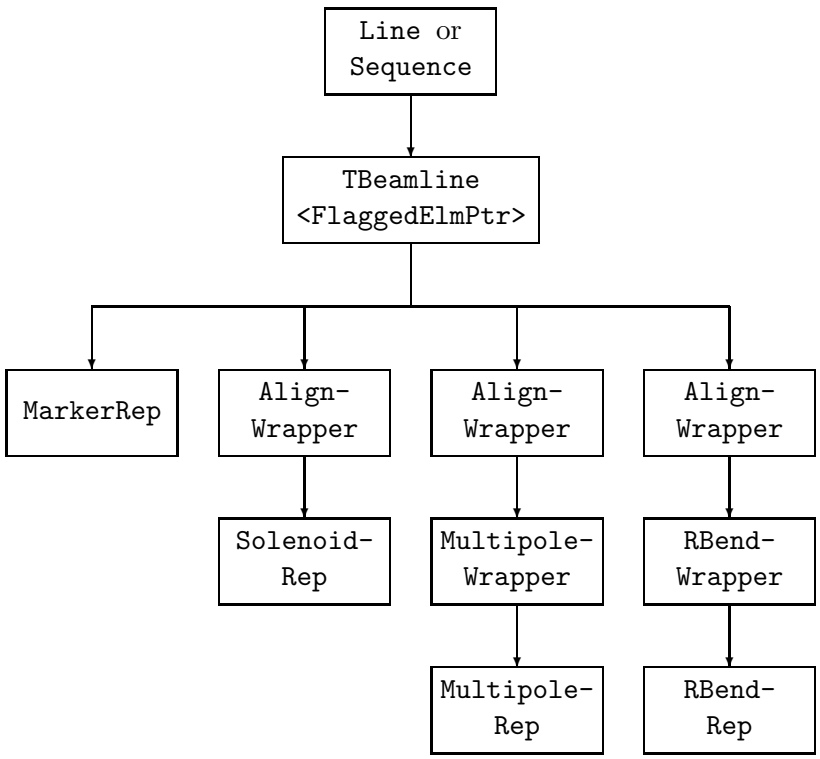


Figure 1.9: Structure of a MAD Beam Line or Sequence

1.7 Algorithms

The algorithm hierarchy is shown in Fig. 1.10. An algorithms is constructed with a beam line as a parameter, and normally applied to that line by its `execute()` method. Most algorithms can also be applied to a table by simply looping over the table and using the `accept(BeamlineVisitor &)` method of each contained `ElmPtr`.
The integrator hierarchy is shown in Fig. 1.11.

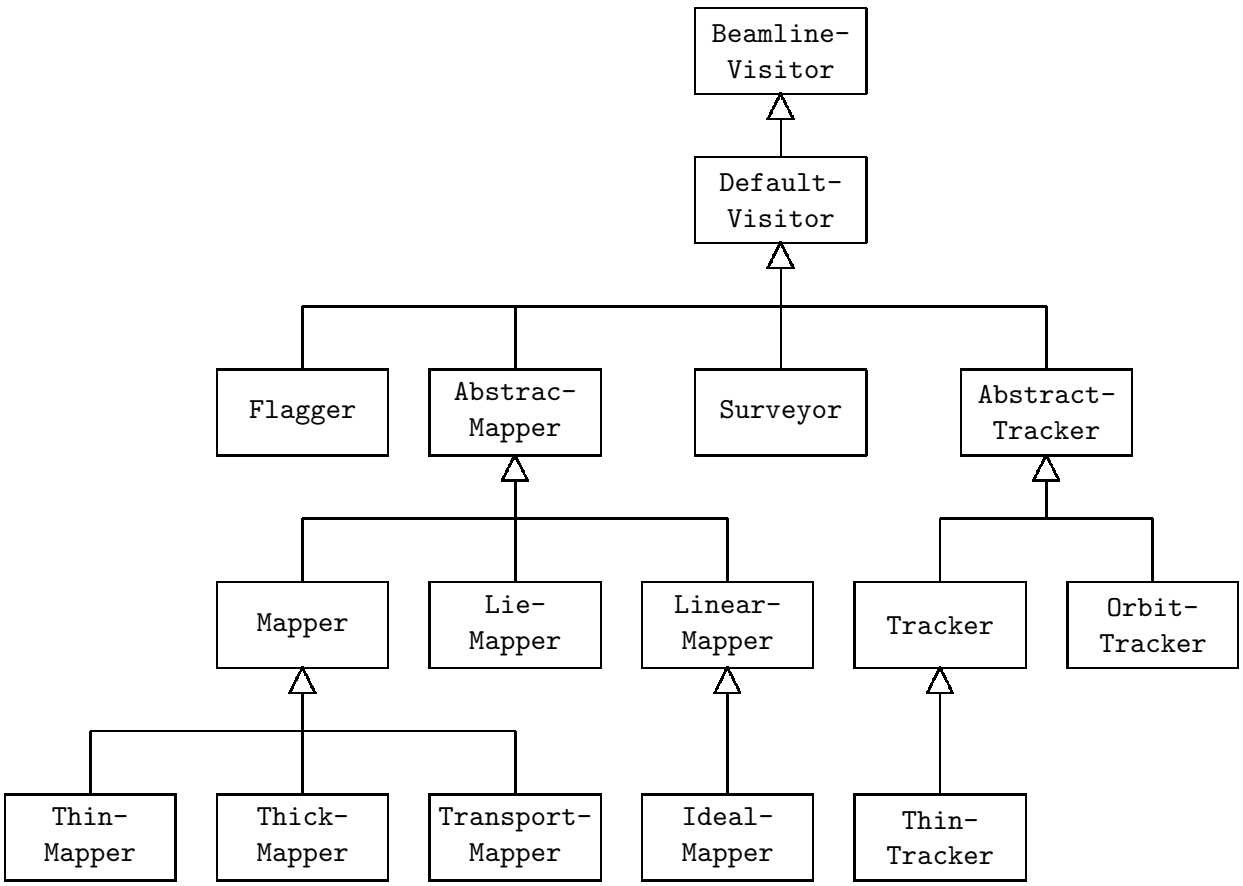


Figure 1.10: Hierarchy for MAD and CLASSIC Algorithms

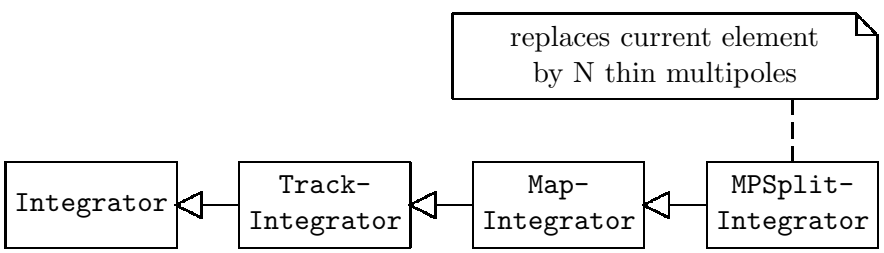


Figure 1.11: Hierarchy for MAD and CLASSIC Integrators

1.8 MAD Parser and Streams

MAD uses the class `MadParser` for parsing in the normal mode. When entering one of the special modes, it switches to another parser, derived from `MadParser`. The hierarchy for parsers and input streams is shown in Fig. 1.12.

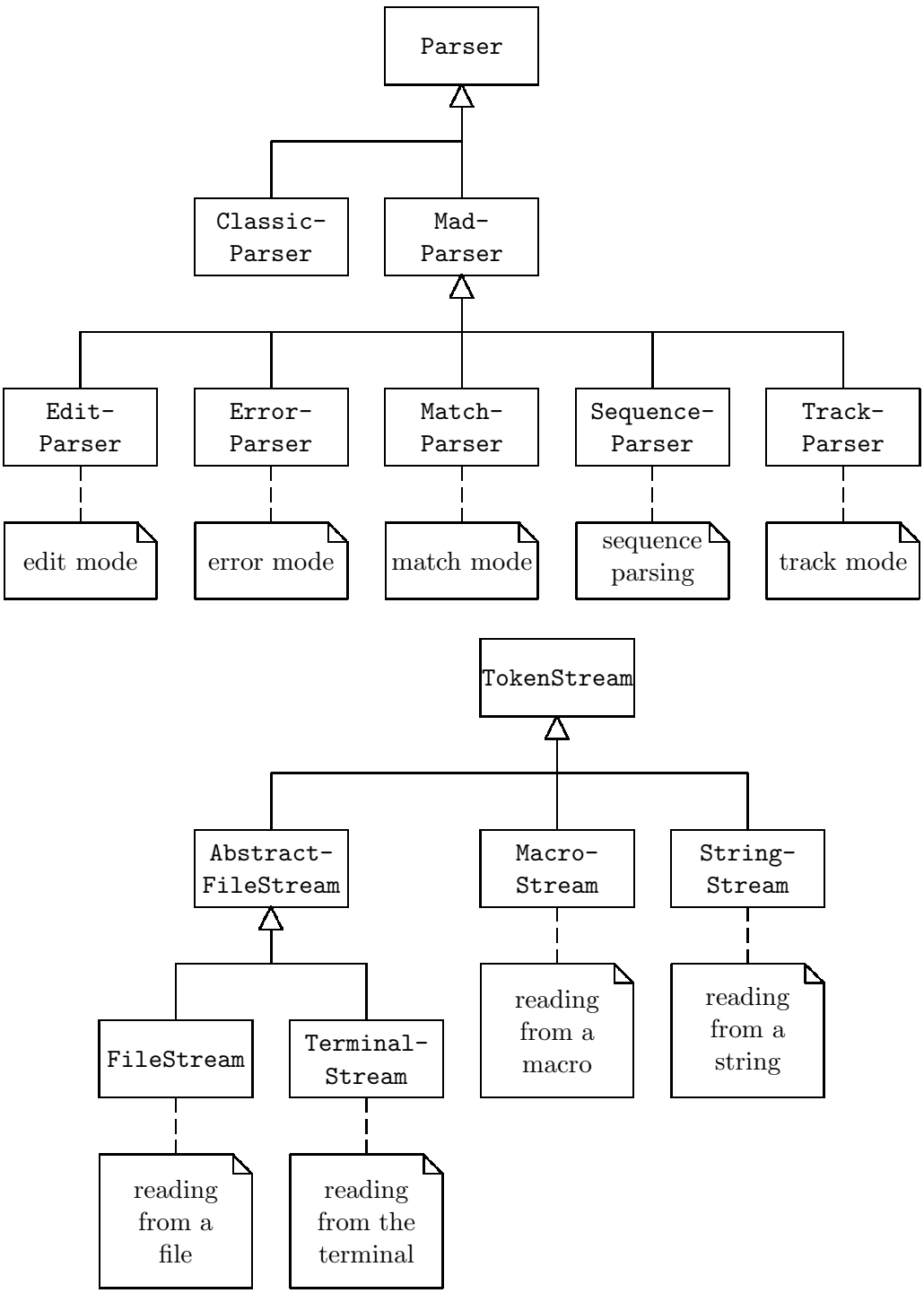


Figure 1.12: Hierarchy for Parsers (top) and Streams (bottom)

Chapter 2

Detailed Class Descriptions

2.1 namespace Attributes

A collection of routines to construct and read object attributes.
This namespace is part of the interface to the class Attribute, used to store object attributes.

Include file:	./Attributes/Attributes.hh

Synopsis

Members

- **class Bool**
  Parser for attribute of type logical. (see Section 2.1.1)
- **class BoolArray**
  Parser for an attribute of type logical array. (see Section 2.1.1)
- **class Place**
  Parser for an attribute of type place reference. (see Section 2.1.1)
- **class Range**
  Parser for an attribute of type range definition. (see Section 2.1.1)
- **class Real**
  Parser for an attribute of type real. (see Section 2.1.1)
- **class RealArray**
  Parser for an attribute of type real array. (see Section 2.1.1)
- **class Reference**
  Parser for an attribute of type attribute reference. (see Section 2.1.1)
- **class String**
  Parser for an attribute of type string. (see Section 2.1.1)
- **class StringArray**
  Parser for an attribute of type string array. (see Section 2.1.1)
- **class TableRow**
  Parser for an attribute of type table row reference. (see Section 2.1.1)
- **class TokenList**
  Parser for an attribute of type token list. (see Section 2.1.1)
- **class TokenListArray**
  Parser for an attribute of type token list array. (see Section 2.1.1)
- **bool getBool (const Attribute&)**
  Return logical value. (see Section 2.1.1)
- **extern std::vector<bool> getBoolArray (const Attribute&)**
  Get logical array value. (see Section 2.1.1)
- **extern PlaceRep getPlace (const Attribute&)**
  Get place value. (see Section 2.1.1)

- **extern RangeRep getRange (const Attribute&)**
  Get range value. (see Section 2.1.1)
- **extern double getReal (const Attribute&)**
  Return real value. (see Section 2.1.1)
- **extern std::vector<double> getRealArray (const Attribute&)**
  Get array value. (see Section 2.1.1)
- **extern string getString (const Attribute&)**
  Get string value. (see Section 2.1.1)
- **extern std::vector<string> getStringArray (const Attribute&)**
  Get string array value. (see Section 2.1.1)
- **extern TableRowRep getTableRow (const Attribute&)**
  Get table row value. (see Section 2.1.1)
- **extern std::list<Token> getTokenList (const Attribute&)**
  Return token list value. (see Section 2.1.1)
- **extern std::vector<std::list<Token> > getTokenListArray (const Attribute&)**
  Return token list array value. (see Section 2.1.1)
- **extern Attribute makeBool (const string&,const string&)**
  Make logical attribute. (see Section 2.1.1)
- **extern Attribute makeBool (const string&,const string&,bool)**
  Make logical attribute. (see Section 2.1.1)
- **extern Attribute makeBoolArray (const string&,const string&)**
  Create a logical array attribute. (see Section 2.1.1)
- **extern Attribute makePlace (const string&,const string&)**
  Create a place attribute. (see Section 2.1.1)
- **extern Attribute makeRange (const string&,const string&)**
  Create a range attribute. (see Section 2.1.1)
- **extern Attribute makeReal (const string&,const string&)**
  Make real attribute. (see Section 2.1.1)
- **extern Attribute makeReal (const string&,const string&,double)**
  Make real attribute. (see Section 2.1.1)
- **extern Attribute makeRealArray (const string&,const string&)**
  Create real array attribute. (see Section 2.1.1)
- **extern Attribute makeReference (const string&,const string&)**
  Create a reference attribute. (see Section 2.1.1)
- **extern Attribute makeString (const string&,const string&)**
  Make string attribute. (see Section 2.1.1)

- **extern Attribute makeString (const string&,const string&,const string&)**
  Make string attribute. (see Section 2.1.1)
- **extern Attribute makeStringArray (const string&,const string&)**
  Create a string array attribute. (see Section 2.1.1)
- **extern Attribute makeTableRow (const string&,const string&)**
  Create a table row attribute. (see Section 2.1.1)
- **extern Attribute makeTokenList (const string&,const string&)**
  Make token list attribute. (see Section 2.1.1)
- **extern Attribute makeTokenListArray (const string&,const string&)**
  Make token list attribute. (see Section 2.1.1)
- **extern void setBool (Attribute&,bool)**
  Set logical value. (see Section 2.1.1)
- **extern void setBoolArray (Attribute&,const std::vector<bool>&)**
  Set logical array value. (see Section 2.1.1)
- **extern void setPlace (Attribute&,const PlaceRep&)**
  Set place value. (see Section 2.1.1)
- **extern void setRange (Attribute&,const RangeRep&)**
  Set range value. (see Section 2.1.1)
- **extern void setReal (Attribute&,double)**
  Set real value. (see Section 2.1.1)
- **extern void setRealArray (Attribute&,const std::vector<double>&)**
  Set array value. (see Section 2.1.1)
- **extern void setString (Attribute&,const string&)**
  Set string value. (see Section 2.1.1)
- **extern void setStringArray (Attribute&,const std::vector<string>&)**
  Set string array value. (see Section 2.1.1)
- **extern void setTableRow (Attribute&,const TableRowRep&)**
  Set table row value. (see Section 2.1.1)
- **extern void setTokenList (Attribute&,const std::list<Token>&)**
  Set token list value. (see Section 2.1.1)
- **extern void setTokenListArray (Attribute&,const std::vector<std::list<Token>>&)**
  Set token list array value. (see Section 2.1.1)

2.1.1 Detailed descriptions

Public members

- **class Bool**
  Parser for attribute of type logical.
- **class BoolArray**
  Parser for an attribute of type logical array.
- **class Place**
  Parser for an attribute of type place reference.
- **class Range**
  Parser for an attribute of type range definition.
- **class Real**
  Parser for an attribute of type real.
- **class RealArray**
  Parser for an attribute of type real array.
- **class Reference**
  Parser for an attribute of type attribute reference.

  The attribute referred to may be logical, real, or string.
- **class String**
  Parser for an attribute of type string.
- **class StringArray**
  Parser for an attribute of type string array.
- **class TableRow**
  Parser for an attribute of type table row reference.
- **class TokenList**
  Parser for an attribute of type token list.

  Such an attribute can encode e.g. an expression.
- **class TokenListArray**
  Parser for an attribute of type token list array.

  Such an attribute may encode a list of expressions.
- **bool getBool (const Attribute&)**
  Return logical value.

  Evaluate any expression and return value. If the expression is non-deferred, the value is cached. Re-evaluation takes place only if a new definition has been read.
- **extern std::vector<bool> getBoolArray (const Attribute&)**
  Get logical array value.
- **extern PlaceRep getPlace (const Attribute&)**
  Get place value.

- **extern RangeRep getRange (const Attribute&)**
  Get range value.
- **extern double getReal (const Attribute&)**
  Return real value.
- **extern std::vector<double> getRealArray (const Attribute&)**
  Get array value.
- **extern string getString (const Attribute&)**
  Get string value.
- **extern std::vector<string> getStringArray (const Attribute&)**
  Get string array value.
- **extern TableRowRep getTableRow (const Attribute&)**
  Get table row value.
- **extern std::list<Token> getTokenList (const Attribute&)**
  Return token list value.
- **extern std::vector<std::list<Token> > getTokenListArray (const Attribute&)**
  Return token list array value.
- **extern Attribute makeBool (const string&,const string&)**
  Make logical attribute.

  Initial value undefined.
- **extern Attribute makeBool (const string&,const string&,bool)**
  Make logical attribute.

  Initial value defined.
- **extern Attribute makeBoolArray (const string&,const string&)**
  Create a logical array attribute.

  Initial value is empty array.
- **extern Attribute makePlace (const string&,const string&)**
  Create a place attribute.

  Initial value is "#S".
- **extern Attribute makeRange (const string&,const string&)**
  Create a range attribute.

  Initial value is "FULL".
- **extern Attribute makeReal (const string&,const string&)**
  Make real attribute.

  Initial value is undefined.
- **extern Attribute makeReal (const string&,const string&,double)**
  Make real attribute.

  Initial value is defined.

- **extern Attribute makeRealArray (const string&,const string&)**
  Create real array attribute.

  Initial value is empty array.

- **extern Attribute makeReference (const string&,const string&)**
  Create a reference attribute.

  Initial value is undefined.

- **extern Attribute makeString (const string&,const string&)**
  Make string attribute.

  Initial value undefined.

- **extern Attribute makeString (const string&,const string&,const string&)**
  Make string attribute.

  Initial value is defined.

- **extern Attribute makeStringArray (const string&,const string&)**
  Create a string array attribute.

  Initial value is empty array.

- **extern Attribute makeTableRow (const string&,const string&)**
  Create a table row attribute.

  Initial value is undefined.

- **extern Attribute makeTokenList (const string&,const string&)**
  Make token list attribute.

  Initial value is empty list.

- **extern Attribute makeTokenListArray (const string&,const string&)**
  Make token list attribute.

  Initial value is empty array.

- **extern void setBool (Attribute&,bool)**
  Set logical value.

- **extern void setBoolArray (Attribute&,const std::vector<bool>&)**
  Set logical array value.

- **extern void setPlace (Attribute&,const PlaceRep&)**
  Set place value.

- **extern void setRange (Attribute&,const RangeRep&)**
  Set range value.

- **extern void setReal (Attribute&,double)**
  Set real value.

- **extern void setRealArray (Attribute&,const std::vector<double>&)**
  Set array value.

- **extern void setString (Attribute&,const string&)**
  Set string value.
- **extern void setStringArray (Attribute&,const std::vector<string>&)**
  Set string array value.
- **extern void setTableRow (Attribute&,const TableRowRep&)**
  Set table row value.
- **extern void setTokenList (Attribute&,const std::list<Token>&)**
  Set token list value.
- **extern void setTokenListArray (Attribute&,const std::vector<std::list<Token>>&)**
  Set token list array value.

2.2 namespace Configure

The MAD configurator.
This class must be modified to configure the commands to be contained in an executable MAD-9 program. For each command an exemplar object is constructed and linked to the main directory. This exemplar is then available to the MAD parser for cloning. This class could be part of the class MadData. It is separated from that class and made into a special module in order to reduce dependencies between modules.

Include file:	./Configure/Configure.hh

Synopsis

Members

- **extern void configure ()**
  Configure all commands. (see Section 2.2.1)

2.2.1 Detailed descriptions

Public members

- **extern void configure ()**
  Configure all commands.

2.3 namespace Expressions

Representation objects and parsers for attribute expressions.

Include file:	./Expressions/TFunction2.hh

Synopsis

Members

- **class AAutomatic**
  Object attribute with an "automatic" array value. (see Section 2.3.1)
- **class ABinary**
  An array expression with two array operands. (see Section 2.3.1)
- **class AColumn**
  An array expression defined as a table column. (see Section 2.3.1)
- **class ADeferred**
  Object attribute with a "deferred" array value. (see Section 2.3.1)
- **class AList**
  An array expression defined by a list of scalar expressions. (see Section 2.3.1)
- **class ARefAttr**
  An attribute defined as a reference to an array. (see Section 2.3.1)
- **class ARefExpr**
  An expression defined as a reference to an array. (see Section 2.3.1)
- **class ARow**
  An array expression defined as a table row. (see Section 2.3.1)
- **class ASUnary**
  A scalar expression with one array operand. (see Section 2.3.1)
- **class ATable**
  An array expression generated from a TABLE() function. (see Section 2.3.1)
- **class AUnary**
  An array expression with one array operand. (see Section 2.3.1)
- **class AValue**
  Object attribute with a constant array value. (see Section 2.3.1)
- **class Array**
  An array expression. (see Section 2.3.1)
- **class ArrayOfPtrs**
  An array of pointers to scalar expressions. (see Section 2.3.1)
- **class Indexer**
  A scalar expression used to retrieve an indexed component from an (see Section 2.3.1)

- **class PtrToArray**
  A pointer to an array expression. (see Section 2.3.1)
- **class PtrToScalar**
  A pointer to a scalar expression. (see Section 2.3.1)
- **class SAutomatic**
  Object attribute with an "automatic" scalar value. (see Section 2.3.1)
- **class SBinary**
  A scalar expression with two scalar operands. (see Section 2.3.1)
- **class SCell**
  A scalar expression referring to a table cell. (see Section 2.3.1)
- **class SConstant**
  A scalar constant expression. (see Section 2.3.1)
- **class SDeferred**
  Object attribute with a "deferred" scalar value. (see Section 2.3.1)
- **class SHash**
  A scalar expression. (see Section 2.3.1)
- **class SNull**
  A scalar expression without operands. (see Section 2.3.1)
- **class SRefAttr**
  An attribute defined as a reference to a scalar. (see Section 2.3.1)
- **class SRefExpr**
  An expression defined as a reference to a scalar. (see Section 2.3.1)
- **class SUnary**
  A scalar expression with one scalar operand. (see Section 2.3.1)
- **class SValue**
  Object attribute with a constant scalar value. (see Section 2.3.1)
- **class Scalar**
  A scalar expression. (see Section 2.3.1)
- **struct TFunction0**
  An operand-less function returning a T. (see Section 2.3.1)
- **struct TFunction1**
  A function of one U, returning a T. (see Section 2.3.1)
- **struct TFunction2**
  A function of two U's returning a T. (see Section 2.3.1)
- **inline const T* find (const T[],const string&)**
  Look up name. (see Section 2.3.1)

- **inline std::ostream& operator<< (std::ostream&,const ARefAttr<T>&)**
  (see Section 2.3.1)
- **inline std::ostream& operator<< (std::ostream&,const SRefAttr<T>&)**
  (see Section 2.3.1)
- **extern PtrToScalar<bool> parseBool (Statement&)**
  Parse boolean expression. (see Section 2.3.1)
- **extern PtrToArray<bool> parseBoolArray (Statement&)**
  Parse boolean array expression. (see Section 2.3.1)
- **extern void parseDelimiter (Statement&,char)**
  Test for one-character delimiter. (see Section 2.3.1)
- **extern void parseDelimiter (Statement&,const char[2])**
  Test for two-character delimiter. (see Section 2.3.1)
- **extern PlaceRep parsePlace (Statement&)**
  Parse place specification. (see Section 2.3.1)
- **extern RangeRep parseRange (Statement&)**
  Parse range specification. (see Section 2.3.1)
- **extern PtrToScalar<double> parseReal (Statement&)**
  Parse real expression. (see Section 2.3.1)
- **extern PtrToArray<double> parseRealArray (Statement&)**
  Parse real array expression. (see Section 2.3.1)
- **extern double parseRealConst (Statement&)**
  Parse real constant. (see Section 2.3.1)
- **extern PtrToArray<double> parseRealConstArray (Statement&)**
  Parse real array constant. (see Section 2.3.1)
- **extern SRefAttr<double>* parseReference (Statement&)**
  Parse variable reference. (see Section 2.3.1)
- **extern string parseString (Statement&,const char[])**
  Parse string value. (see Section 2.3.1)
- **extern std::vector<string> parseStringArray (Statement&)**
  Parse string array. (see Section 2.3.1)
- **extern PtrToScalar<double> parseTableExpression (Statement&,const Table*)**
  Parse table expression (depends on a table's rows). (see Section 2.3.1)
- **extern TableRowRep parseTableRow (Statement&)**
  Parse a token list (for macro argument and the like). (see Section 2.3.1)
- **extern std::list<Token> parseTokenList (Statement&)**
  Parse a token list (for macro argument and the like). (see Section 2.3.1)
- **extern std::vector<std::list<Token> > parseTokenListArray (Statement&)**
  Parse a token list array (for LIST commands). (see Section 2.3.1)

2.3.1 Detailed descriptions

Public members

- **class AAutomatic**
  Object attribute with an "automatic" array value.

  An automatic expression is marked as unknown and registered in a list when it is created. When its value is required, it is evaluated, and the value is cached. Whenever a new definition is read, or an existing one is changed, the expressions is marked as unknown. This forces a new evaluation when it is used the next time.

- **class ABinary**
  An array expression with two array operands.

- **class AColumn**
  An array expression defined as a table column.

- **class ADeferred**
  Object attribute with a "deferred" array value.

  An deferred expression is always re-evaluated when its value is required. This is notably needed to implement random generators.

- **class AList**
  An array expression defined by a list of scalar expressions.

- **class ARefAttr**
  An attribute defined as a reference to an array.

  The components of the array may be real, logical or string.

- **class ARefExpr**
  An expression defined as a reference to an array.

  The components of the array may be real, logical or string.

- **class ARow**
  An array expression defined as a table row.

- **class ASUnary**
  A scalar expression with one array operand.

- **class ATable**
  An array expression generated from a TABLE() function.

  This expression uses one or more AHash objects to represent the current index in the TABLE expression. These may retrieve the index from the ATable object by calling getHash().

- **class AUnary**
  An array expression with one array operand.

- **class AValue**
  Object attribute with a constant array value.

- **class Array**
  An array expression.

- **class ArrayOfPtrs**
  An array of pointers to scalar expressions.

- **class Indexer**
  A scalar expression used to retrieve an indexed component from an array.

- **class PtrToArray**
  A pointer to an array expression.

- **class PtrToScalar**
  A pointer to a scalar expression.

- **class SAutomatic**
  Object attribute with an "automatic" scalar value.

  An automatic expression is marked as unknown and registered in a list when it is created. When its value is required, it is evaluated, and the value is cached. Whenever a new definition is read, or an existing one is changed, the expressions is again marked as unknown. This forces a new evaluation when it is used the next time.

- **class SBinary**
  A scalar expression with two scalar operands.

- **class SCell**
  A scalar expression referring to a table cell.

- **class SConstant**
  A scalar constant expression.

- **class SDeferred**
  Object attribute with a "deferred" scalar value.

  An deferred expression is always re-evaluated when its value is required. This is notably needed to implement random generators.

- **class SHash**
  A scalar expression.

  Represents a "#" value in a TABLE() expression.

- **class SNull**
  A scalar expression without operands.

- **class SRefAttr**
  An attribute defined as a reference to a scalar.

  The referred attribute may have values of type real, logical or string.

- **class SRefExpr**
  An expression defined as a reference to a scalar.

  The referred attribute may have values of type real, logical or string.

- **class SUnary**
  A scalar expression with one scalar operand.

- **class SValue**
  Object attribute with a constant scalar value.

- **class Scalar**
  A scalar expression.

- **struct TFunction0**
  An operand-less function returning a T.

- **struct TFunction1**
  A function of one U, returning a T.

- **struct TFunction2**
  A function of two U's returning a T.

- **inline const T* find (const T[],const string&)**
  Look up name.

  The input table is a C array of structures containing a **name** entry. The result is a pointer to one component of the array or NULL.

- **inline std::ostream& operator<< (std::ostream&,const ARefAttr<T>&)**

- **inline std::ostream& operator<< (std::ostream&,const SRefAttr<T>&)**

- **extern PtrToScalar<bool> parseBool (Statement&)**
  Parse boolean expression.

- **extern PtrToArray<bool> parseBoolArray (Statement&)**
  Parse boolean array expression.

- **extern void parseDelimiter (Statement&,char)**
  Test for one-character delimiter.

- **extern void parseDelimiter (Statement&,const char[2])**
  Test for two-character delimiter.

- **extern PlaceRep parsePlace (Statement&)**
  Parse place specification.

- **extern RangeRep parseRange (Statement&)**
  Parse range specification.

- **extern PtrToScalar<double> parseReal (Statement&)**
  Parse real expression.

- **extern PtrToArray<double> parseRealArray (Statement&)**
  Parse real array expression.

- **extern double parseRealConst (Statement&)**
  Parse real constant.

- **extern PtrToArray<double> parseRealConstArray (Statement&)**
  Parse real array constant.

- **extern SRefAttr<double>* parseReference (Statement&)**
  Parse variable reference.

- **extern string parseString (Statement&,const char[])**
  Parse string value.

  When no string is seen, a ParseError is thrown with the message given as the second argument.

- **extern std::vector<string> parseStringArray (Statement&)**
  Parse string array.

- **extern PtrToScalar<double> parseTableExpression (Statement&,const Table*)**
  Parse table expression (depends on a table's rows).

- **extern TableRowRep parseTableRow (Statement&)**
  Parse a token list (for macro argument and the like).

- **extern std::list<Token> parseTokenList (Statement&)**
  Parse a token list (for macro argument and the like).

- **extern std::vector<std::list<Token> > parseTokenListArray (Statement&)**
  Parse a token list array (for LIST commands).

2.4 namespace Options

The global MAD option flags.
This namespace contains the global option flags.

Include file:	./Utilities/Options.hh

Synopsis

Members

- **extern bool echo**
  Echo flag. (see Section 2.4.1)
- **extern bool info**
  Info flag. (see Section 2.4.1)
- **extern bool mad8**
  MAD-8 flag. (see Section 2.4.1)
- **extern Random rangen**
  Random generator. (see Section 2.4.1)
- **extern int seed**
  The current random seed. (see Section 2.4.1)
- **extern bool tfsFormat**
  The table format flag. (see Section 2.4.1)
- **extern bool trace**
  Trace flag. (see Section 2.4.1)
- **extern bool verify**
  Verify flag. (see Section 2.4.1)
- **extern bool warn**
  Warn flag. (see Section 2.4.1)

2.4.1 Detailed descriptions

Public members

- **extern bool echo**
  Echo flag.

  If true, print an input echo.
- **extern bool info**
  Info flag.

  If true, print informative messages.
- **extern bool mad8**
  MAD-8 flag.

  If true, give output in MAD-8 format.

- **extern Random rangen**
  Random generator.

  The global random generator.

- **extern int seed**
  The current random seed.

- **extern bool tfsFormat**
  The table format flag.

  If true, print tables in TFS format.

- **extern bool trace**
  Trace flag.

  If true, print CPU time before and after each command.

- **extern bool verify**
  Verify flag.

  If true, print warning about undefined variables.

- **extern bool warn**
  Warn flag.

  If true, print warning messages.

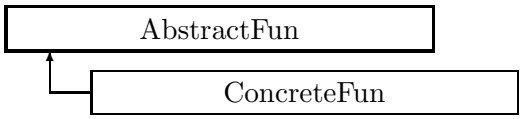


Figure 2.1: Inheritance for class AbstractFun

2.5 class AbstractFun

Abstract base for matching constraints.

Type:	abstract
Include file:	./Match/AbstractFun.hh

Synopsis (including inherited members)

Public members

- **AbstractFun ()**
  (see Section 2.5.1)
- **virtual int countConstraints ()const**
  Get number of constrained values. (see Section 2.5.1)
- **virtual void evaluate (Vector<double>&,int&)const**
  Evaluate the matching function(s). (see Section 2.5.1)
- **virtual void print (std::ostream&)const**
  Print the function name and value(s). (see Section 2.5.1)
- **virtual ~AbstractFun ()**
  (see Section 2.5.1)

2.5.1 Detailed descriptions

Public members

- **AbstractFun ()**
- **virtual int countConstraints ()const**
  Get number of constrained values.
- **virtual void evaluate (Vector<double>&,int&)const**
  Evaluate the matching function(s).

  Increment **n** for each constrained value and store the value in vector **f**.
- **virtual void print (std::ostream&)const**
  Print the function name and value(s).
- **virtual ~AbstractFun ()**

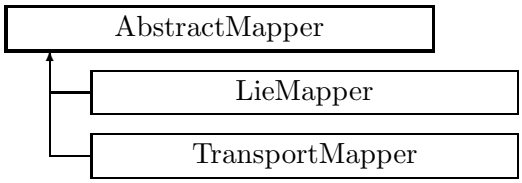


Figure 2.2: Inheritance for class AbstractMapper

2.6 class AbstractMapper

Type:	Instantiable

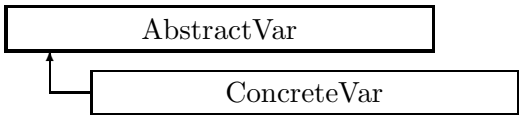


Figure 2.3: Inheritance for class AbstractVar

2.7 class AbstractVar

Abstract base for a matching variable.

Type:	abstract
Include file:	./Match/AbstractVar.hh

Synopsis (including inherited members)

Public members

- **AbstractVar (const string&)**
  Constructor. (see Section 2.7.1)
- **virtual double getExternalValue ()const**
  Get the current external parameter value. (see Section 2.7.1)
- **virtual double getInternalValue ()const**
  Get the current internal parameter value. (see Section 2.7.1)
- **virtual const string& getName ()const**
  Get the variable name. (see Section 2.7.1)
- **virtual void print (std::ostream&)const**
  Print the variable name and value. (see Section 2.7.1)
- **virtual void setExternalValue (double)**
  Set the current external parameter value. (see Section 2.7.1)
- **virtual void setInternalValue (double)**
  Set the current internal parameter value. (see Section 2.7.1)
- **virtual ~AbstractVar ()**
  (see Section 2.7.1)

Protected members

- **const string itsName**
  Name of the variable. (see Section 2.7.1)

2.7.1 Detailed descriptions

Public members

- **AbstractVar (const string&)**
  Constructor.

  Assign the variable name.

- **virtual double getExternalValue ()const**
  Get the current external parameter value.
- **virtual double getInternalValue ()const**
  Get the current internal parameter value.
- **virtual const string& getName ()const**
  Get the variable name.
- **virtual void print (std::ostream&)const**
  Print the variable name and value.
- **virtual void setExternalValue (double)**
  Set the current external parameter value.
- **virtual void setInternalValue (double)**
  Set the current internal parameter value.
- **virtual ~AbstractVar ()**

Protected members

- **const string itsName**
  Name of the variable.

2.8 class Action

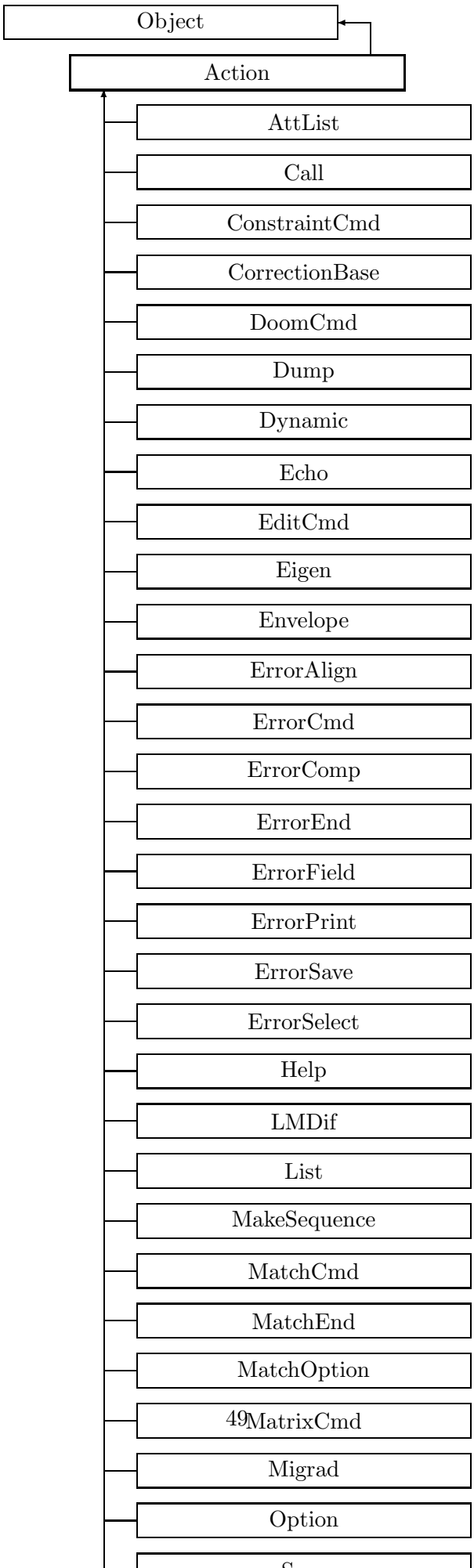
The base class for all MAD actions.
It implements the common behaviour of actions, it can also be used via dynamic casting to determine whether an object represents an action.

Type:	abstract
Superclasses:	public Object
Include file:	./AbstractObjects/Action.hh

Synopsis (including inherited members)

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)


Object
Action
AttList
Call
ConstraintCmd
CorrectionBase
DoomCmd
Dump
Dynamic
Echo
EditCmd
Eigen
Envelope
ErrorAlign
ErrorCmd
ErrorComp
ErrorEnd
ErrorField
ErrorPrint
ErrorSave
ErrorSelect
Help
LMDif
List
MakeSequence
MatchCmd
MatchEnd
MatchOption
MatrixCmd
Migrad
Option

- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)

- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Action ()**
  (see Section 2.8.1)

Protected members

- **Action (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.8.1)
- **Action (const string&,Action*)**
  Constructor for cloning. (see Section 2.8.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.8.1 Detailed descriptions

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed.

  Always return **true**.

- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "ACTION".

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always true for actions.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always true for actions.

- **virtual ~Action ()**

Protected members

- **Action (int,const char*,const char*)**
  Constructor for exemplars.

- **Action (const string&,Action*)**
  Constructor for cloning.

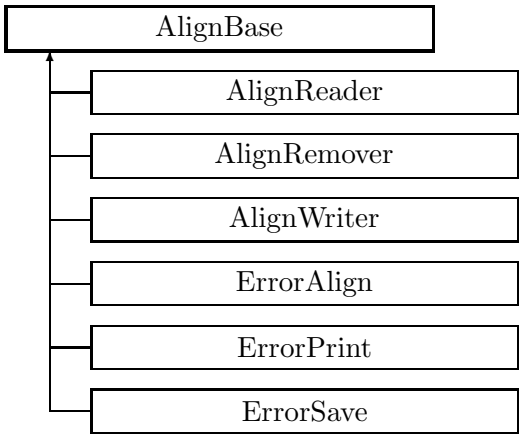


Figure 2.5: Inheritance for class AlignBase

2.9 class AlignBase

Handle alignment errors.
An abstract mixin class, used for all classes which must have access to misalignment errors for setting or retrieving.

Type:	abstract
Include file:	./Errors/AlignBase.hh

Synopsis (including inherited members)

Public members

- **AlignBase ()**
  (see Section 2.9.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Deal with misalignment. (see Section 2.9.1)
- **virtual ~AlignBase ()**
  (see Section 2.9.1)

2.9.1 Detailed descriptions

Public members

- **AlignBase ()**
- **virtual void misalignment (const AlignWrapper&,int)**
  Deal with misalignment.

  This is the interface method "mixed in".
- **virtual ~AlignBase ()**

Figure 2.6: Inheritance for class AlignHandler

2.10 class AlignHandler

Access misalignment errors.
A "Visitor" class giving access to the misalignment errors. It calls the "AlignBase" for each element which has (may have) a misalignment.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Errors/AlignHandler.hh

Synopsis (including inherited members)

Public members

- **AlignHandler (const Beamline&,AlignBase&,bool)**
  Constructor. (see Section 2.10.1)
- **virtual void visitAlignWrapper (const AlignWrapper&)**
  Apply visitor to AlignWrapper. (see Section 2.10.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply visitor to FlaggedElmPtr. (see Section 2.10.1)
- **virtual ~AlignHandler ()**
  (see Section 2.10.1)

2.10.1 Detailed descriptions

Public members

- **AlignHandler (const Beamline&,AlignBase&,bool)**
  Constructor.

  Set up the visitor to use the given beam line **bl** and to apply the command **base** to its contained elements. If **full** is true, apply to all elements, otherwise only to selected elemens.
- **virtual void visitAlignWrapper (const AlignWrapper&)**
  Apply visitor to AlignWrapper.

  Access the misalignment.
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply visitor to FlaggedElmPtr.

  Makes sure the selection flag is set.

- **virtual ~AlignHandler ()**

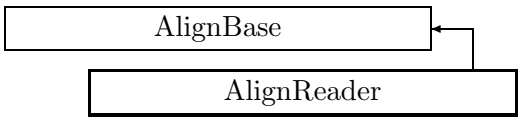


Figure 2.7: Inheritance for class AlignReader

2.11 class AlignReader

DOOM reader for reading alignment errors.
Allows retriving of misalignment errors. Can be applied via an AlignHandler to a beam line.

Type:	Instantiable
Superclasses:	public AlignBase
Include file:	./Errors/AlignReader.hh

Synopsis (including inherited members)

Public members

- **AlignReader (const string&)**
  Constructor. (see Section 2.11.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Read misalignment. (see Section 2.11.1)
- **virtual ~AlignReader ()**
  (see Section 2.11.1)

2.11.1 Detailed descriptions

Public members

- **AlignReader (const string&)**
  Constructor.

  Store **name** in the DOOM environment.
- **virtual void misalignment (const AlignWrapper&,int)**
  Read misalignment.

  The misalignment is written to the given AlignWrapper.
- **virtual ~AlignReader ()**

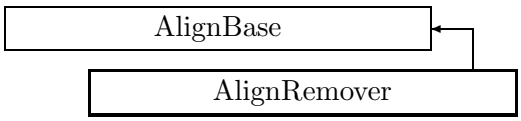


Figure 2.8: Inheritance for class AlignRemover

2.12 class AlignRemover

A visitor used to remove all misalignments from a line.
Can be applied via an AlignHandler.

Type:	Instantiable
Superclasses:	public AlignBase
Include file:	./Errors/AlignRemover.hh

Synopsis (including inherited members)

Public members

- **AlignRemover ()**
  (see Section 2.12.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Remove misalignement. (see Section 2.12.1)
- **virtual ~AlignRemover ()**
  (see Section 2.12.1)

2.12.1 Detailed descriptions

Public members

- **AlignRemover ()**
- **virtual void misalignment (const AlignWrapper&,int)**
  Remove misalignement.
- **virtual ~AlignRemover ()**

2.13 class AlignWrapper

Type:	Instantiable

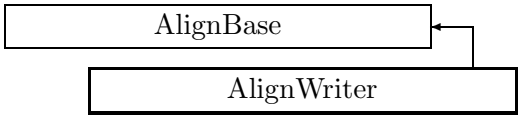


Figure 2.9: Inheritance for class AlignWriter

2.14 class AlignWriter

A DOOM writer for writing alignment errors.
Can be applied via an AlignHandler.

Type:	Instantiable
Superclasses:	public AlignBase
Include file:	./Errors/AlignWriter.hh

Synopsis (including inherited members)

Public members

- **AlignWriter (const string&)**
  Constructor. (see Section 2.14.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Write misalignment. (see Section 2.14.1)
- **virtual ~AlignWriter ()**
  (see Section 2.14.1)

2.14.1 Detailed descriptions

Public members

- **AlignWriter (const string&)**
  Constructor.

  Store **name** in the DOOM environment.
- **virtual void misalignment (const AlignWrapper&,int)**
  Write misalignment.

  The misalignment is read from the given AlignWrapper.
- **virtual ~AlignWriter ()**

2.15 template class Array1D <class>

Type:	Instantiable

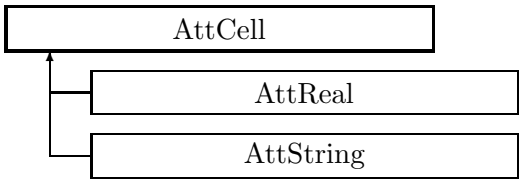


Figure 2.10: Inheritance for class AttCell

2.16 class AttCell

The abstract base class for attribute cells.
Representation of a table cell for ATTLIST command.

Type:	abstract
Include file:	./Elements/AttCell.hh

Synopsis (including inherited members)

Public members

- **AttCell ()**
  (see Section 2.16.1)
- **virtual void clearValue ()**
  Clear the value. (see Section 2.16.1)
- **virtual void printFormat (std::ostream&)const**
  Print the attribute format. (see Section 2.16.1)
- **virtual void printValue (std::ostream&)const**
  Print the attribute value. (see Section 2.16.1)
- **virtual void setReal (double)**
  Store the value. (see Section 2.16.1)
- **virtual void setString (const std::string&)**
  Store the value. (see Section 2.16.1)
- **virtual ~AttCell ()**
  (see Section 2.16.1)

2.16.1 Detailed descriptions

Public members

- **AttCell ()**

- **virtual void clearValue ()**
  Clear the value.

  Reset the value to undefined.

- **virtual void printFormat (std::ostream&)const**
  Print the attribute format.

  Print the format string in C style.

- **virtual void printValue (std::ostream&)const**
  Print the attribute value.

  According to the format string.

- **virtual void setReal (double)**
  Store the value.

  Set the cell value to the given double.

- **virtual void setString (const std::string&)**
  Store the value.

  Set the cell value to the given string.

- **virtual ~AttCell ()**

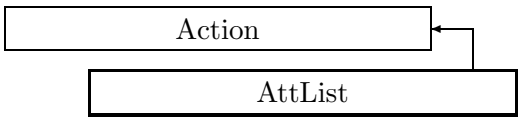


Figure 2.11: Inheritance for class AttList

2.17 class AttList

The ATTLIST command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Tables/AttList.hh

Synopsis (including inherited members)

Public members

- **AttList ()**
  Exemplar constructor. (see Section 2.17.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual AttList* clone (const string&)**
  Make clone. (see Section 2.17.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.17.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~AttList ()**
  (see Section 2.17.1)

2.17.1 Detailed descriptions

Public members

- **AttList ()**
  Exemplar constructor.
- **virtual AttList* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~AttList ()**

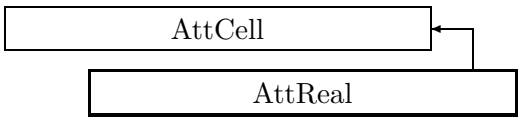


Figure 2.12: Inheritance for class AttReal

2.18 class AttReal

The class for attribute cells with a real value.

Type:	Instantiable
Superclasses:	public AttCell
Include file:	./Elements/AttCell.hh

Synopsis (including inherited members)

Public members

- **AttReal ()**
  (see Section 2.18.1)
- **virtual void clearValue ()**
  Clear the value. (see Section 2.18.1)
- **virtual void printFormat (std::ostream&)const**
  Print the attribute format. (see Section 2.18.1)
- **virtual void printValue (std::ostream&)const**
  Print the attribute value. (see Section 2.18.1)
- **virtual void setReal (double)**
  Store the value. (see Section 2.18.1)
- **virtual void setString (const std::string&)**
  Store the value. (see Section 2.16.1)
- **virtual ~AttReal ()**
  (see Section 2.18.1)

2.18.1 Detailed descriptions

Public members

- **AttReal ()**

- **virtual void clearValue ()**
  Clear the value.
- **virtual void printFormat (std::ostream&)const**
  Print the attribute format.

  Prints ”

- **virtual void printValue (std::ostream&)const**
  Print the attribute value.
- **virtual void setReal (double)**
  Store the value.
- **virtual ~AttReal ()**

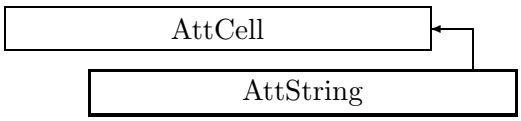


Figure 2.13: Inheritance for class AttString

2.19 class AttString

The class for attribute cells with a string value.

Type:	Instantiable
Superclasses:	public AttCell
Include file:	./Elements/AttCell.hh

Synopsis (including inherited members)

Public members

- **AttString ()**
  (see Section 2.19.1)
- **virtual void clearValue ()**
  Clear the value. (see Section 2.19.1)
- **virtual void printFormat (std::ostream&)const**
  Print the attribute format. (see Section 2.19.1)
- **virtual void printValue (std::ostream&)const**
  Print the attribute value. (see Section 2.19.1)
- **virtual void setReal (double)**
  Store the value. (see Section 2.16.1)
- **virtual void setString (const std::string&)**
  Store the value. (see Section 2.19.1)
- **virtual ~AttString ()**
  (see Section 2.19.1)

2.19.1 Detailed descriptions

Public members

- **AttString ()**

- **virtual void clearValue ()**
  Clear the value.
- **virtual void printFormat (std::ostream&)const**
  Print the attribute format.

  Prints ”

- **virtual void printValue (std::ostream&)const**
  Print the attribute value.
- **virtual void setString (const std::string&)**
  Store the value.
- **virtual ~AttString ()**

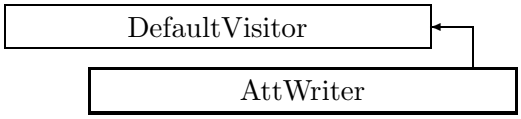


Figure 2.14: Inheritance for class AttWriter

2.20 class AttWriter

The worker class for ATTLIST commands.
A "Visitor" which looks at all elements in turn. For each element it asks to store all defined attributes in a registry, and then it requests the desired values from that registry to build a print line.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Tables/AttWriter.hh

Synopsis (including inherited members)

Public members

- **AttWriter (const Beamline&,std::ostream&,MadElement::ValueFlag,const std::vector<AttCell**
  Constructor. (see Section 2.20.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the algorithm to a FlaggedElmPtr. (see Section 2.20.1)
- **virtual ~AttWriter ()**
  (see Section 2.20.1)

2.20.1 Detailed descriptions

Public members

- **AttWriter (const Beamline&,std::ostream&,MadElement::ValueFlag,const std::vector<AttCell**
  Constructor.
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the algorithm to a FlaggedElmPtr.
- **virtual ~AttWriter ()**

2.21 class Attribute

A representation of an Object attribute.
Contains a pointer to the polymorphic attribute value (derived from **AttributeBase**, and a pointer to the polymorphic attribute parser (derived from **AttributeHandler**).

Type:	Instantiable
Include file:	./AbstractObjects/Attribute.hh

Synopsis (including inherited members)

Public members

- **Attribute ()**
  Default constructor. (see Section 2.21.1)
- **Attribute (const Attribute&)**
  Copy constructor. (see Section 2.21.1)
- **Attribute (AttributeHandler*,AttributeBase*)**
  Constructor defining a parser and an optional value. (see Section 2.21.1)
- **void doomGet (const DoomReader&,int)**
  Read attribute from the DOOM data base. (see Section 2.21.1)
- **void doomPut (DoomWriter&,int)const**
  Write attribute to the DOOM data base. (see Section 2.21.1)
- **AttributeBase& getBase ()const**
  Return reference to polymorphic value. (see Section 2.21.1)
- **AttributeHandler& getHandler ()const**
  Return a reference to the parser. (see Section 2.21.1)
- **const string& getHelp ()const**
  Return the help string. (see Section 2.21.1)
- **string getImage ()const**
  Return printable representation. (see Section 2.21.1)
- **const string& getName ()const**
  Return the attribute name. (see Section 2.21.1)
- **const string& getType ()const**
  Return the attribute type. (see Section 2.21.1)
- **bool isDeferred ()const**
  Return **deferred** flag. (see Section 2.21.1)
- **bool isExpression ()const**
  Test for expression. (see Section 2.21.1)
- **bool isReadOnly ()const**
  Test for read only. (see Section 2.21.1)

- **operator bool ()const**
  Test for valid pointer. (see Section 2.21.1)
- **const Attribute& operator= (const Attribute&)**
  (see Section 2.21.1)
- **void parse (Statement&,bool)**
  Parse attribute. (see Section 2.21.1)
- **void parseComponent (Statement&,bool,int)**
  Parse array component. (see Section 2.21.1)
- **void print (std::ostream&,int&)const**
  Print attribute. (see Section 2.21.1)
- **void set (AttributeBase*)**
  Define new value. (see Section 2.21.1)
- **void setDefault ()**
  Assign default value. (see Section 2.21.1)
- **void setDeferred (bool)**
  Set read-only flag. (see Section 2.21.1)
- **void setReadOnly (bool)**
  Set read-only flag. (see Section 2.21.1)
- **~Attribute ()**
  (see Section 2.21.1)

2.21.1 Detailed descriptions

Public members

- **Attribute ()**
  Default constructor.

  Leaves both pointers **NULL**. An object constructed by the default constructor must be initialized using assignment before it can be used. Failing to do so may cause a program crash.
- **Attribute (const Attribute&)**
  Copy constructor.

  Both value and parser are shared with **rhs**.
- **Attribute (AttributeHandler*,AttributeBase*)**
  Constructor defining a parser and an optional value.

  The default value is "undefined".
- **void doomGet (const DoomReader&,int)**
  Read attribute from the DOOM data base.

  Use the contained parser to read the attribute.

- **void doomPut (DoomWriter&,int)const**
  Write attribute to the DOOM data base.

  Use the contained parser to write the attribute.

- **AttributeBase& getBase ()const**
  Return reference to polymorphic value.

- **AttributeHandler& getHandler ()const**
  Return a reference to the parser.

- **const string& getHelp ()const**
  Return the help string.

  This string is stored in the parser object.

- **string getImage ()const**
  Return printable representation.

  This string duplicates the input expression defining the attribute.

- **const string& getName ()const**
  Return the attribute name.

  This string is stored in the parser object.

- **const string& getType ()const**
  Return the attribute type.

  This string ("real", "logical", etc.) is stored in the parser object.

- **bool isDeferred ()const**
  Return deferred flag.

  If this flag is set, any expression is re-evaluated each time the value is fetched. Normally attribute evaluation is not deferred, i. e. any expression is evaluated only the first time the value is fetched after a new definition has been read. See **Expressions::ADeferred** and **Expressions::SDeferred** for information.

- **bool isExpression ()const**
  Test for expression.

  Return true, if the attribute is defined as an expression.

- **bool isReadOnly ()const**
  Test for read only.

  Return true, if the attribute cannot be redefined by the user.

- **operator bool ()const**
  Test for valid pointer.

  Returns true, when the value is defined (non-null pointer).

- **const Attribute& operator= (const Attribute&)**

- **void parse (Statement&,bool)**
  Parse attribute.

  Use the contained parser to parse a new value for the attribute.

- **void parseComponent (Statement&,bool,int)**
  Parse array component.

  Use the contained parser to parse a new value for an existing vector component. If the attribute is a scalar value, or if the component does not exist, this method throws **MadException**.

- **void print (std::ostream&,int&)const**
  Print attribute.

  Print the attribute name, followed by an equals sign and its value.

- **void set (AttributeBase*)**
  Define new value.

  Assign a new value. The value must be compatible with the parser type, otherwise a **MadException** is thrown.

- **void setDefault ()**
  Assign default value.

  Set the attribute value to the default value stored in the parser. If no default value exists, set it to "undefined".

- **void setDeferred (bool)**
  Set read-only flag.

  If this flag is set, the attribute cannot be redefined by the user. See **Expressions::ADeferred** and **Expressions::SDeferred** for information.

- **void setReadOnly (bool)**
  Set read-only flag.

  Set or reset the attribute's read-only flag.

- **~Attribute ()**

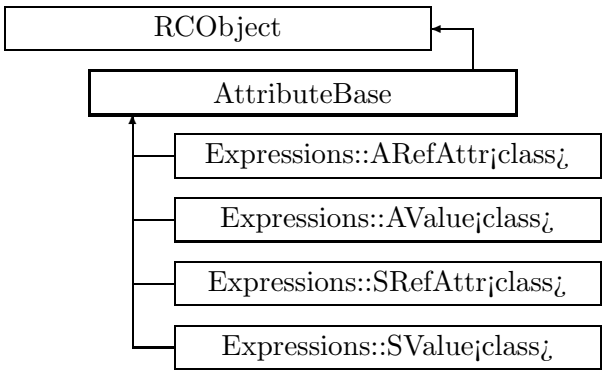


Figure 2.15: Inheritance for class AttributeBase

2.22 class AttributeBase

Abstract base class for attribute values of different types.

Type:	abstract
Superclasses:	public RCObject
Include file:	./AbstractObjects/AttributeBase.hh

Synopsis (including inherited members)

Public members

- **AttributeBase ()**
  (see Section 2.22.1)
- **virtual AttributeBase* clone ()const**
  Make clone. (see Section 2.22.1)
- **string getImage ()const**
  Convert to string. (see Section 2.22.1)
- **virtual void invalidate ()**
  Force re-evaluation. (see Section 2.22.1)
- **virtual bool isExpression ()const**
  Test for expression. (see Section 2.22.1)
- **virtual std::ostream& print (std::ostream&)const**
  Print value. (see Section 2.22.1)
- **virtual ~AttributeBase ()**
  (see Section 2.22.1)

2.22.1 Detailed descriptions

Public members

- **AttributeBase ()**

- **virtual AttributeBase* clone ()const**
  Make clone.

  Construct an exact copy of the value.

- **string getImage ()const**
  Convert to string.

  Uses **print()** to convert the input expression for the attribute to a printable representation.

- **virtual void invalidate ()**
  Force re-evaluation.

  Set an internal flag so as to force re-evaluation of any expression when the value is referred next time.

- **virtual bool isExpression ()const**
  Test for expression.

  Return **true** if the value is an expression.

- **virtual std::ostream& print (std::ostream&)const**
  Print value.

  Print the value on the given output stream. The result allows reconstruction of any expression.

- **virtual ~AttributeBase ()**

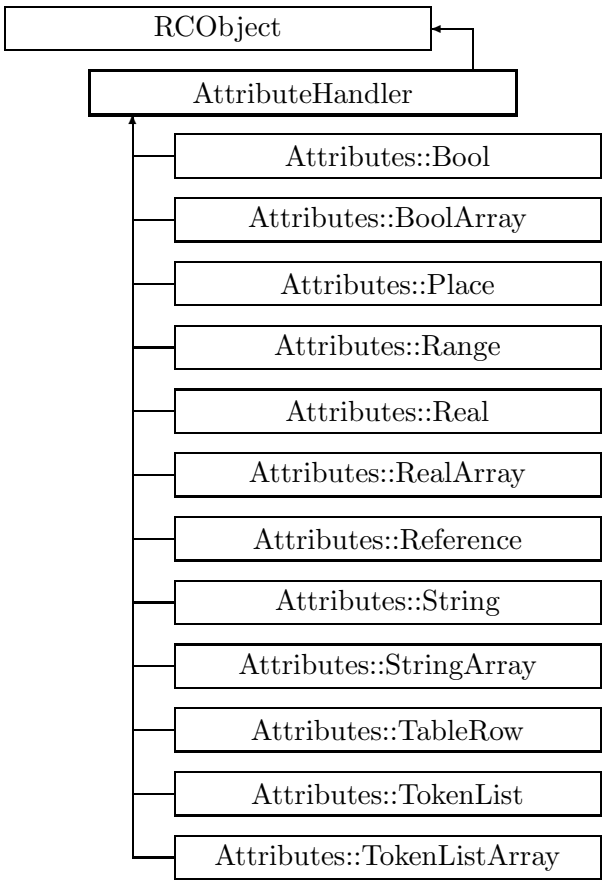


Figure 2.16: Inheritance for class AttributeHandler

2.23 class AttributeHandler

Abstract base class for attribute parsers.
An attribute parser defines the data type for an attribute. It is used to parse and print the attribute, to read the attribute from the DOOM data base, and to write it to that data base. It contains contains the name and help text for the attribute, and sometimes a default value for the attribute.
When "is_readonly" is true, the attribute cannot be redefined by the user.
When "is_deferred" is true, the attribute must be re-evaluated whenever it is used. This is the case for random error values. When "is_deferred" is false, any expression for the attribute is cached. It is re-evaluated only when any other definition has changed.

Type:	abstract
Superclasses:	public RCObject
Include file:	./AbstractObjects/AttributeHandler.hh

Synopsis (including inherited members)

Public members

- **AttributeHandler (const string&,const string&,AttributeBase*)**
  Constructor. (see Section 2.23.1)

- **virtual AttributeHandler* clone ()const**
  Make clone. (see Section 2.23.1)
- **virtual void doomGet (Attribute&,const DoomReader&,int)const**
  Read attribute from the DOOM data base. (see Section 2.23.1)
- **virtual void doomPut (const Attribute&,DoomWriter&,int)const**
  Write the attribute a to the DOOM data base. (see Section 2.23.1)
- **virtual AttributeBase* getDefault ()const**
  Return default value. (see Section 2.23.1)
- **virtual const string& getHelp ()const**
  Return help string. (see Section 2.23.1)
- **virtual const string& getName ()const**
  Return attribute name. (see Section 2.23.1)
- **virtual const string& getType ()const**
  Return attribute type. (see Section 2.23.1)
- **bool isDeferred ()const**
  Return defer flag. (see Section 2.23.1)
- **bool isReadOnly ()const**
  Return read-only flag. (see Section 2.23.1)
- **virtual void parse (Attribute&,Statement&,bool)const**
  Parse new value. (see Section 2.23.1)
- **virtual void parseComponent (Attribute&,Statement&,bool,int)const**
  Parse component value. (see Section 2.23.1)
- **void setDeferred (bool)**
  Set or reset defer flag. (see Section 2.23.1)
- **void setReadOnly (bool)**
  Set or reset read-only flag. (see Section 2.23.1)
- **virtual ~AttributeHandler ()**
  (see Section 2.23.1)

Protected members

- **bool is_deferred**
  Defer flag. (see Section 2.23.1)
- **bool is_readonly**
  Read-only flag. (see Section 2.23.1)
- **Pointer<AttributeBase> itsDefault**
  Default value. (see Section 2.23.1)

- **const string itsHelp**
  Help text. (see Section 2.23.1)

- **const string itsName**
  Attribute name. (see Section 2.23.1)

2.23.1 Detailed descriptions

Public members

- **AttributeHandler (const string&,const string&,AttributeBase*)**
  Constructor.

  Assigns the attribute name **name** and the help text **help**, as well as a possible default value **def**for the attribute.

- **virtual AttributeHandler* clone ()const**
  Make clone.

  Attribute handlers are always shared, so this method should never be called. It exists only to fulfill the requirements of the class **Pointer**.

- **virtual void doomGet (Attribute&,const DoomReader&,int)const**
  Read attribute from the DOOM data base.

  Uses the DoomReader **r** for the object being read, and the position **i** within this reader. Called by **Attribute::doomGet()**

- **virtual void doomPut (const Attribute&,DoomWriter&,int)const**
  Write the attribute a to the DOOM data base.

  Uses the DoomWriter **w** for the object being read, and the position **i** within this writer. Called by **Attribute::doomPut()**

- **virtual AttributeBase* getDefault ()const**
  Return default value.

  Return the default value stored in this parser.

- **virtual const string& getHelp ()const**
  Return help string.

- **virtual const string& getName ()const**
  Return attribute name.

- **virtual const string& getType ()const**
  Return attribute type.

  Return a string describing the attribute type ("logical", "real", etc.).

- **bool isDeferred ()const**
  Return defer flag.

  True, if any expression evaluation is to be deferred. See **Expressions::ADeferred** and **Expressions::SDeferred** for details.

- **bool isReadOnly ()const**
  Return read-only flag.

  If **parse** is called with this flag set, then **MadException** is thrown.

- **virtual void parse (Attribute&,Statement&,bool)const**
  Parse new value.

  Parse value from the statement **s** and assign it to the attribute **a**.

- **virtual void parseComponent (Attribute&,Statement&,bool,int)const**
  Parse component value.

  Parse value from the statement **s** and assign it to the attribute **a**, indexed by **i**. The default version assumes that the value is scalar, and it throws **MadException**.

- **void setDeferred (bool)**
  Set or reset defer flag.

  If the flag is set, expressions are evaluated only when the value is fetched.

- **void setReadOnly (bool)**
  Set or reset read-only flag.

  If **parse** is called with the flag set, then **MadException** is thrown.

- **virtual ~AttributeHandler ()**

Protected members

- **bool is_deferred**
  Defer flag.

- **bool is_readonly**
  Read-only flag.

- **Pointer<AttributeBase> itsDefault**
  Default value.

- **const string itsHelp**
  Help text.

- **const string itsName**
  Attribute name.

2.24 class BMultipoleField

Type:	Instantiable

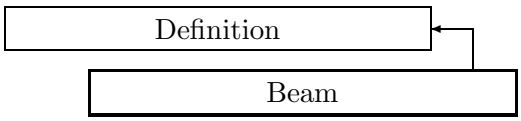


Figure 2.17: Inheritance for class Beam

2.25 class Beam

The BEAM definition.

Type:	Instantiable
Superclasses:	public Definition
Include file:	./Structure/Beam.hh

Synopsis (including inherited members)

Public members

- **Beam ()**
  Exemplar constructor. (see Section 2.25.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.25.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Beam* clone (const string&)**
  Make clone. (see Section 2.25.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Check the BEAM data. (see Section 2.25.1)
- **static Beam* find (const string&)**
  Find named BEAM. (see Section 2.25.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.40.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **double getET ()const**
  Return emittance for mode 3. (see Section 2.25.1)
- **double getEX ()const**
  Return emittance for mode 1. (see Section 2.25.1)
- **double getEY ()const**
  Return emittance for mode 2. (see Section 2.25.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const PartData& getReference ()const**
  Return the embedded CLASSIC PartData. (see Section 2.25.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)

- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setET (double)**
  Store emittance for mode 3. (see Section 2.25.1)
- **void setEX (double)**
  Store emittance for mode 1. (see Section 2.25.1)
- **void setEY (double)**
  Store emittance for mode 2. (see Section 2.25.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.40.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.40.1)
- **void tfsDescriptors (std::ostream&)const**
  Print the TFS descriptors for the beam. (see Section 2.25.1)

- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the BEAM data. (see Section 2.25.1)
- **virtual ~Beam ()**
  (see Section 2.25.1)

2.25.1 Detailed descriptions

Public members

- **Beam ()**
  Exemplar constructor.
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed.

  Can replace only by another BEAM.
- **virtual Beam* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Check the BEAM data.
- **static Beam* find (const string&)**
  Find named BEAM.
- **double getET ()const**
  Return emittance for mode 3.
- **double getEX ()const**
  Return emittance for mode 1.
- **double getEY ()const**
  Return emittance for mode 2.
- **const PartData& getReference ()const**
  Return the embedded CLASSIC PartData.
- **void setET (double)**
  Store emittance for mode 3.
- **void setEX (double)**
  Store emittance for mode 1.
- **void setEY (double)**
  Store emittance for mode 2.
- **void tfsDescriptors (std::ostream&)const**
  Print the TFS descriptors for the beam.

- **virtual void update ()**
  Update the BEAM data.
- **virtual ~Beam ()**

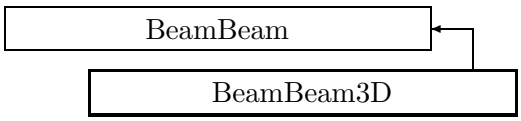


Figure 2.18: Inheritance for class BeamBeam3D

2.26 class BeamBeam3D

A concrete representation for a beam-beam interaction.
The "strong" bunch has a Gaussian distribution of the form
`N(x,y,s) = c * exp(- transpose(z - delta) sigma**(-1) (z - delta)),`
with the definitions

z: position vector in space,

delta: centroid of strong bunch in global reference system,

sigma: "beam" matrix in the TRANSPORT sense,

c: a normalising factor, such that the total charge is Q,

sigma(i,i): standard deviation sigma(i) in direction i,

r(i,j): = sigma(i,j) / sqrt(sigma(i,i)*sigma(i,j)), correlations between phase space coordinates i and j.

***** MISSING ***** This class needs further work, based on MAD-8.

Type:	Instantiable
Superclasses:	public BeamBeam
Include file:	./Algorithms/BeamBeam3D.hh

Synopsis (including inherited members)

Public members

- **BeamBeam3D (const string&)**
  Constructor with given name. (see Section 2.26.1)
- **BeamBeam3D ()**
  (see Section 2.26.1)
- **BeamBeam3D (const BeamBeam3D&)**
  (see Section 2.26.1)
- **struct Beta**
  Collection of lattice functions and emittances. (see Section 2.26.1)
- **virtual void accept (BeamlineVisitor&)const**
  Apply visitor to BeamBeam. (see Section 2.26.1)
- **virtual ElementBase* clone ()const**
  Return exact copy of the element. (see Section 2.26.1)

- **virtual double getBunchCharge ()const**
  Get the bunch charge. (see Section 2.26.1)
- **virtual const Vector3D& getBunchDisplacement ()const**
  Get displacement vector for the strong bunch. (see Section 2.26.1)
- **virtual const Matrix3D& getBunchMoment ()const**
  Get the moment matrix for the strong bunch. (see Section 2.26.1)
- **virtual Channel* getChannel (const string&)**
  Return a channel to an attribute. (see Section 2.26.1)
- **virtual NullField& getField ()**
  Return the zero electromagnetic field. (see Section 2.26.1)
- **virtual const NullField& getField ()const**
  Return the zero electromagnetic field. Version for const object. (see Section 2.26.1)
- **virtual NullGeometry& getGeometry ()**
  Return the null geometry. (see Section 2.26.1)
- **virtual const NullGeometry& getGeometry ()const**
  Return the null geometry. Version for const object. (see Section 2.26.1)
- **virtual ElementImage* getImage ()const**
  Return an image of the element. (see Section 2.26.1)
- **virtual const string& getType ()const**
  Return type name string. (see Section 2.26.1)
- **void setBeamBeamParameter (double)**
  Set the proportionality factor. (see Section 2.26.1)
- **void setBeamDescription (const Vector3D&,const Beta&)**
  Store the description of the strong beam. (see Section 2.26.1)
- **void setCrossingAngle (double)**
  Set the crossing angle. (see Section 2.26.1)
- **void setErrorFunctionPointer (std::complex<double>*(std::complex<double>))**
  Set the pointer to the complex error function. (see Section 2.26.1)
- **void setSlices (int)**
  Select number of slices for strong beam. (see Section 2.26.1)
- **virtual void trackBunch (PartBunch&,const PartData&,bool,bool)const**
  Track a particle bunch. (see Section 2.26.1)
- **virtual void trackMap (Map&,const PartData&,bool,bool)const**
  Track a transfer map. (see Section 2.26.1)
- **virtual ~BeamBeam3D ()**
  (see Section 2.26.1)

2.26.1 Detailed descriptions

Public members

- **BeamBeam3D (const string&)**
  Constructor with given name.
- **BeamBeam3D ()**
- **BeamBeam3D (const BeamBeam3D&)**
- **struct Beta**
  Collection of lattice functions and emittances.
- **virtual void accept (BeamlineVisitor&)const**
  Apply visitor to BeamBeam.
- **virtual ElementBase* clone ()const**
  Return exact copy of the element.
- **virtual double getBunchCharge ()const**
  Get the bunch charge.

  Return number of particles times the particle charge in the strong bunch. Units are proton charges.
- **virtual const Vector3D& getBunchDisplacement ()const**
  Get displacement vector for the strong bunch.

  Return the displacement in metres.
- **virtual const Matrix3D& getBunchMoment ()const**
  Get the moment matrix for the strong bunch.

  Return matrix of second momenta. Units are square metres.
- **virtual Channel* getChannel (const string&)**
  Return a channel to an attribute.
- **virtual NullField& getField ()**
  Return the zero electromagnetic field.
- **virtual const NullField& getField ()const**
  Return the zero electromagnetic field. Version for const object.
- **virtual NullGeometry& getGeometry ()**
  Return the null geometry.
- **virtual const NullGeometry& getGeometry ()const**
  Return the null geometry. Version for const object.
- **virtual ElementImage* getImage ()const**
  Return an image of the element.

- **virtual const string& getType ()const**
  Return type name string.

- **void setBeamBeamParameter (double)**
  Set the proportionality factor.

- **void setBeamDescription (const Vector3D&,const Beta&)**
  Store the description of the strong beam.

  disp is the displacement of the closed orbit of the strong beam. **latFun** is the collection of lattice functions and emittances of the strong beam.

- **void setCrossingAngle (double)**
  Set the crossing angle.

- **void setErrorFunctionPointer (std::complex<double>*(std::complex<double>))**
  Set the pointer to the complex error function.

  Changing this pointer allows use of a different, potentially faster algorithm for determination of the kick.

- **void setSlices (int)**
  Select number of slices for strong beam.

- **virtual void trackBunch (PartBunch&,const PartData&,bool,bool)const**
  Track a particle bunch.

- **virtual void trackMap (Map&,const PartData&,bool,bool)const**
  Track a transfer map.

- **virtual ~BeamBeam3D ()**

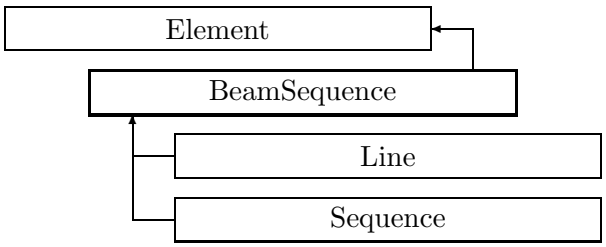


Figure 2.19: Inheritance for class BeamSequence

2.27 class BeamSequence

The base class for all MAD beam lines and sequences.
It implements the common behaviour of sequences, it can also be used via dynamic casting to determine whether an object represents a sequence.

Type:	abstract
Superclasses:	public Element
Include file:	./AbstractObjects/BeamSequence.hh

Synopsis (including inherited members)

Public members

- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **virtual BeamSequence* copy (const string&)**
  Make complete copy. (see Section 2.27.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Beamline* fetchLine ()const**
  Return the embedded CLASSIC beam line. (see Section 2.27.1)

- **static BeamSequence* find (const string&)**
  Find a BeamSequence by name. (see Section 2.27.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.27.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.64.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)

- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)

- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~BeamSequence ()**
  (see Section 2.27.1)

Protected members

- **BeamSequence (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.27.1)
- **BeamSequence (const string&,BeamSequence*)**
  Constructor for clones. (see Section 2.27.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.27.1 Detailed descriptions

Public members

- **virtual BeamSequence* copy (const string&)**
  Make complete copy.

  Copy also the line list.
- **virtual Beamline* fetchLine ()const**
  Return the embedded CLASSIC beam line.

  The result it the ideal line.
- **static BeamSequence* find (const string&)**
  Find a BeamSequence by name.
- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "SEQUENCE".
- **virtual ~BeamSequence ()**

Protected members

- **BeamSequence (int,const char*,const char*)**
  Constructor for exemplars.

- **BeamSequence (const string&,BeamSequence*)**
  Constructor for clones.

  The clone will be **empty**. It has to be filled in by the corresponding parser.

2.28 class Beamline

Type:	Instantiable

2.29 class BeamlineVisitor

Type:	Instantiable

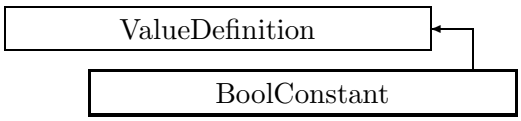


Figure 2.20: Inheritance for class BoolConstant

2.30 class BoolConstant

The BOOL CONSTANT definition.

Type:	Instantiable
Superclasses:	public ValueDefinition
Include file:	./ValueDefinitions/BoolConstant.hh

Synopsis (including inherited members)

Public members

- **BoolConstant ()**
  Exemplar constructor. (see Section 2.30.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.30.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual BoolConstant* clone (const string&)**
  Make clone. (see Section 2.30.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read constant from DOOM data base. (see Section 2.30.1)
- **virtual void doomPut (DoomWriter&)const**
  Write constant to DOOM data base. (see Section 2.30.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual bool getBool ()const**
  Return value. (see Section 2.30.1)

- **virtual bool getBoolComponent (int)const**
  Return indexed logical value. (see Section 2.208.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.208.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual double getReal ()const**
  Return real value. (see Section 2.208.1)
- **virtual double getRealComponent (int)const**
  Return indexed real value. (see Section 2.208.1)
- **virtual string getString ()const**
  Return string value. (see Section 2.208.1)
- **virtual string getStringComponent (int)const**
  Return indexed string value. (see Section 2.208.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the constant. (see Section 2.30.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.208.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.208.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **Attribute& value ()**
  Return the attribute representing the value of the definition. (see Section 2.208.1)

- **const Attribute& value ()const**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **virtual ~BoolConstant ()**
  (see Section 2.30.1)

2.30.1 Detailed descriptions

Public members

- **BoolConstant ()**
  Exemplar constructor.
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced.

  Always false for constants.
- **virtual BoolConstant* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read constant from DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write constant to DOOM data base.
- **virtual bool getBool ()const**
  Return value.
- **virtual void print (std::ostream&)const**
  Print the constant.
- **virtual ~BoolConstant ()**

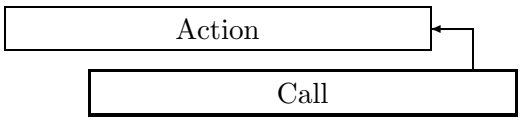


Figure 2.21: Inheritance for class Call

2.31 class Call

The CALL command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Call.hh

Synopsis (including inherited members)

Public members

- **Call ()**
  Exemplar constructor. (see Section 2.31.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Call* clone (const string&)**
  Make clone. (see Section 2.31.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.31.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.31.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Call ()**
  (see Section 2.31.1)

2.31.1 Detailed descriptions

Public members

- **Call ()**
  Exemplar constructor.
- **virtual Call* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Call ()**

2.32 class Collimator

Type:	Instantiable

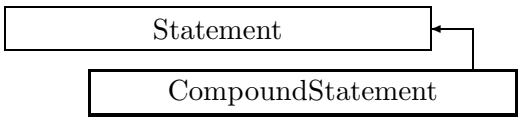


Figure 2.22: Inheritance for class CompoundStatement

2.33 class CompoundStatement

Compound statement.
A statement of the form " <statement>; <statement>; " (may be used for FOR, IF, or WHILE blocks).

Type:	Instantiable
Superclasses:	public Statement
Include file:	./MadParser/CompoundStatement.hh

Synopsis (including inherited members)

Public members

- **CompoundStatement (TokenStream&)**
  Constructor. (see Section 2.33.1)
- **virtual void execute (const Parser&)**
  Execute. (see Section 2.33.1)
- **virtual ~CompoundStatement ()**
  (see Section 2.33.1)

2.33.1 Detailed descriptions

Public members

- **CompoundStatement (TokenStream&)**
  Constructor.

  Parse the statement on the given token stream.
- **virtual void execute (const Parser&)**
  Execute.

  Use the given parser to execute the contained statements.
- **virtual ~CompoundStatement ()**

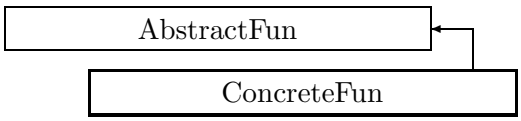


Figure 2.23: Inheritance for class ConcreteFun

2.34 class ConcreteFun

Concrete class class for a single matching constraints or for an array of matching constraints.

Type:	Instantiable
Superclasses:	public AbstractFun
Include file:	./Match/ConcreteFun.hh

Synopsis (including inherited members)

Public members

- **ConcreteFun (Attribute&,int,Attribute&,Attribute&)**
  Constructor. (see Section 2.34.1)
- **virtual int countConstraints ()const**
  Get the number of constrained values. (see Section 2.34.1)
- **virtual void evaluate (Vector<double>&,int&)const**
  Evaluate the matching function(s). (see Section 2.34.1)
- **virtual void print (std::ostream&)const**
  Print the function name and value(s). (see Section 2.34.1)
- **virtual ~ConcreteFun ()**
  (see Section 2.34.1)

2.34.1 Detailed descriptions

Public members

- **ConcreteFun (Attribute&,int,Attribute&,Attribute&)**
  Constructor.

  Uses the following arguments:

  The left-hand side(s) for the constraint.

  A code for the type of constraint.

  The right-hand side(s) for the constraint.

  The weight(s) for the constraint.
- **virtual int countConstraints ()const**
  Get the number of constrained values.

- **virtual void evaluate (Vector<double>&,int&)const**
  Evaluate the matching function(s).

  Increment **n** for each constrained value and store the value in vector **f**.

- **virtual void print (std::ostream&)const**
  Print the function name and value(s).

- **virtual ~ConcreteFun ()**

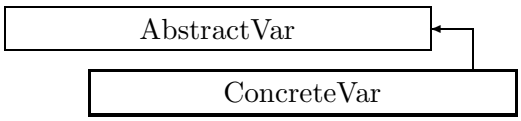


Figure 2.24: Inheritance for class ConcreteVar

2.35 class ConcreteVar

Concrete class for a matching variable.
Implements the setting and retrieving of the value in the system to be adjusted.

Type:	Instantiable
Superclasses:	public AbstractVar
Include file:	./Match/ConcreteVar.hh

Synopsis (including inherited members)

Public members

- **ConcreteVar (const string&,Attribute&,int,double[4])**
  Constructor. (see Section 2.35.1)
- **virtual double getExternalValue ()const**
  Get the current external parameter value. (see Section 2.35.1)
- **virtual double getInternalValue ()const**
  Get the current internal parameter value. (see Section 2.35.1)
- **virtual const string& getName ()const**
  Get the variable name. (see Section 2.7.1)
- **virtual void print (std::ostream&)const**
  Print the variable name and value. (see Section 2.35.1)
- **virtual void setExternalValue (double)**
  Set the current external parameter value. (see Section 2.35.1)
- **virtual void setInternalValue (double)**
  Set the current internal parameter value. (see Section 2.35.1)
- **virtual ~ConcreteVar ()**
  (see Section 2.35.1)

2.35.1 Detailed descriptions

Public members

- **ConcreteVar (const string&,Attribute&,int,double[4])**
  Constructor.

  Uses the following arguments:

  The variable name.

An attribute containing a reference to thevalue to be adjusted.

A code for limit values: 0=no limit, 1=lower, 2=upper, 3=both.

Value, lower, upper, and step.

- **virtual double getExternalValue ()const**
  Get the current external parameter value.

  The external value should be consistent with the given limits.

- **virtual double getInternalValue ()const**
  Get the current internal parameter value.

  The internal value is unlimited, it maps to the external value so as to keep the latter constrained.

- **virtual void print (std::ostream&)const**
  Print the variable name and value.

- **virtual void setExternalValue (double)**
  Set the current external parameter value.

  The external value should be consistent with the given limits.

- **virtual void setInternalValue (double)**
  Set the current internal parameter value.

  The internal value is unlimited, it maps to the external value so as to keep the latter constrained.

- **virtual ~ConcreteVar ()**

2.36 class ConstChannel

Type:	Instantiable

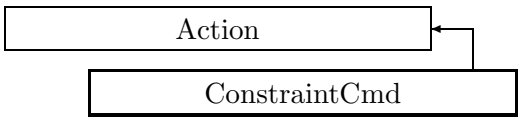


Figure 2.25: Inheritance for class ConstraintCmd

2.37 class ConstraintCmd

The CONSTRAINT command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/ConstraintCmd.hh

Synopsis (including inherited members)

Public members

- **ConstraintCmd ()**
  Exemplar constructor. (see Section 2.37.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ConstraintCmd* clone (const string&)**
  Make clone. (see Section 2.37.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.37.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the command. (see Section 2.37.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the command. (see Section 2.37.1)
- **virtual void printHelp (ostream&)const**
  Print help for the command. (see Section 2.37.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)

- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ConstraintCmd ()**
  (see Section 2.37.1)

2.37.1 Detailed descriptions

Public members

- **ConstraintCmd ()**
  Exemplar constructor.
- **virtual ConstraintCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse the command.

- **virtual void print (std::ostream&)const**
  Print the command.
- **virtual void printHelp (ostream&)const**
  Print help for the command.
- **virtual ~ConstraintCmd ()**

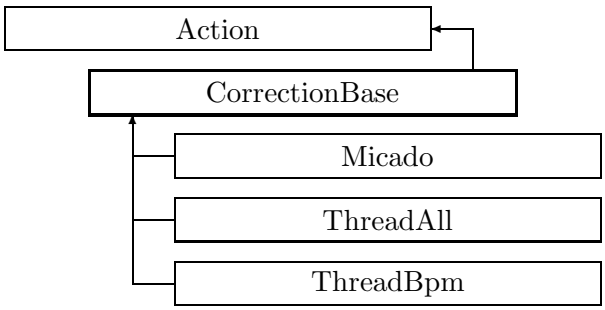


Figure 2.26: Inheritance for class CorrectionBase

2.38 class CorrectionBase

Abstract base class for all orbit correction commands.
Factors out all common behaviour for these algorithms.

Type:	abstract
Superclasses:	public Action
Include file:	./Tables/CorrectionBase.hh

Synopsis (including inherited members)

Public members

- **typedef TBeamline<Row> TLine**
  (see Section 2.38.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)

- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)

- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~CorrectionBase ()**
  (see Section 2.38.1)

Protected members

- **CorrectionBase (int,const char*,const char*)**
  Exemplar constructor. (see Section 2.38.1)
- **CorrectionBase (const string&,CorrectionBase*)**
  Clone constructor. (see Section 2.38.1)

- **struct Row**
  The comon attributes for orbit correction commands. Structure for a row of the Twiss table. (see Section 2.38.1)
- **void addKick (int,TLine::iterator&,double)**
  Add to kicker strength. (see Section 2.38.1)
- **void buildIdealMatrix ()**
  Build the ideal transfer matrices from beginning to current position. (see Section 2.38.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool isCorr [2]**
  Flags telling wether a corrector exists. (see Section 2.38.1)
- **bool isMoni [2]**
  Flag telling wether a monitor exists. (see Section 2.38.1)
- **TLine itsLine**
  The flat beam line on which the correction is done. (see Section 2.38.1)
- **void listCorrectors (int)**
  List correctors before or after correction. (see Section 2.38.1)
- **void listMonitors (int)**
  List monitors before or after correction. (see Section 2.38.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)
- **FVector<double,6> orbitGuess**
  The closed orbit guess. (see Section 2.38.1)
- **PartData reference**
  The particle reference data. (see Section 2.38.1)
- **void test (ElementBase*)**
  Routine to test for corrector or monitor. (see Section 2.38.1)

2.38.1 Detailed descriptions

Public members

- **typedef TBeamline<Row> TLine**

- **virtual ~CorrectionBase ()**

Protected members

- **CorrectionBase (int,const char*,const char*)**
  Exemplar constructor.
- **CorrectionBase (const string&,CorrectionBase*)**
  Clone constructor.
- **struct Row**
  The comon attributes for orbit correction commands. Structure for a row of the Twiss table.
- **void addKick (int,TLine::iterator&,double)**
  Add to kicker strength.

  Arguments:

  The plane: (0 = x, 1 = y).

  The position, given by an iterator into the Twiss table.

  The kick change.
- **void buildIdealMatrix ()**
  Build the ideal transfer matrices from beginning to current position.
- **bool isCorr [2]**
  Flags telling wether a corrector exists.
- **bool isMoni [2]**
  Flag telling wether a monitor exists.
- **TLine itsLine**
  The flat beam line on which the correction is done.
- **void listCorrectors (int)**
  List correctors before or after correction.
- **void listMonitors (int)**
  List monitors before or after correction.
- **FVector<double,6> orbitGuess**
  The closed orbit guess.
- **PartData reference**
  The particle reference data.
- **void test (ElementBase*)**
  Routine to test for corrector or monitor.

2.39 class Corrector

Type:	Instantiable

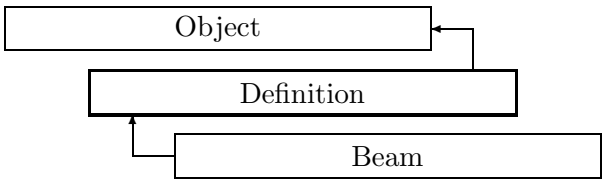


Figure 2.27: Inheritance for class Definition

2.40 class Definition

The base class for all MAD definitions.
It implements the common behaviour of definitions, it can also be used via dynamic casting to determine whether an object represents a definition.

Type:	abstract
Superclasses:	public Object
Include file:	./AbstractObjects/Definition.hh

Synopsis (including inherited members)

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.40.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)

- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.40.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.40.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Definition ()**
  (see Section 2.40.1)

Protected members

- **Definition (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.40.1)
- **Definition (const string&,Definition*)**
  Constructor for clones. (see Section 2.40.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.40.1 Detailed descriptions

Public members

- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "DEFINITION".

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always false for definitions.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always false for definitions.

- **virtual ~Definition ()**

Protected members

- **Definition (int,const char*,const char*)**
  Constructor for exemplars.

- **Definition (const string&,Definition*)**
  Constructor for clones.

2.41 class Directory

A map of string versus pointer to Object.
Used as the directory for MAD objects.

Type:	Instantiable
Include file:	./AbstractObjects/Directory.hh

Synopsis (including inherited members)

Public members

- **Directory ()**
  Constructor. (see Section 2.41.1)
- **ObjectDir::iterator begin ()**
  First object in alphabetic order of name (see Section 2.41.1)
- **ObjectDir::const_iterator begin ()const**
  First object in alphabetic order of name (see Section 2.41.1)
- **ObjectDir::iterator end ()**
  Last object in alphabetic order of name (see Section 2.41.1)
- **ObjectDir::const_iterator end ()const**
  Last object in alphabetic order of name (see Section 2.41.1)
- **void erase ()**
  Delete all entries. (see Section 2.41.1)
- **void erase (const string&)**
  Remove existing entry. (see Section 2.41.1)
- **Object* find (const string&)const**
  Find entry. (see Section 2.41.1)
- **void insert (const string&,Object*)**
  Define new object. (see Section 2.41.1)
- **~Directory ()**
  (see Section 2.41.1)

2.41.1 Detailed descriptions

Public members

- **Directory ()**
  Constructor.

  Build empty directory.
- **ObjectDir::iterator begin ()**
  First object in alphabetic order of name

  (Version for non-constant directory).

- **ObjectDir::const_iterator begin ()const**
  First object in alphabetic order of name

  (Version for constant directory).

- **ObjectDir::iterator end ()**
  Last object in alphabetic order of name

  (Version for non-constant directory).

- **ObjectDir::const_iterator end ()const**
  Last object in alphabetic order of name

  (Version for constant directory).

- **void erase ()**
  Delete all entries.

- **void erase (const string&)**
  Remove existing entry.

  The entry is identified by **name**.

- **Object* find (const string&)const**
  Find entry.

  The entry is identified by **name**. If the entry **name** does not exist, return **NULL**.

- **void insert (const string&,Object*)**
  Define new object.

  Insert new object in directory. If the entry **name** exists already, it is removed.

- **~Directory ()**

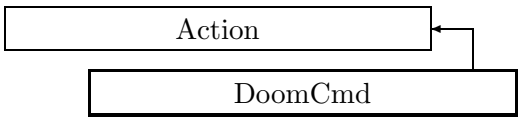


Figure 2.28: Inheritance for class DoomCmd

2.42 class DoomCmd

The DOOM command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/DoomCmd.hh

Synopsis (including inherited members)

Public members

- **DoomCmd ()**
  Exemplar constructor. (see Section 2.42.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual DoomCmd* clone (const string&)**
  Make clone. (see Section 2.42.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.42.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~DoomCmd ()**
  (see Section 2.42.1)

2.42.1 Detailed descriptions

Public members

- **DoomCmd ()**
  Exemplar constructor.
- **virtual DoomCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~DoomCmd ()**

2.43 class DoomDB

Encapsulates the DOOM data base.
Objects are read from the data base using a DoomReader object, and written using a DoomWriter object. These objects encapsulate Hans Grote's "object" structure. Their constructors and destructors handle the necessary transmission from and to the data base.

Type:	Instantiable
Include file:	./AbstractObjects/DoomDB.hh

Synopsis (including inherited members)

Public members

- **DoomDB ()**
  Constructor. (see Section 2.43.1)
- **void firstOpen (const char*)**
  Open data base (first time) identified by file name **dbName**. (see Section 2.43.1)
- **static int getAttributeIndex (const string&)**
  Find DOOM index for element attribute. (see Section 2.43.1)
- **void reOpen ()**
  Re-open data base which was already open. (see Section 2.43.1)
- **bool readAlign (const string&,int,Euclid3D&)**
  Read misalignment. (see Section 2.43.1)
- **static bool readField (const string&,int,double,const BMultipoleField&,BMultipoleField&)**
  (see Section 2.43.1)
- **Object* readObject (const string&)**
  Read object. (see Section 2.43.1)
- **void setDebug (int)**
  Set the debug flag. (see Section 2.43.1)
- **static void setEnvironment (const char[],const string&)**
  Set DOOM environment variable. (see Section 2.43.1)
- **void setUpdate (bool)**
  Set update flag. (see Section 2.43.1)
- **void shut ()**
  Close the data base. (see Section 2.43.1)
- **void writeAlign (const string&,int,const Euclid3D&)**
  Write misalignment. (see Section 2.43.1)
- **static void writeField (const string&,int,double,const BMultipoleField&,const BMultipoleField&)**
  (see Section 2.43.1)

- **void writeObject (Object*)**
  Write object. (see Section 2.43.1)
- **~DoomDB ()**
  Destructor. (see Section 2.43.1)

2.43.1 Detailed descriptions

Public members

- **DoomDB ()**
  Constructor.

  Initialise the data base, but do not open it.
- **void firstOpen (const char*)**
  Open data base (first time) identified by file name dbName.
- **static int getAttributeIndex (const string&)**
  Find DOOM index for element attribute.

  Find the internal index for the element attribute identified by **name**.
- **void reOpen ()**
  Re-open data base which was already open.
- **bool readAlign (const string&,int,Euclid3D&)**
  Read misalignment.
- **static bool readField (const string&,int,double,const BMultipoleField&,BMultipoleField&)**
- **Object* readObject (const string&)**
  Read object.

  Identified by the name **name**.
- **void setDebug (int)**
  Set the debug flag.

  See the DOOM documentation for details.
- **static void setEnvironment (const char[],const string&)**
  Set DOOM environment variable.

  Call doom_setenv(name, value). See the DOOM documentation for details.
- **void setUpdate (bool)**
  Set update flag.

  Called when an object is created or modified, to force writing all modified objects to the data base when MAD is shut down.
- **void shut ()**
  Close the data base.

- **void writeAlign (const string&,int,const Euclid3D&)**
  Write misalignment.
- **static void writeField (const string&,int,double,const BMultipoleField&,const BMultipoleField&)**
- **void writeObject (Object*)**
  Write object.
- **~DoomDB ()**
  Destructor.

  Do nothing.

2.44 class DoomReader

Encapsulates reading from the DOOM data base.
This class reads an object from the DOOM data base and gives access to its data.

Type:	Instantiable
Include file:	./AbstractObjects/DoomReader.hh

Synopsis (including inherited members)

Public members

- **DoomReader (const string&)**
  Constructor. (see Section 2.44.1)
- **DoomReader (const string&,int[])**
  Constructor. (see Section 2.44.1)
- **string getBaseName ()const**
  Return base name. (see Section 2.44.1)
- **int getInt (int)const**
  Return integer value. (see Section 2.44.1)
- **int getIntSize ()const**
  Number of integer values. (see Section 2.44.1)
- **string getParentName ()const**
  Return parent name. (see Section 2.44.1)
- **double getReal (int)const**
  Return real value. (see Section 2.44.1)
- **int getRealSize ()const**
  Number of real values. (see Section 2.44.1)
- **string getString (int)const**
  Return string value. (see Section 2.44.1)
- **int getStringSize ()const**
  Number of string values. (see Section 2.44.1)
- **string getTypeName ()const**
  Return type name. (see Section 2.44.1)
- **~DoomReader ()**
  (see Section 2.44.1)

2.44.1 Detailed descriptions

Public members

- **DoomReader (const string&)**
  Constructor.

Constructs a reader object for the MAD Object **name**. It reads the data from the DOOM data base and stores them internally.

- **DoomReader (const string&,int[])**
  Constructor.

  Like **DoomReader(name)**, but use **keyList** to further identify the object.

- **string getBaseName ()const**
  Return base name.

  This is the name of the MAD exemplar object from which the object is ultimately derived.

- **int getInt (int)const**
  Return integer value.

  Identified by **index**.

- **int getIntSize ()const**
  Number of integer values.

  Return number of integer values stored in this reader.

- **string getParentName ()const**
  Return parent name.

  This is the name of the MAD class object from which the object is derived directly.

- **double getReal (int)const**
  Return real value.

  Identified by **index**.

- **int getRealSize ()const**
  Number of real values.

  Return number of real values stored in this reader.

- **string getString (int)const**
  Return string value.

  Identified by **index**.

- **int getStringSize ()const**
  Number of string values.

  Return number of string values stored in this reader.

- **string getTypeName ()const**
  Return type name.

  This is the DOOM object type like "ELEMENT", "ACTION", etc.

- **~DoomReader ()**

2.45 class DoomWriter

Encapsulates writing to the DOOM data base.
This class constructs a DOOM "object" to be writtin to the DOOM data base and writes it out automatically when the DoomWriter goes out of scope.

Type:	Instantiable
Include file:	./AbstractObjects/DoomWriter.hh

Synopsis (including inherited members)

Public members

- **DoomWriter ()**
  Default constructor. (see Section 2.45.1)
- **DoomWriter (const string&)**
  Constructor. (see Section 2.45.1)
- **DoomWriter (const string&,const int[])**
  Constructor. (see Section 2.45.1)
- **void putInt (int,int)**
  Store real value. (see Section 2.45.1)
- **void putReal (int,double)**
  Store real value. (see Section 2.45.1)
- **void putString (int,const string&)**
  Store real value. (see Section 2.45.1)
- **void setBaseName (const string&)**
  Set base name. (see Section 2.45.1)
- **void setObjectName (const string&)**
  Set object name. (see Section 2.45.1)
- **void setObjectName (const string&,const int[])**
  Set object name. (see Section 2.45.1)
- **void setParentName (const string&)**
  Set parent name. (see Section 2.45.1)
- **void setTypeName (const string&)**
  Set type name. (see Section 2.45.1)
- **~DoomWriter ()**
  Destructor. (see Section 2.45.1)

2.45.1 Detailed descriptions

Public members

- **DoomWriter ()**
  Default constructor.

  Constructs an empty writer object. An object constructed with the default constructor can only be written if its name is set before the destructor is called.

- **DoomWriter (const string&)**
  Constructor.

  Construct an empty writer object for the MAD Object **name**.

- **DoomWriter (const string&,const int[])**
  Constructor.

  Construct an empty writer object for the MAD Object **name**. Like **DoomWriter(name)**, but use **keyList** to further identify the object.

- **void putInt (int,int)**
  Store real value.

  Identified by **index**.

- **void putReal (int,double)**
  Store real value.

  Identified by **index**.

- **void putString (int,const string&)**
  Store real value.

  Identified by **index**.

- **void setBaseName (const string&)**
  Set base name.

  This is the name of the MAD exemplar object from which the object is ultimately derived.

- **void setObjectName (const string&)**
  Set object name.

  This is the name of the MAD object itself.

- **void setObjectName (const string&,const int[])**
  Set object name.

  Like **setObjectName(name)**, but use **keyList** to further identify the object.

- **void setParentName (const string&)**
  Set parent name.

  This is the name of the MAD class object from which the object is derived directly.

- **void setTypeName (const string&)**
  Set type name.

  This is the DOOM object type like "ELEMENT", "ACTION", etc.

- **~DoomWriter ()**
  Destructor.

  Save the object to the DOOM data base, if its name is set.

2.46 class Drift

Type:	Instantiable

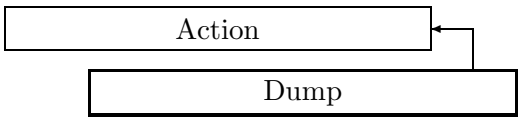


Figure 2.29: Inheritance for class Dump

2.47 class Dump

The DUMP command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Dump.hh

Synopsis (including inherited members)

Public members

- **Dump ()**
  Exemplar constructor. (see Section 2.47.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Dump* clone (const string&)**
  Make clone. (see Section 2.47.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.47.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Dump ()**
  (see Section 2.47.1)

2.47.1 Detailed descriptions

Public members

- **Dump ()**
  Exemplar constructor.
- **virtual Dump* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Dump ()**

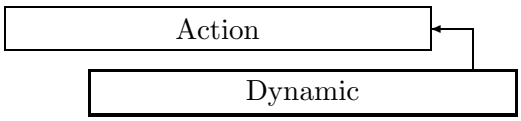


Figure 2.30: Inheritance for class Dynamic

2.48 class Dynamic

The DYNAMIC command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./PhysicsActions/Dynamic.hh

Synopsis (including inherited members)

Public members

- **Dynamic ()**
  Exemplar constructor. (see Section 2.48.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Dynamic* clone (const string&)**
  Make clone. (see Section 2.48.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.48.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Dynamic ()**
  (see Section 2.48.1)

2.48.1 Detailed descriptions

Public members

- **Dynamic ()**
  Exemplar constructor.
- **virtual Dynamic* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Dynamic ()**

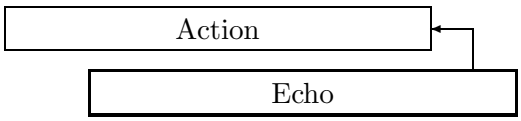


Figure 2.31: Inheritance for class Echo

2.49 class Echo

The ECHO command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Echo.hh

Synopsis (including inherited members)

Public members

- **Echo ()**
  Exemplar constructor. (see Section 2.49.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Echo* clone (const string&)**
  Make clone. (see Section 2.49.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.49.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.49.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Echo ()**
  (see Section 2.49.1)

2.49.1 Detailed descriptions

Public members

- **Echo ()**
  Exemplar constructor.
- **virtual Echo* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Echo ()**

2.50 class Edit

This class contains all data for the sequence editor.
It acts as a communication area between the sequence editor commands.

Type:	Instantiable
Include file:	./Editor/Edit.hh

Synopsis (including inherited members)

Public members

- **Edit (Sequence*)**
  Constructor. (see Section 2.50.1)
- **typedef Sequence::TLine TLine**
  The type of line contained in a sequence. (see Section 2.50.1)
- **static Edit* block**
  Pointer to the edit data. (see Section 2.50.1)
- **bool cycle (const PlaceRep&)**
  Cycle the edit sequence. (see Section 2.50.1)
- **void finish (const string&)**
  Finish editing. (see Section 2.50.1)
- **void flatten ()**
  Flatten the edit sequence. (see Section 2.50.1)
- **int installMultiple (ElementBase*,double)**
  Install multiple elements. (see Section 2.50.1)
- **int installSingle (const PlaceRep&,ElementBase*,double)**
  Install element relative to place. (see Section 2.50.1)
- **bool isModified**
  Modify flag. (see Section 2.50.1)
- **typedef Sequence::TLine::iterator iterator**
  The line iterator for a sequence. (see Section 2.50.1)
- **Pointer<TLine> itsLine**
  The edit sequence. (see Section 2.50.1)
- **Pointer<Sequence> itsSequence**
  The original sequence. (see Section 2.50.1)
- **int moveMultiple (double)**
  Move multiple elements. (see Section 2.50.1)
- **int moveSingleAbs (const PlaceRep&,double)**
  Move single element. (see Section 2.50.1)

- **int moveSingleRel (const PlaceRep&,const PlaceRep&,double)**
  Move single element. (see Section 2.50.1)
- **EditParser parser**
  The parser used during a sequence edit. (see Section 2.50.1)
- **void reflect ()**
  Reflect the edit sequence. (see Section 2.50.1)
- **int removeMultiple ()**
  Remove multiple elements. (see Section 2.50.1)
- **int removeSingle (const PlaceRep&)**
  Remove single element. (see Section 2.50.1)
- **int replaceMultiple (ElementBase*)**
  Replace multiple elements. (see Section 2.50.1)
- **int replaceSingle (const PlaceRep&,ElementBase*)**
  Replace single element. (see Section 2.50.1)
- **int select (const RangeRep&,const string&,const string&,const string&)**
  Select elements in the edit sequence. (see Section 2.50.1)
- **void selectClear ()**
  Clear all selection flags. (see Section 2.50.1)
- **void selectFull ()**
  Set all selection flags. (see Section 2.50.1)
- **~Edit ()**
  (see Section 2.50.1)

2.50.1 Detailed descriptions

Public members

- **Edit (Sequence*)**
  Constructor.

  Prepares the given sequence for editing. Makes a copy with all drifts removed.
- **typedef Sequence::TLine TLine**
  The type of line contained in a sequence.
- **static Edit* block**
  Pointer to the edit data.
- **bool cycle (const PlaceRep&)**
  Cycle the edit sequence.

  The new start point is at **start**.

- **void finish (const string&)**
  Finish editing.

  Reconstruct the modified sequence as required. If a new name is given, make a copy, otherwise if the sequence is modified, overwrite the original.

- **void flatten ()**
  Flatten the edit sequence.

- **int installMultiple (ElementBase*,double)**
  Install multiple elements.

  New element **elem** at position **at** from all selected elements.

- **int installSingle (const PlaceRep&,ElementBase*,double)**
  Install element relative to place.

  New element **elem** at position **at** from **from** (if given) or from origin.

- **bool isModified**
  Modify flag.

  If true, the edit sequence is different from the original.

- **typedef Sequence::TLine::iterator iterator**
  The line iterator for a sequence.

- **Pointer<TLine> itsLine**
  The edit sequence.

- **Pointer<Sequence> itsSequence**
  The original sequence.

- **int moveMultiple (double)**
  Move multiple elements.

  Move all selected elements by **by**.

- **int moveSingleAbs (const PlaceRep&,double)**
  Move single element.

  Move element at **pos** to absolute position.

- **int moveSingleRel (const PlaceRep&,const PlaceRep&,double)**
  Move single element.

  Move element at **pos** by given amount.

- **EditParser parser**
  The parser used during a sequence edit.

- **void reflect ()**
  Reflect the edit sequence.

- **int removeMultiple ()**
  Remove multiple elements.

  Remove all selected elements.

- **int removeSingle (const PlaceRep&)**
  Remove single element.

  Remove element at **pos**

- **int replaceMultiple (ElementBase*)**
  Replace multiple elements.

  Replace all selected elements by **elem**.

- **int replaceSingle (const PlaceRep&,ElementBase*)**
  Replace single element.

  Replace element at **pos** by **elem**.

- **int select (const RangeRep&,const string&,const string&,const string&)**
  Select elements in the edit sequence.

  Use range, class and regular expression.

- **void selectClear ()**
  Clear all selection flags.

- **void selectFull ()**
  Set all selection flags.

- **~Edit ()**

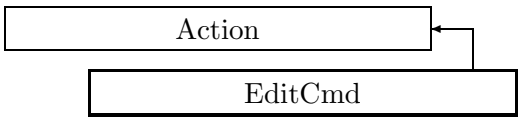


Figure 2.32: Inheritance for class EditCmd

2.51 class EditCmd

The sequence editor SEQEDIT command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Editor/EditCmd.hh

Synopsis (including inherited members)

Public members

- **EditCmd ()**
  Exemplar constructor. (see Section 2.51.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditCmd* clone (const string&)**
  Make clone. (see Section 2.51.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.51.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditCmd ()**
  (see Section 2.51.1)

2.51.1 Detailed descriptions

Public members

- **EditCmd ()**
  Exemplar constructor.
- **virtual EditCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditCmd ()**

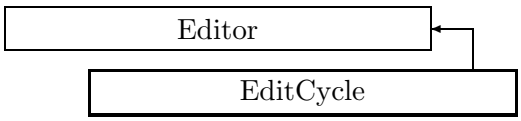


Figure 2.33: Inheritance for class EditCycle

2.52 class EditCycle

The sequence editor CYCLE command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditCycle.hh

Synopsis (including inherited members)

Public members

- **EditCycle ()**
  Exemplar constructor. (see Section 2.52.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditCycle* clone (const string&)**
  Make clone. (see Section 2.52.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.52.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditCycle ()**
  (see Section 2.52.1)

2.52.1 Detailed descriptions

Public members

- **EditCycle ()**
  Exemplar constructor.
- **virtual EditCycle* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditCycle ()**

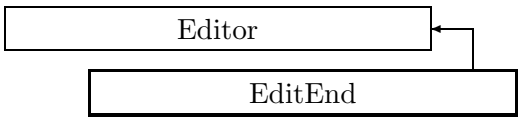


Figure 2.34: Inheritance for class EditEnd

2.53 class EditEnd

The sequence editor ENDEDIT command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditEnd.hh

Synopsis (including inherited members)

Public members

- **EditEnd ()**
  Exemplar constructor. (see Section 2.53.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditEnd* clone (const string&)**
  Make clone. (see Section 2.53.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.53.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditEnd ()**
  (see Section 2.53.1)

2.53.1 Detailed descriptions

Public members

- **EditEnd ()**
  Exemplar constructor.
- **virtual EditEnd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditEnd ()**

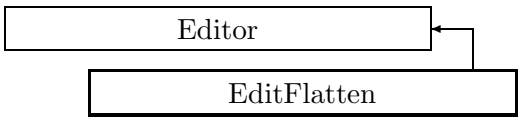


Figure 2.35: Inheritance for class EditFlatten

2.54 class EditFlatten

The sequence editor FLATTEN command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditFlatten.hh

Synopsis (including inherited members)

Public members

- **EditFlatten ()**
  Exemplar constructor. (see Section 2.54.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditFlatten* clone (const string&)**
  Make clone. (see Section 2.54.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.54.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditFlatten ()**
  (see Section 2.54.1)

2.54.1 Detailed descriptions

Public members

- **EditFlatten ()**
  Exemplar constructor.
- **virtual EditFlatten* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditFlatten ()**

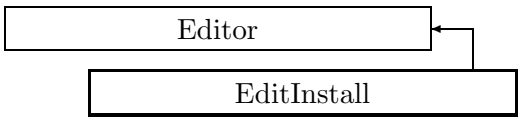


Figure 2.36: Inheritance for class EditInstall

2.55 class EditInstall

The sequence editor INSTALL command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditInstall.hh

Synopsis (including inherited members)

Public members

- **EditInstall ()**
  Exemplar constructor. (see Section 2.55.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditInstall* clone (const string&)**
  Make clone. (see Section 2.55.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.55.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the command. (see Section 2.55.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditInstall ()**
  (see Section 2.55.1)

2.55.1 Detailed descriptions

Public members

- **EditInstall ()**
  Exemplar constructor.
- **virtual EditInstall* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse the command.

  Special format for this sub-command.
- **virtual ~EditInstall ()**

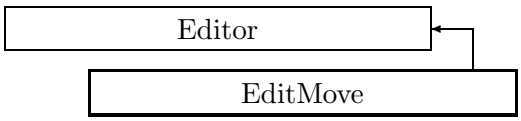


Figure 2.37: Inheritance for class EditMove

2.56 class EditMove

The sequence editor MOVE command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditMove.hh

Synopsis (including inherited members)

Public members

- **EditMove ()**
  Exemplar constructor. (see Section 2.56.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditMove* clone (const string&)**
  Make clone. (see Section 2.56.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.56.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditMove ()**
  (see Section 2.56.1)

2.56.1 Detailed descriptions

Public members

- **EditMove ()**
  Exemplar constructor.
- **virtual EditMove* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditMove ()**

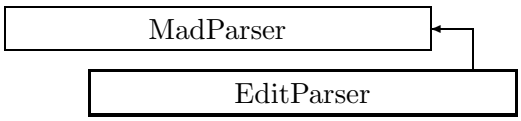


Figure 2.38: Inheritance for class EditParser

2.57 class EditParser

The parser for the MAD sequence editor.
As long as control remains in this class, MAD recognises only the commands allowed during sequence editing.

Type:	Instantiable
Superclasses:	public MadParser
Include file:	./Editor/EditParser.hh

Synopsis (including inherited members)

Public members

- **EditParser ()**
  (see Section 2.57.1)
- **virtual void parse (Statement&)const**
  Parse and execute current statement. (see Section 2.118.1)
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream. (see Section 2.118.1)
- **static Token readToken ()**
  Return next input token. (see Section 2.118.1)
- **virtual void run ()const**
  Read current stream. (see Section 2.118.1)
- **virtual void run (TokenStream*)const**
  Read given stream. (see Section 2.118.1)
- **void stop ()const**
  Set stop flag. (see Section 2.118.1)
- **virtual ~EditParser ()**
  (see Section 2.57.1)

Protected members

- **void execute (Object*,const string&)const**
  Execute or check the current command. (see Section 2.118.1)
- **virtual Object* find (const string&)const**
  Find object by name in the sequence editor command directory. (see Section 2.57.1)

- **virtual void parse (Statement&)const**
  Parse and execute current statement. (see Section 2.57.1)
- **virtual void parseAction (Statement&)const**
  Parse executable command. (see Section 2.118.1)
- **virtual void parseAssign (Statement&)const**
  Parse assignment statement. (see Section 2.118.1)
- **virtual void parseDefine (Statement&)const**
  Parse definition. (see Section 2.118.1)
- **virtual void parseEnd (Statement&)const**
  Check for end of statement. (see Section 2.118.1)
- **virtual void parseInstall (Statement&)const**
  Parse definition. (see Section 2.57.1)
- **virtual void parseMacro (const string&,Statement&)const**
  Parse macro definition or call. (see Section 2.118.1)
- **virtual void printHelp (const string&)const**
  Print help on named command. (see Section 2.118.1)

2.57.1 Detailed descriptions

Public members

- **EditParser ()**

- **virtual ~EditParser ()**

Protected members

- **virtual Object* find (const string&)const**
  Find object by name in the sequence editor command directory.
- **virtual void parse (Statement&)const**
  Parse and execute current statement.
- **virtual void parseInstall (Statement&)const**
  Parse definition.

  Special version for INSTALL command.

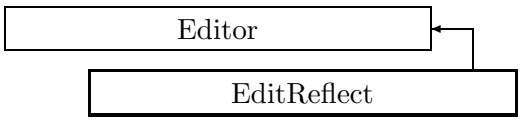


Figure 2.39: Inheritance for class EditReflect

2.58 class EditReflect

The sequence editor REFLECT command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditReflect.hh

Synopsis (including inherited members)

Public members

- **EditReflect ()**
  Exemplar constructor. (see Section 2.58.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditReflect* clone (const string&)**
  Make clone. (see Section 2.58.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.58.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditReflect ()**
  (see Section 2.58.1)

2.58.1 Detailed descriptions

Public members

- **EditReflect ()**
  Exemplar constructor.
- **virtual EditReflect* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditReflect ()**

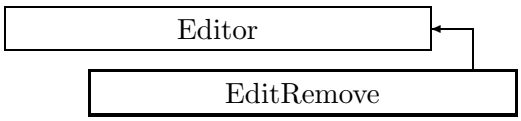


Figure 2.40: Inheritance for class EditRemove

2.59 class EditRemove

The sequence editor REMOVE command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditRemove.hh

Synopsis (including inherited members)

Public members

- **EditRemove ()**
  Exemplar constructor. (see Section 2.59.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditRemove* clone (const string&)**
  Make clone. (see Section 2.59.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.59.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditRemove ()**
  (see Section 2.59.1)

2.59.1 Detailed descriptions

Public members

- **EditRemove ()**
  Exemplar constructor.
- **virtual EditRemove* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditRemove ()**

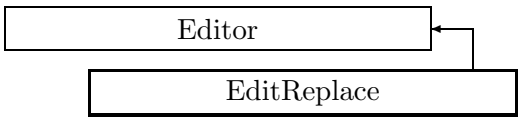


Figure 2.41: Inheritance for class EditReplace

2.60 class EditReplace

The sequence editor REPLACE command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditReplace.hh

Synopsis (including inherited members)

Public members

- **EditReplace ()**
  Exemplar constructor. (see Section 2.60.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditReplace* clone (const string&)**
  Make clone. (see Section 2.60.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.60.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditReplace ()**
  (see Section 2.60.1)

2.60.1 Detailed descriptions

Public members

- **EditReplace ()**
  Exemplar constructor.
- **virtual EditReplace* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditReplace ()**

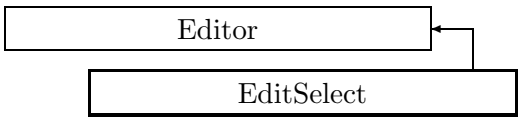


Figure 2.42: Inheritance for class EditSelect

2.61 class EditSelect

The sequence editor SELECT command.

Type:	Instantiable
Superclasses:	public Editor
Include file:	./Editor/EditSelect.hh

Synopsis (including inherited members)

Public members

- **EditSelect ()**
  Exemplar constructor. (see Section 2.61.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual EditSelect* clone (const string&)**
  Make clone. (see Section 2.61.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.61.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~EditSelect ()**
  (see Section 2.61.1)

2.61.1 Detailed descriptions

Public members

- **EditSelect ()**
  Exemplar constructor.
- **virtual EditSelect* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~EditSelect ()**

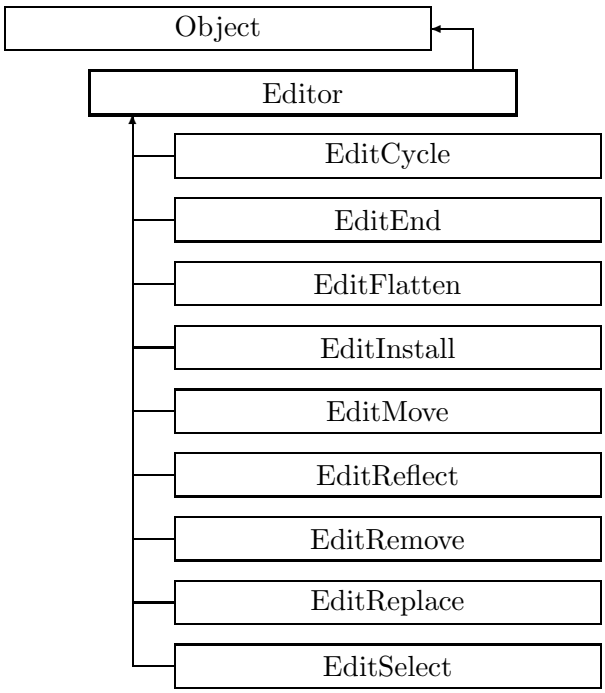


Figure 2.43: Inheritance for class Editor

2.62 class Editor

The base class for all MAD sequence editor commands.
It implements the common behaviour of editor commands, it can also be used via dynamic casting to determine whether an object represents an editor command.

Type:	abstract
Superclasses:	public Object
Include file:	./AbstractObjects/Editor.hh

Synopsis (including inherited members)

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)

- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.62.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.62.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.62.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Editor ()**
  (see Section 2.62.1)

Protected members

- **Editor (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.62.1)
- **Editor (const string&,Editor*)**
  Constructor for cloning. (see Section 2.62.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.62.1 Detailed descriptions

Public members

- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "EDITOR".
- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always true for editor commands.
- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always false for editor commands.
- **virtual ~Editor ()**

Protected members

- **Editor (int,const char*,const char*)**
  Constructor for exemplars.
- **Editor (const string&,Editor*)**
  Constructor for cloning.

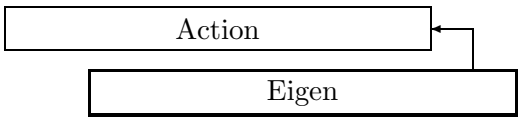


Figure 2.44: Inheritance for class Eigen

2.63 class Eigen

The EIGEN command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Tables/Eigen.hh

Synopsis (including inherited members)

Public members

- **Eigen ()**
  Exemplar constructor. (see Section 2.63.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Eigen* clone (const string&)**
  Make clone. (see Section 2.63.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.63.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Eigen ()**
  (see Section 2.63.1)

2.63.1 Detailed descriptions

Public members

- **Eigen ()**
  Exemplar constructor.
- **virtual Eigen* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Eigen ()**

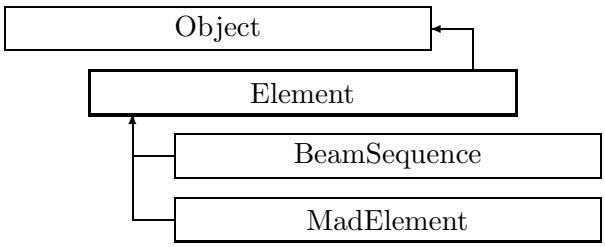


Figure 2.45: Inheritance for class Element

2.64 class Element

The base class for all MAD elements.
It implements the common behaviour of elements, it can also be used via dynamic casting to determine whether an object represents an element. Each Element object contains a pointer to a CLASSIC beam line element. This element is wrapped as required in a field wrapper and an AlignWrapper. The assembly represents the element as it occurs in a beam line. According to the setting of the sharable flag, the assembly is shared or unique when used in more than one place. When shared, the wrappers ensure one and the same imperfection for all shared occurrences. When unique, each occurrence has its own imperfections. The "design" element is normally shared.

Type:	abstract
Superclasses:	public Object
Include file:	./AbstractObjects/Element.hh

Synopsis (including inherited members)

Public members

- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)

- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.64.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)

- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)

- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Element ()**
  (see Section 2.64.1)

Protected members

- **Element (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.64.1)
- **Element (const string&,Element*)**
  Constructor for clones. (see Section 2.64.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.64.1 Detailed descriptions

Public members

- **enum ReferenceType**
  Reference for element positioning.

  Used in the SEQUENCE command.
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed.

  Return true, if the replacement is also an Element.
- **static Element* find (const string&)**
  Find named Element.

  If an element with the name **name** exists, return a pointer to that element. If no such element exists, throw **MadException**.
- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "ELEMENT".

- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element.

  Return a pointer to the embedded CLASSIC ElementBase

- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !).

- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !).

- **virtual double getLength ()const**
  Return element length.

- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element.

- **virtual void setShared (bool)**
  Set shared flag.

  If true, all references to this name are to the same object.

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always false for elements.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always false for elements.

- **virtual ~Element ()**

Protected members

- **Element (int,const char*,const char*)**
  Constructor for exemplars.

- **Element (const string&,Element*)**
  Constructor for clones.

2.65 class ElementBase

Type:	Instantiable

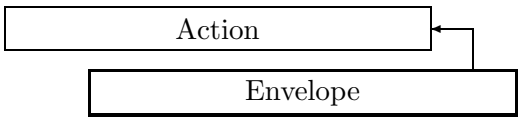


Figure 2.46: Inheritance for class Envelope

2.66 class Envelope

The ENVELOPE command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Tables/Envelope.hh

Synopsis (including inherited members)

Public members

- **Envelope ()**
  Exemplar constructor. (see Section 2.66.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Envelope* clone (const string&)**
  Make clone. (see Section 2.66.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.66.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Envelope ()**
  (see Section 2.66.1)

2.66.1 Detailed descriptions

Public members

- **Envelope ()**
  Exemplar constructor.
- **virtual Envelope* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Envelope ()**

2.67 class Error

Error block.
This class encapsulates all data for Error definitions. It acts as a communication area between the error definition commands.

Type:	Instantiable
Include file:	./Errors/Error.hh

Synopsis (including inherited members)

Public members

- **Error ()**
  (see Section 2.67.1)
- **bool addError**
  If true, errors are additive. (see Section 2.67.1)
- **static Error* block**
  The static block for the error data. (see Section 2.67.1)
- **Beamline* itsLine**
  The line for which errors are to be generated. (see Section 2.67.1)
- **ErrorParser parser**
  The parser active during error definition. (see Section 2.67.1)
- **~Error ()**
  (see Section 2.67.1)

2.67.1 Detailed descriptions

Public members

- **Error ()**

- **bool addError**
  If true, errors are additive.
- **static Error* block**
  The static block for the error data.
- **Beamline* itsLine**
  The line for which errors are to be generated.
- **ErrorParser parser**
  The parser active during error definition.
- **~Error ()**

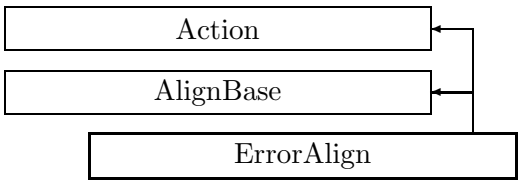


Figure 2.47: Inheritance for class ErrorAlign

2.68 class ErrorAlign

The EALIGN command.

Type:	Instantiable
Superclasses:	public Action, private AlignBase
Include file:	./Errors/ErrorAlign.hh

Synopsis (including inherited members)

Public members

- **ErrorAlign ()**
  Exemplar constructor. (see Section 2.68.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorAlign* clone (const string&)**
  Make clone. (see Section 2.68.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.68.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Generate a misalignment. (see Section 2.68.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorAlign ()**
  (see Section 2.68.1)

2.68.1 Detailed descriptions

Public members

- **ErrorAlign ()**
  Exemplar constructor.
- **virtual ErrorAlign* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.

- **virtual void misalignment (const AlignWrapper&,int)**
  Generate a misalignment.

  Prepare a misalignment error according to the specification in this command and assign it to the given AlignWrapper.

- **virtual ~ErrorAlign ()**

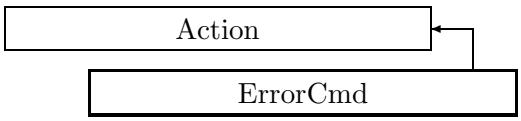


Figure 2.48: Inheritance for class ErrorCmd

2.69 class ErrorCmd

The ERROR command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Errors/ErrorCmd.hh

Synopsis (including inherited members)

Public members

- **ErrorCmd ()**
  Exemplar constructor. (see Section 2.69.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorCmd* clone (const string&)**
  Make clone. (see Section 2.69.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.69.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorCmd ()**
  (see Section 2.69.1)

2.69.1 Detailed descriptions

Public members

- **ErrorCmd ()**
  Exemplar constructor.
- **virtual ErrorCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~ErrorCmd ()**

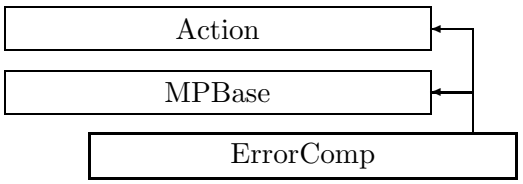


Figure 2.49: Inheritance for class ErrorComp

2.70 class ErrorComp

The EFCOMP command.

Type:	Instantiable
Superclasses:	public Action, private MPBase
Include file:	./Errors/ErrorComp.hh

Synopsis (including inherited members)

Public members

- **ErrorComp ()**
  Exemplar constructor. (see Section 2.70.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorComp* clone (const string&)**
  Make clone. (see Section 2.70.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.70.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Generate error field. (see Section 2.70.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)

- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorComp ()**
  (see Section 2.70.1)

2.70.1 Detailed descriptions

Public members

- **ErrorComp ()**
  Exemplar constructor.
- **virtual ErrorComp* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.

- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&) Generate error field.**

  The first two arguments are not used, relative errors are related to **mainField**. The result is stored in **errorField**.

- **virtual ~ErrorComp ()**

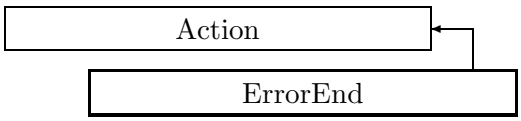


Figure 2.50: Inheritance for class ErrorEnd

2.71 class ErrorEnd

The ENDERROR command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Errors/ErrorEnd.hh

Synopsis (including inherited members)

Public members

- **ErrorEnd ()**
  Exemplar constructor. (see Section 2.71.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorEnd* clone (const string&)**
  Make clone. (see Section 2.71.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.71.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorEnd ()**
  (see Section 2.71.1)

2.71.1 Detailed descriptions

Public members

- **ErrorEnd ()**
  Exemplar constructor.
- **virtual ErrorEnd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~ErrorEnd ()**

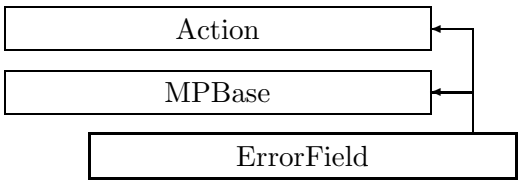


Figure 2.51: Inheritance for class ErrorField

2.72 class ErrorField

The EFIELD command.

Type:	Instantiable
Superclasses:	public Action, private MPBase
Include file:	./Errors/ErrorField.hh

Synopsis (including inherited members)

Public members

- **ErrorField ()**
  Exemplar constructor. (see Section 2.72.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorField* clone (const string&)**
  Make clone. (see Section 2.72.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.72.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Generate error field. (see Section 2.72.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)

- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorField ()**
  (see Section 2.72.1)

2.72.1 Detailed descriptions

Public members

- **ErrorField ()**
  Exemplar constructor.
- **virtual ErrorField* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.

- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&) Generate error field.**

  Uses the following arguments:

  Element name.

  Occurrence count.

  The main multipole field.

  The modifier list.

- **virtual ~ErrorField ()**

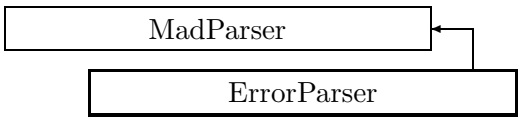


Figure 2.52: Inheritance for class ErrorParser

2.73 class ErrorParser

The parser used in the error module.
As long as control remains in this class, MAD recognises only the commands allowed in error mode.

Type:	Instantiable
Superclasses:	public MadParser
Include file:	./Errors/ErrorParser.hh

Synopsis (including inherited members)

Public members

- **ErrorParser ()**
  (see Section 2.73.1)
- **virtual void parse (Statement&)const**
  Parse and execute current statement. (see Section 2.118.1)
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream. (see Section 2.118.1)
- **static Token readToken ()**
  Return next input token. (see Section 2.118.1)
- **virtual void run ()const**
  Read current stream. (see Section 2.118.1)
- **virtual void run (TokenStream*)const**
  Read given stream. (see Section 2.118.1)
- **void stop ()const**
  Set stop flag. (see Section 2.118.1)
- **virtual ~ErrorParser ()**
  (see Section 2.73.1)

Protected members

- **void execute (Object*,const string&)const**
  Execute or check the current command. (see Section 2.118.1)
- **virtual Object* find (const string&)const**
  Find object by name in the error command directory. (see Section 2.73.1)

- **virtual void parseAction (Statement&)const**
  Parse executable command. (see Section 2.118.1)
- **virtual void parseAssign (Statement&)const**
  Parse assignment statement. (see Section 2.118.1)
- **virtual void parseDefine (Statement&)const**
  Parse definition. (see Section 2.118.1)
- **virtual void parseEnd (Statement&)const**
  Check for end of statement. (see Section 2.118.1)
- **virtual void parseMacro (const string&,Statement&)const**
  Parse macro definition or call. (see Section 2.118.1)
- **virtual void printHelp (const string&)const**
  Print help on named command. (see Section 2.118.1)

2.73.1 Detailed descriptions

Public members

- **ErrorParser ()**

- **virtual ~ErrorParser ()**

Protected members

- **virtual Object* find (const string&)const**
  Find object by name in the error command directory.

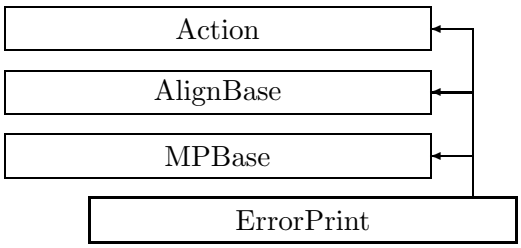


Figure 2.53: Inheritance for class ErrorPrint

2.74 class ErrorPrint

The EPRINT command.

Type:	Instantiable
Superclasses:	public Action, private AlignBase, private MPBase
Include file:	./Errors/ErrorPrint.hh

Synopsis (including inherited members)

Public members

- **ErrorPrint ()**
  Exemplar constructor. (see Section 2.74.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorPrint* clone (const string&)**
  Make clone. (see Section 2.74.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  (see Section 2.74.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Print error field. (see Section 2.74.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)

- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Print misalignement. (see Section 2.74.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)

- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorPrint ()**
  (see Section 2.74.1)

2.74.1 Detailed descriptions

Public members

- **ErrorPrint ()**
  Exemplar constructor.
- **virtual ErrorPrint* clone (const string&)**
  Make clone.
- **virtual void execute ()**
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Print error field.

  Use the name and occurrence count for the current element. Relative errors are related to **mainField**. The field error is fetched from **modList**.
- **virtual void misalignment (const AlignWrapper&,int)**
  Print misalignement.

  All information is taken from the given AlignWrapper.
- **virtual ~ErrorPrint ()**

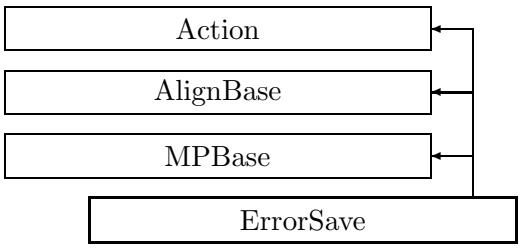


Figure 2.54: Inheritance for class ErrorSave

2.75 class ErrorSave

The ESAVE command.

Type:	Instantiable
Superclasses:	public Action, private AlignBase, private MPBase
Include file:	./Errors/ErrorSave.hh

Synopsis (including inherited members)

Public members

- **ErrorSave ()**
  Exemplar constructor. (see Section 2.75.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorSave* clone (const string&)**
  Make clone. (see Section 2.75.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.75.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Save error field. (see Section 2.75.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)

- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual void misalignment (const AlignWrapper&,int)**
  Save misalignement. (see Section 2.75.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)

- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorSave ()**
  (see Section 2.75.1)

2.75.1 Detailed descriptions

Public members

- **ErrorSave ()**
  Exemplar constructor.
- **virtual ErrorSave* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Save error field.

  Use the name and occurrence count for the current element. Relative errors are related to **mainField**. The field error is fetched from **modList**.
- **virtual void misalignment (const AlignWrapper&,int)**
  Save misalignement.

  All information is taken from the given AlignWrapper.
- **virtual ~ErrorSave ()**

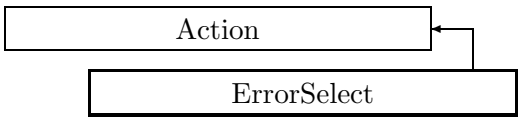


Figure 2.55: Inheritance for class ErrorSelect

2.76 class ErrorSelect

The error SELECT command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Errors/ErrorSelect.hh

Synopsis (including inherited members)

Public members

- **ErrorSelect ()**
  Exemplar constructor. (see Section 2.76.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ErrorSelect* clone (const string&)**
  Make clone. (see Section 2.76.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.76.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ErrorSelect ()**
  (see Section 2.76.1)

2.76.1 Detailed descriptions

Public members

- **ErrorSelect ()**
  Exemplar constructor.
- **virtual ErrorSelect* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~ErrorSelect ()**

2.77 class Euclid3D

Type:	Instantiable

2.78 template class FTps <class,int>

Type:	Instantiable

2.79 template class FVps <class,int>

Type:	Instantiable

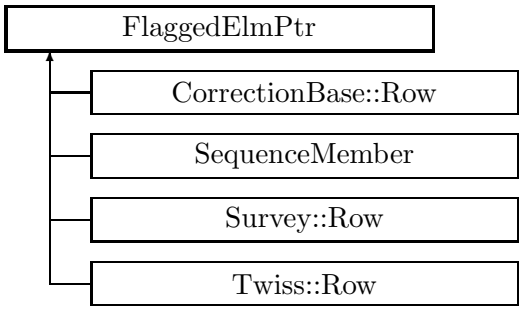


Figure 2.56: Inheritance for class FlaggedElmPtr

2.80 class FlaggedElmPtr

Type:	Instantiable

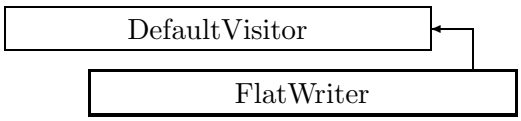


Figure 2.57: Inheritance for class FlatWriter

2.81 class FlatWriter

Visitor class for writing a line or sequence as a flat sequence in MAD-8 format.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Lines/FlatWriter.hh

Synopsis (including inherited members)

Public members

- **FlatWriter (const Beamline&,Sequence::TLine&,std::ostream&)**
  (see Section 2.81.1)
- **const Sequence::TLine& getSequence ()const**
  Return the temporary sequence. (see Section 2.81.1)
- **virtual void visitDrift (const Drift&)**
  Override drift operation. (see Section 2.81.1)
- **virtual void visitMapIntegrator (const MapIntegrator&)**
  Override beamline exit. (see Section 2.81.1)
- **~FlatWriter ()**
  (see Section 2.81.1)

Protected members

- **virtual void applyDefault (const ElementBase&)**
  Apply default. (see Section 2.81.1)

2.81.1 Detailed descriptions

Public members

- **FlatWriter (const Beamline&,Sequence::TLine&,std::ostream&)**

- **const Sequence::TLine& getSequence ()const**
  Return the temporary sequence.
- **virtual void visitDrift (const Drift&)**
  Override drift operation.

  Accumulate length, but do not save anything.

- **virtual void visitMapIntegrator (const MapIntegrator&)**
  Override beamline exit.
- **~FlatWriter ()**

Protected members

- **virtual void applyDefault (const ElementBase&)**
  Apply default.

  Add the element to the sequence list.

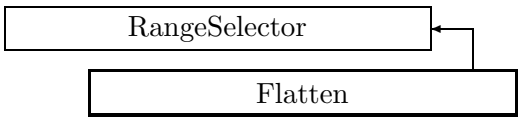


Figure 2.58: Inheritance for class Flatten

2.82 template class Flatten <class>

Flatten a beamline.
The type of beam line members in the flat line is given as a template parameter. It may be any class derived from ElmPtr.

Type:	Instantiable
Superclasses:	public RangeSelector
Include file:	./Tables/Flatten.hh

Synopsis (including inherited members)

Public members

- **Flatten (const Beamline&,TBeamline<Member>&,const RangeRep&)**
  Constructor. (see Section 2.82.1)
- **virtual void execute ()**
  Apply the algorithm to the top-level beamline. (see Section 2.82.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the visitor to an FlaggedElmPtr. (see Section 2.158.1)
- **virtual ~Flatten ()**
  (see Section 2.82.1)

Protected members

- **virtual void handleBeamline (const FlaggedElmPtr&)**
  The operation to be done for beamlines. (see Section 2.82.1)
- **virtual void handleElement (const FlaggedElmPtr&)**
  The operation to be done for elements. (see Section 2.82.1)
- **RangeRep itsRange**
  Working data for range. (see Section 2.158.1)
- **TBeamline<Member>& itsTable**
  The flat list to be filled. (see Section 2.82.1)

2.82.1 Detailed descriptions

Public members

- **Flatten (const Beamline&,TBeamline<Member>&,const RangeRep&)**
  Constructor.

  Attach this visitor to **bl**, **mem** will receive the range **range** ot the flat line.

- **virtual void execute ()**
  Apply the algorithm to the top-level beamline.
- **virtual ~Flatten ()**

Protected members

- **virtual void handleBeamline (const FlaggedElmPtr&)**
  The operation to be done for beamlines.
- **virtual void handleElement (const FlaggedElmPtr&)**
  The operation to be done for elements.
- **TBeamline<Member>& itsTable**
  The flat list to be filled.

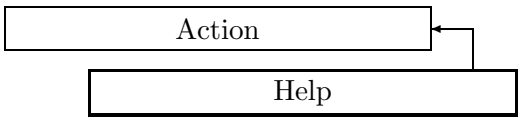


Figure 2.59: Inheritance for class Help

2.83 class Help

The HELP commands.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Help.hh

Synopsis (including inherited members)

Public members

- **Help ()**
  Exemplar constructor. (see Section 2.83.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Help* clone (const string&)**
  Make clone. (see Section 2.83.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.83.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.83.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Help ()**
  (see Section 2.83.1)

2.83.1 Detailed descriptions

Public members

- **Help ()**
  Exemplar constructor.
- **virtual Help* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Help ()**

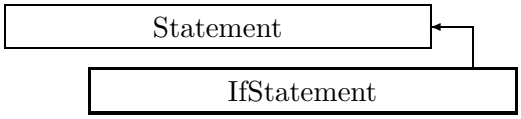


Figure 2.60: Inheritance for class IfStatement

2.84 class IfStatement

If statement.
A statement of the form "IF (<condition>) <statement>".

Type:	Instantiable
Superclasses:	public Statement
Include file:	./MadParser/IfStatement.hh

Synopsis (including inherited members)

Public members

- **IfStatement (const Parser&,TokenStream&)**
  Constructor. (see Section 2.84.1)
- **virtual void execute (const Parser&)**
  Execute. (see Section 2.84.1)
- **virtual ~IfStatement ()**
  (see Section 2.84.1)

2.84.1 Detailed descriptions

Public members

- **IfStatement (const Parser&,TokenStream&)**
  Constructor.

  Parse the statement on the given token stream, using the given parser.
- **virtual void execute (const Parser&)**
  Execute.

  Use the given parser to execute the controlled statements.
- **virtual ~IfStatement ()**

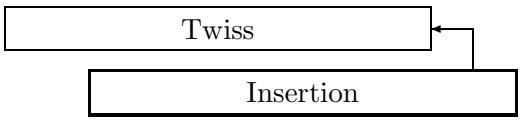


Figure 2.61: Inheritance for class Insertion

2.85 class Insertion

The TWISSTRACK command.

Type:	Instantiable
Superclasses:	public Twiss
Include file:	./Tables/Insertion.hh

Synopsis (including inherited members)

Public members

- **struct Cell**
  Descriptor for printing a table cell. (see Section 2.187.1)
- **typedef std::vector<Cell> CellArray**
  An array of cell descriptors. (see Section 2.187.1)
- **Insertion ()**
  Exemplar construction. (see Section 2.85.1)
- **class Row**
  Structure for a row of the Twiss table. (see Section 2.205.1)
- **typedef TBeamline<Row> TLine**
  (see Section 2.205.1)
- **TLine::const_iterator begin ()const**
  Access to first row. (see Section 2.205.1)
- **TLine::iterator begin ()**
  Access to first row. (see Section 2.205.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.187.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Insertion* clone (const string&)**
  Make clone. (see Section 2.85.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)

- **virtual void doomPut (DoomWriter&)const**
  Write the table to the DOOM data base. (see Section 2.205.1)
- **virtual void doomSummary (DoomWriter&)const**
  Fill in summary record. (see Section 2.85.1)
- **TLine::const_iterator end ()const**
  Access to last row. (see Section 2.205.1)
- **TLine::iterator end ()**
  Access to last row. (see Section 2.205.1)
- **virtual void execute ()**
  Check validity of the table definition. (see Section 2.205.1)
- **virtual void fill ()**
  Fill the buffer using the defined algorithm. (see Section 2.85.1)
- **static Table* find (const string&)**
  Find named Table. (see Section 2.187.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **double getALFi (const Row&,int,int)const**
  (see Section 2.205.1)
- **double getALFik (const Row&,int,int)const**
  Mais-Ripken alpha functions. (see Section 2.205.1)
- **double getBETi (const Row&,int,int)const**
  (see Section 2.205.1)
- **double getBETik (const Row&,int,int)const**
  Mais-Ripken beta functions. (see Section 2.205.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **double getCO (const Row&,int,int)const**
  Closed orbit. (see Section 2.205.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.187.1)
- **virtual double getCell (const PlaceRep&,const string&)**
  Return a selected value in a selected row. (see Section 2.205.1)
- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column **col** of this table, limited by **range**. (see Section 2.205.1)

- **FMatrix<double,6,6> getCurlyA ()const**
  Return initial curly A matrix. (see Section 2.205.1)
- **FMatrix<double,6,6> getCurlyA (const Row&)const**
  Curly A map for given row. (see Section 2.205.1)
- **const Row& getCurrent ()const**
  Return current table row in iteration. (see Section 2.205.1)
- **virtual CellArray getDefault ()const**
  Return the default print columns. (see Section 2.205.1)
- **double getDisp (const Row&,int,int)const**
  Dispersion. (see Section 2.205.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **double getET ()const**
  Return emittance for mode 3. (see Section 2.205.1)
- **double getEX ()const**
  Return emittance for mode 1. (see Section 2.205.1)
- **double getEY ()const**
  Return emittance for mode 2. (see Section 2.205.1)
- **double getEigen (const Row&,int,int)const**
  Eigenvectors. (see Section 2.205.1)
- **double getGAMik (const Row&,int,int)const**
  Mais-Ripken gamma functions. (see Section 2.205.1)
- **virtual double getLength ()**
  Return the length of the table. (see Section 2.205.1)
- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline. (see Section 2.205.1)
- **double getMUi (const Row&,int,int)const**
  Three modes, "naive" Twiss functions. (see Section 2.205.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **FMatrix<double,6,6> getMatrix (const Row&)const**
  Accumulated transfer map. (see Section 2.205.1)
- **double getMatrix (const Row&,int,int)const**
  Transfer matrix. (see Section 2.205.1)
- **FVector<double,6> getOrbit ()const**
  Return initial closed orbit. (see Section 2.205.1)

- **FVector<double,6> getOrbit (const Row&)const**
  Get orbit in given row. (see Section 2.205.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row, possible user-defined. (see Section 2.205.1)
- **double getS (const Row&,int,int)const**
  Arc length for given row. (see Section 2.205.1)
- **FMatrix<double,6,6> getSigma ()const**
  Initial envelope (Sigma) matrix. (see Section 2.205.1)
- **FMatrix<double,6,6> getSigma (const Row&)const**
  Envelope (Sigma) matrix for given row. (see Section 2.205.1)
- **double getSigma (const Row&,int,int)const**
  Sigma matrix. (see Section 2.205.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **virtual void invalidate ()**
  Mark this table as invalid, if it is dynamic. (see Section 2.187.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **virtual bool isDependent (const string&)const**
  Check dependency. (see Section 2.205.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return column expression. (see Section 2.205.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table. (see Section 2.85.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual bool matches (Table*)const**
  Check compatibility. (see Section 2.205.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print the table on an ASCII stream. (see Section 2.85.1)
- **void printTableBody (std::ostream&,const CellArray&)const**
  Print the body to this TWISS table. (see Section 2.205.1)
- **void printTableTitle (std::ostream&,const char*)const**
  Print standard information about the TWISS table. (see Section 2.205.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.187.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.187.1)
- **void tfsBody (std::ostream&,const CellArray&)const**
  Write TFS body for this table. (see Section 2.205.1)
- **void tfsSummary (std::ostream&)const**
  List TFS descriptors for table summary. (see Section 2.205.1)
- **void tfsTableDescriptors (std::ostream&)const**
  Write TFS descriptors existing for all tables. (see Section 2.187.1)
- **virtual void tfsTwissDescriptors (std::ostream&)const**
  Write TFS descriptors for a TWISS table. (see Section 2.85.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Insertion ()**
  (see Section 2.85.1)

2.85.1 Detailed descriptions

Public members

- **Insertion ()**
  Exemplar construction.
- **virtual Insertion* clone (const string&)**
  Make clone.
- **virtual void doomSummary (DoomWriter&)const**
  Fill in summary record.
- **virtual void fill ()**
  Fill the buffer using the defined algorithm.
- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table.
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print the table on an ASCII stream.

- **virtual void tfsTwissDescriptors (std::ostream&)const**
  Write TFS descriptors for a TWISS table.

  Writes the line and beam info, as well as length and tunes.

- **virtual ~Insertion ()**

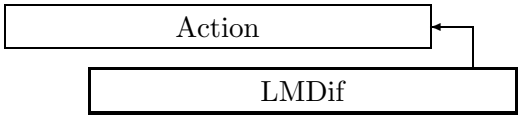


Figure 2.62: Inheritance for class LMDif

2.86 class LMDif

The LMDIF command.
This class encapsulates a minimisation according to the method used by Levenberg and Marquardt in

B. S. Garbow, K. E. Hillstrom, and J. J. More, User Guide for MINPACK-1, ANL 80-74.
Also described in

J. F. Bonnans et al., Optimisation Numerique, pp. 70-72. Springer, Berlin, 1997
Algorithm rewritten following the MINPACK package.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/LMDif.hh

Synopsis (including inherited members)

Public members

- **LMDif ()**
  Exemplar constructor. (see Section 2.86.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual LMDif* clone (const string&)**
  Make clone. (see Section 2.86.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.86.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)

- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)

- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~LMDif ()**
  (see Section 2.86.1)

2.86.1 Detailed descriptions

Public members

- **LMDif ()**
  Exemplar constructor.
- **virtual LMDif* clone (const string&)**
  Make clone.

- **virtual void execute ()**
  Execute the command.
- **virtual ~LMDif ()**

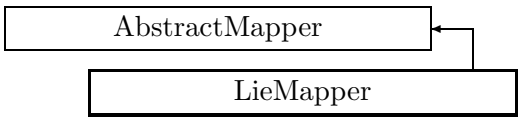


Figure 2.63: Inheritance for class LieMapper

2.87 class LieMapper

Build a Lie-algebraic map using a finite-length lens for each elements.
All maps are factored in ascending Dragt-Finn order.
Phase space coordinates numbering:

number	name	unit
0	x	metres
1	p_x/p_r	1
2	y	metres
3	p_y/p_r	1
4	$v * delta_t$	metres
5	$delta_p/p_r$	1

Where p_r is the constant reference momentum defining the reference frame velocity, m is the rest mass of the particles, and v is the instantaneous velocity of the particle.
Other units used:

quantity	unit
reference momentum	electron-volts
velocity	metres/second
accelerating voltage	volts
separator voltage	volts
frequencies	hertz
phase lags	$2 * pi$

All elements are represented by maps for finite-length elements.

Type:	Instantiable
Superclasses:	public AbstractMapper
Include file:	./Algorithms/LieMapper.hh

Synopsis (including inherited members)

Public members

- **LieMapper (const Beamline&,const PartData&,bool,bool,int)**
  Constructor. (see Section 2.87.1)
- **virtual void getMap (LinearMap<double,6>&)const**
  Return the linear part of the accumulated map. (see Section 2.87.1)
- **virtual void getMap (FVps<double,6>&)const**
  Return the accumulated map. (see Section 2.87.1)
- **virtual void getMap (DragtFinnMap<3>&)const**
  Return the full map accumulated so far. (see Section 2.87.1)

- **virtual void setMap (const LinearMap<double,6>&)**
  Reset the linear part of the accumulated map for restart. (see Section 2.87.1)
- **virtual void setMap (const FVps<double,6>&)**
  Reset the accumulated map for restart. (see Section 2.87.1)
- **virtual void setMap (const DragtFinnMap<3>&)**
  Reset the full map for restart. (see Section 2.87.1)
- **virtual void visitBeamBeam (const BeamBeam&)**
  Apply the algorithm to a BeamBeam. (see Section 2.87.1)
- **virtual void visitCollimator (const Collimator&)**
  Apply the algorithm to a Collimator. (see Section 2.87.1)
- **virtual void visitCorrector (const Corrector&)**
  Apply the algorithm to a Corrector. (see Section 2.87.1)
- **virtual void visitDiagnostic (const Diagnostic&)**
  Apply the algorithm to a Diagnostic. (see Section 2.87.1)
- **virtual void visitDrift (const Drift&)**
  Apply the algorithm to a Drift. (see Section 2.87.1)
- **virtual void visitLambertson (const Lambertson&)**
  Apply the algorithm to a Lambertson. (see Section 2.87.1)
- **virtual void visitMarker (const Marker&)**
  Apply the algorithm to a Marker. (see Section 2.87.1)
- **virtual void visitMonitor (const Monitor&)**
  Apply the algorithm to a Monitor. (see Section 2.87.1)
- **virtual void visitMultipole (const Multipole&)**
  Apply the algorithm to a Multipole. (see Section 2.87.1)
- **virtual void visitRBend (const RBend&)**
  Apply the algorithm to a RBend. (see Section 2.87.1)
- **virtual void visitRFCavity (const RFCavity&)**
  Apply the algorithm to a RFCavity. (see Section 2.87.1)
- **virtual void visitRFQuadrupole (const RFQuadrupole&)**
  Apply the algorithm to a RFQuadrupole. (see Section 2.87.1)
- **virtual void visitSBend (const SBend&)**
  Apply the algorithm to a SBend. (see Section 2.87.1)
- **virtual void visitSeparator (const Separator&)**
  Apply the algorithm to a Separator. (see Section 2.87.1)
- **virtual void visitSeptum (const Septum&)**
  Apply the algorithm to a Septum. (see Section 2.87.1)

- **virtual void visitSolenoid (const Solenoid&)**
  Apply the algorithm to a Solenoid. (see Section 2.87.1)
- **virtual ~LieMapper ()**
  (see Section 2.87.1)

2.87.1 Detailed descriptions

Public members

- **LieMapper (const Beamline&,const PartData&,bool,bool,int)**
  Constructor.

  The beam line to be tracked is "bl". The particle reference data are taken from "data". If **revBeam** is true, the beam runs from s = C to s = 0. If **revTrack** is true, we track against the beam.
- **virtual void getMap (LinearMap<double,6>&)const**
  Return the linear part of the accumulated map.
- **virtual void getMap (FVps<double,6>&)const**
  Return the accumulated map.
- **virtual void getMap (DragtFinnMap<3>&)const**
  Return the full map accumulated so far.
- **virtual void setMap (const LinearMap<double,6>&)**
  Reset the linear part of the accumulated map for restart.
- **virtual void setMap (const FVps<double,6>&)**
  Reset the accumulated map for restart.
- **virtual void setMap (const DragtFinnMap<3>&)**
  Reset the full map for restart.
- **virtual void visitBeamBeam (const BeamBeam&)**
  Apply the algorithm to a BeamBeam.
- **virtual void visitCollimator (const Collimator&)**
  Apply the algorithm to a Collimator.
- **virtual void visitCorrector (const Corrector&)**
  Apply the algorithm to a Corrector.
- **virtual void visitDiagnostic (const Diagnostic&)**
  Apply the algorithm to a Diagnostic.
- **virtual void visitDrift (const Drift&)**
  Apply the algorithm to a Drift.
- **virtual void visitLambertson (const Lambertson&)**
  Apply the algorithm to a Lambertson.
- **virtual void visitMarker (const Marker&)**
  Apply the algorithm to a Marker.

- **virtual void visitMonitor (const Monitor&)**
  Apply the algorithm to a Monitor.
- **virtual void visitMultipole (const Multipole&)**
  Apply the algorithm to a Multipole.
- **virtual void visitRBend (const RBend&)**
  Apply the algorithm to a RBend.
- **virtual void visitRFCavity (const RFCavity&)**
  Apply the algorithm to a RFCavity.
- **virtual void visitRFQuadrupole (const RFQuadrupole&)**
  Apply the algorithm to a RFQuadrupole.
- **virtual void visitSBend (const SBend&)**
  Apply the algorithm to a SBend.
- **virtual void visitSeparator (const Separator&)**
  Apply the algorithm to a Separator.
- **virtual void visitSeptum (const Septum&)**
  Apply the algorithm to a Septum.
- **virtual void visitSolenoid (const Solenoid&)**
  Apply the algorithm to a Solenoid.
- **virtual ~LieMapper ()**

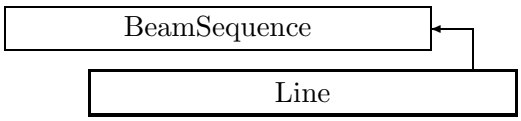


Figure 2.64: Inheritance for class Line

2.88 class Line

The LINE definition.

Type:	Instantiable
Superclasses:	public BeamSequence
Include file:	./Lines/Line.hh

Synopsis (including inherited members)

Public members

- **Line ()**
  Exemplar constructor. (see Section 2.88.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Line* clone (const string&)**
  Make clone. (see Section 2.88.1)
- **virtual Line* copy (const string&)**
  Make complete copy. (see Section 2.88.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read the line from the DOOM data base. (see Section 2.88.1)
- **virtual void doomPut (DoomWriter&)const**
  Write the line to the DOOM data base. (see Section 2.88.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Beamline* fetchLine ()const**
  Return the embedded CLASSIC beam line. (see Section 2.27.1)
- **static BeamSequence* find (const string&)**
  Find a BeamSequence by name. (see Section 2.27.1)

- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.27.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return line length. (see Section 2.88.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)

- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a line template. (see Section 2.88.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the line object. (see Section 2.88.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the line. (see Section 2.88.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)

- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Line ()**
  (see Section 2.88.1)

2.88.1 Detailed descriptions

Public members

- **Line ()**
  Exemplar constructor.
- **virtual Line* clone (const string&)**
  Make clone.

  The new object is an empty line, it will be filled by the parser.
- **virtual Line* copy (const string&)**
  Make complete copy.

  Copy also the line list.
- **virtual void doomGet (const DoomReader&)**
  Read the line from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write the line to the DOOM data base.
- **virtual double getLength ()const**
  Return line length.
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a line template.

  The template gets the name **name**, **is** is ignored, and the formals and the line list are read from **stat**.
- **virtual void parse (Statement&)**
  Parse the line object.

  Read the definition from **stat**.
- **virtual void print (std::ostream&)const**
  Print the line.
- **virtual ~Line ()**

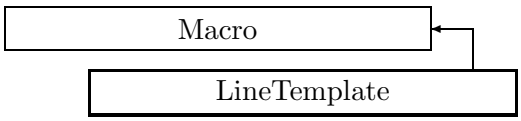


Figure 2.65: Inheritance for class LineTemplate

2.89 class LineTemplate

An "archetype" for a MAD beam line with arguments.

Type:	Instantiable
Superclasses:	public Macro
Include file:	./Lines/LineTemplate.hh

Synopsis (including inherited members)

Public members

- **LineTemplate ()**
  (see Section 2.89.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual LineTemplate* clone (const string&)**
  Make clone. (see Section 2.89.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.98.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Make line instance. (see Section 2.89.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a line template. (see Section 2.89.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (TokenStream&,Statement&)**
  Parse the line template. (see Section 2.89.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseActuals (Statement&)**
  Parse actual arguments. (see Section 2.98.1)
- **virtual void parseFormals (Statement&)**
  Parse formal arguments. (see Section 2.98.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)

- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.98.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.98.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~LineTemplate ()**
  (see Section 2.89.1)

2.89.1 Detailed descriptions

Public members

- **LineTemplate ()**

- **virtual LineTemplate* clone (const string&)**
  Make clone.

  Throw MadException, since the template cannot be cloned.

- **virtual Object* makeInstance (const string&,Statement&)**
  Make line instance.

  The instance gets the name **name**, and its actual arguments are read from **stat**.

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a line template.

  Return NULL, since one cannot make a template from a template.

- **virtual void parse (TokenStream&,Statement&)**
  Parse the line template.

  Read the actual arguments from **stat**. **is** is not used.

- **virtual ~LineTemplate ()**

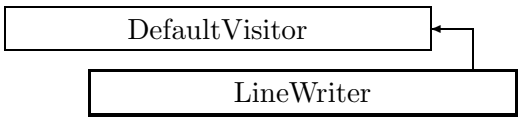


Figure 2.66: Inheritance for class LineWriter

2.90 class LineWriter

Visitor class for writing a sequence list to the DOOM data base.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Lines/LineWriter.hh

Synopsis (including inherited members)

Public members

- **LineWriter (const Beamline&,const string&,const string&)**
  (see Section 2.90.1)
- **virtual void visitBeamline (const Beamline&)**
  Override beamline entrance. (see Section 2.90.1)
- **virtual void visitDrift (const Drift&)**
  Override drift operation. (see Section 2.90.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply visitor to FlaggedElmPtr. (see Section 2.90.1)
- **virtual void visitMapIntegrator (const MapIntegrator&)**
  Override beamline exit. (see Section 2.90.1)
- **~LineWriter ()**
  (see Section 2.90.1)

Protected members

- **virtual void applyDefault (const ElementBase&)**
  Apply default. (see Section 2.90.1)

2.90.1 Detailed descriptions

Public members

- **LineWriter (const Beamline&,const string&,const string&)**

- **virtual void visitBeamline (const Beamline&)**
  Override beamline entrance.

- **virtual void visitDrift (const Drift&)**
  Override drift operation.

  Accumulate length, but do not save anything.

- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply visitor to FlaggedElmPtr.

  Makes sure the selection flag is set.

- **virtual void visitMapIntegrator (const MapIntegrator&)**
  Override beamline exit.

- **~LineWriter ()**

Protected members

- **virtual void applyDefault (const ElementBase&)**
  Apply default.

  Add the element to the DOOM list.

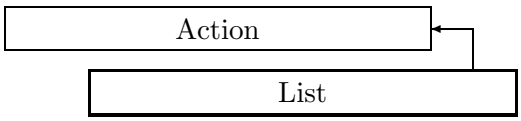


Figure 2.67: Inheritance for class List

2.91 class List

The LIST command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Tables/List.hh

Synopsis (including inherited members)

Public members

- **List ()**
  Exemplar constructor. (see Section 2.91.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual List* clone (const string&)**
  Make clone. (see Section 2.91.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.91.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~List ()**
  (see Section 2.91.1)

2.91.1 Detailed descriptions

Public members

- **List ()**
  Exemplar constructor.
- **virtual List* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~List ()**

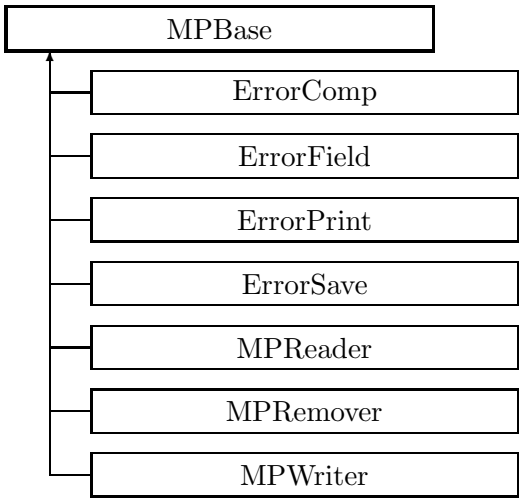


Figure 2.68: Inheritance for class MPBase

2.92 class MPBase

Handle multipole errors.
An abstract mixin class, used for all classes which must have access to field errors for setting or retrieving.

Type:	abstract
Include file:	./Errors/MPBase.hh

Synopsis (including inherited members)

Public members

- **MPBase ()**
  (see Section 2.92.1)
- **static double absFactor (int)**
  Scale factor for absolute error components. (see Section 2.92.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Deal with error field. (see Section 2.92.1)
- **static double relFactor (int,int,double)**
  Scale factor for relative error components. (see Section 2.92.1)
- **virtual ~MPBase ()**
  (see Section 2.92.1)

2.92.1 Detailed descriptions

Public members

- **MPBase ()**

- **static double absFactor (int)**
  Scale factor for absolute error components.

  Return $(p0/c)/i!$.

- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Deal with error field.

  This is the interface method "mixed in".

- **static double relFactor (int,int,double)**
  Scale factor for relative error components.

  Return $r^(o-c)$.

- **virtual ~MPBase ()**

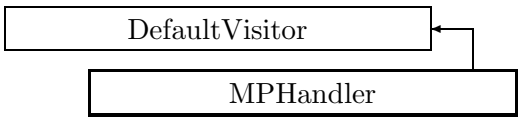


Figure 2.69: Inheritance for class MPHandler

2.93 class MPHandler

Access field errors.
A "Visitor" class giving access to the field errors. It calls the "MPBase" for each element which has (may have) a field error.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Errors/MPHandler.hh

Synopsis (including inherited members)

Public members

- **MPHandler (const Beamline&,MPBase&,bool)**
  Constructor. (see Section 2.93.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply visitor to FlaggedElmPtr. (see Section 2.93.1)
- **virtual void visitMultipoleWrapper (const MultipoleWrapper&)**
  Apply visitor to MultipoleWrapper. (see Section 2.93.1)
- **virtual void visitRBendWrapper (const RBendWrapper&)**
  Apply visitor to RBendWrapper. (see Section 2.93.1)
- **virtual void visitSBendWrapper (const SBendWrapper&)**
  Apply visitor to SBendWrapper. (see Section 2.93.1)
- **virtual ~MPHandler ()**
  (see Section 2.93.1)

2.93.1 Detailed descriptions

Public members

- **MPHandler (const Beamline&,MPBase&,bool)**
  Constructor.

  Set up the visitor to use the given beam line **bl** and to apply the command **base** to its contained elements. If **full** is true, apply to all elements, otherwise only to selected elemens.
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply visitor to FlaggedElmPtr.

  Makes sure the selection flag is set.

- **virtual void visitMultipoleWrapper (const MultipoleWrapper&)**
  Apply visitor to MultipoleWrapper.
- **virtual void visitRBendWrapper (const RBendWrapper&)**
  Apply visitor to RBendWrapper.
- **virtual void visitSBendWrapper (const SBendWrapper&)**
  Apply visitor to SBendWrapper.
- **virtual ~MPHandler ()**

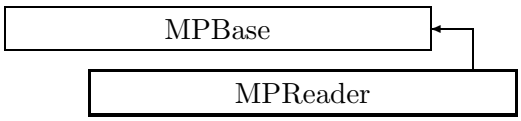


Figure 2.70: Inheritance for class MPReader

2.94 class MPReader

A DOOM reader for reading field errors.
Allows retriving of field errors. Can be applied via an MPHandler to a beam line.

Type:	Instantiable
Superclasses:	public MPBase
Include file:	./Errors/MPReader.hh

Synopsis (including inherited members)

Public members

- **MPReader (const string&)**
  Constructor. (see Section 2.94.1)
- **static double absFactor (int)**
  Scale factor for absolute error components. (see Section 2.92.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Handle field error. (see Section 2.94.1)
- **static double relFactor (int,int,double)**
  Scale factor for relative error components. (see Section 2.92.1)
- **virtual ˜MPReader ()**
  (see Section 2.94.1)

2.94.1 Detailed descriptions

Public members

- **MPReader (const string&)**
  Constructor.

  Store **name** in the DOOM environment.
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Handle field error.
- **virtual ˜MPReader ()**

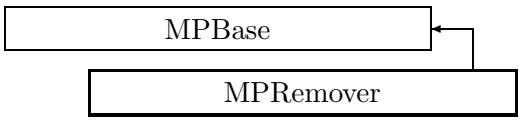


Figure 2.71: Inheritance for class MPRemover

2.95 class MPRemover

A visitor used to remove all random field errors from a line.
Can be applied via an MPHandler to a beam line.

Type:	Instantiable
Superclasses:	public MPBase
Include file:	./Errors/MPRemover.hh

Synopsis (including inherited members)

Public members

- **MPRemover ()**
  (see Section 2.95.1)
- **static double absFactor (int)**
  Scale factor for absolute error components. (see Section 2.92.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Remove error field. (see Section 2.95.1)
- **static double relFactor (int,int,double)**
  Scale factor for relative error components. (see Section 2.92.1)
- **virtual ˜MPRemover ()**
  (see Section 2.95.1)

2.95.1 Detailed descriptions

Public members

- **MPRemover ()**
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Remove error field.
- **virtual ˜MPRemover ()**

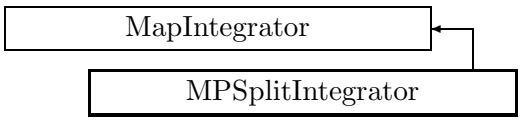


Figure 2.72: Inheritance for class MPSplitIntegrator

2.96 class MPSplitIntegrator

Integrator replacing each multipole by a set of thin lenses.
Phase space coordinates numbering:

number	name	unit
0	x	metres
1	p_x/p_r	1
2	y	metres
3	p_y/p_r	1
4	v*delta_t	metres
5	delta_p/p_r	1

Where p_r is the constant reference momentum defining the reference frame velocity, m is the rest mass of the particles, and v is the instantaneous velocity of the particle.
Other units used:

quantity	unit
reference momentum	electron-volts
velocity	metres/second
accelerating voltage	volts
separator voltage	volts
frequencies	hertz
phase lags	$2 * pi$

A MPSplitIntegrator performs integration through an element using two thin lenses of force 1/2, one placed at 1/6 and the other at 5/6 of the length respectively.

Type:	Instantiable
Superclasses:	public MapIntegrator
Include file:	./Algorithms/MPSplitIntegrator.hh

Synopsis (including inherited members)

Public members

- **MPSplitIntegrator (Multipole*,int)**
  Constructor. (see Section 2.96.1)
- **MPSplitIntegrator (const MPSplitIntegrator&)**
  (see Section 2.96.1)
- **virtual MPSplitIntegrator* clone ()const**
  Make clone. (see Section 2.96.1)
- **virtual Geometry& getGeometry ()**
  Get geometry. (see Section 2.96.1)

- **virtual const Geometry& getGeometry ()const**
  Get geometry. (see Section 2.96.1)
- **virtual void getMap (FVps<double,6>&,const PartData&,bool,bool)const**
  Get map from MPSplitIntegrator. (see Section 2.96.1)
- **void getSlices (std::vector<double>&)const**
  Return slice positions. (see Section 2.96.1)
- **virtual const string& getType ()const**
  Get element type string. (see Section 2.96.1)
- **virtual void trackBunch (PartBunch&,const PartData&,bool,bool)const**
  Track particle bunch through MPSplitIntegrator. (see Section 2.96.1)
- **virtual void trackMap (FVps<double,6>&,const PartData&,bool,bool)const**
  Track map through MPSplitIntegrator. (see Section 2.96.1)
- **virtual void trackParticle (Particle&,const PartData&,bool,bool)const**
  Track particle through MPSplitIntegrator. (see Section 2.96.1)
- **virtual ~MPSplitIntegrator ()**
  (see Section 2.96.1)

2.96.1 Detailed descriptions

Public members

- **MPSplitIntegrator (Multipole*,int)**
  Constructor.

  Attach this integrator to the given Multipole, using **slices** subdivisions.
- **MPSplitIntegrator (const MPSplitIntegrator&)**

- **virtual MPSplitIntegrator* clone ()const**
  Make clone.
- **virtual Geometry& getGeometry ()**
  Get geometry.

  Return the element geometry. Version for non-constant object.
- **virtual const Geometry& getGeometry ()const**
  Get geometry.

  Return the element geometry Version for constant object.
- **virtual void getMap (FVps<double,6>&,const PartData&,bool,bool)const**
  Get map from MPSplitIntegrator.

  The map is returned in **map**, the other values are the same as in the calling mapper.

- **void getSlices (std::vector<double>&)const**
  Return slice positions.

  Build a vector **v** containing the longitudinal positions of the thin lens slices.

- **virtual const string& getType ()const**
  Get element type string.

- **virtual void trackBunch (PartBunch&,const PartData&,bool,bool)const**
  Track particle bunch through MPSplitIntegrator.

  The bunch tracked is **buch**, the other values are the same as in the calling mapper.

- **virtual void trackMap (FVps<double,6>&,const PartData&,bool,bool)const**
  Track map through MPSplitIntegrator.

  The map tracked is **map**, the other values are the same as in the calling mapper.

- **virtual void trackParticle (Particle&,const PartData&,bool,bool)const**
  Track particle through MPSplitIntegrator.

  The particle tracked is **part**, the other values are the same as in the calling mapper.

- **virtual ~MPSplitIntegrator ()**

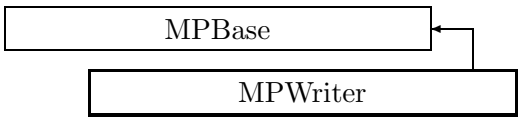


Figure 2.73: Inheritance for class MPWriter

2.97 class MPWriter

A DOOM writer for writing field errors.
Allows storing of field errors. Can be applied via an MPHandler to a beam line.

Type:	Instantiable
Superclasses:	public MPBase
Include file:	./Errors/MPWriter.hh

Synopsis (including inherited members)

Public members

- **MPWriter (const string&)**
  Constructor. (see Section 2.97.1)
- **static double absFactor (int)**
  Scale factor for absolute error components. (see Section 2.92.1)
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Handle error field. (see Section 2.97.1)
- **static double relFactor (int,int,double)**
  Scale factor for relative error components. (see Section 2.92.1)
- **virtual ˜MPWriter ()**
  (see Section 2.97.1)

2.97.1 Detailed descriptions

Public members

- **MPWriter (const string&)**
  Constructor.

  Store **name** in the DOOM environment.
- **virtual void fieldError (const string&,int,const BMultipoleField&,BMultipoleField&)**
  Handle error field.
- **virtual ˜MPWriter ()**

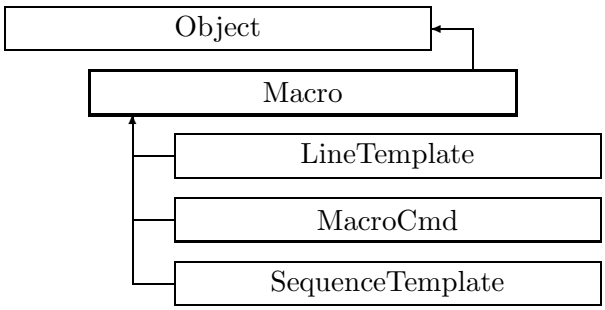


Figure 2.74: Inheritance for class Macro

2.98 class Macro

Abstract base class for macros.
The base class for storing the "archetype" for all macro-like commands.

Type:	abstract
Superclasses:	public Object
Include file:	./MadParser/Macro.hh

Synopsis (including inherited members)

Public members

- **Macro (int,const char*,const char*)**
  Exemplar constructor. (see Section 2.98.1)
- **Macro (const string&,Object*)**
  Clone constructor. (see Section 2.98.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Macro* clone (const string&)**
  Make clone. (see Section 2.98.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)

- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.98.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Make a macro instance. (see Section 2.98.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a macro template. (see Section 2.98.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)

- **virtual void parseActuals (Statement&)**
  Parse actual arguments. (see Section 2.98.1)
- **virtual void parseFormals (Statement&)**
  Parse formal arguments. (see Section 2.98.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.98.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.98.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Macro ()**
  (see Section 2.98.1)

Protected members

- **std::vector<std::vector<Token> > actuals**
  The actual argument list. (see Section 2.98.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **std::vector<string> formals**
  The formal argument list. (see Section 2.98.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.98.1 Detailed descriptions

Public members

- **Macro (int,const char*,const char*)**
  Exemplar constructor.
- **Macro (const string&,Object*)**
  Clone constructor.
- **virtual Macro* clone (const string&)**
  Make clone.

  Throw ParseError, since for macro we must make a template and not a clone.
- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "MACRO".
- **virtual Object* makeInstance (const string&,Statement&)**
  Make a macro instance.

  If **this** is a macro template, make an instance.
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a macro template.

  If **this** is a macro exemplar, make a template.
- **virtual void parseActuals (Statement&)**
  Parse actual arguments.

  Expects parse pointer in the statement to be set on the first argument.
- **virtual void parseFormals (Statement&)**
  Parse formal arguments.

  Expects parse pointer in the statement to be set on the first argument.

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always false for macros.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always false for macros.

- **virtual ~Macro ()**

Protected members

- **std::vector<std::vector<Token> > actuals**
  The actual argument list.

  Each actual argument is stored as a vector of tokens.

- **std::vector<string> formals**
  The formal argument list.

  Each formal argument is represented as a name and stored as a string.

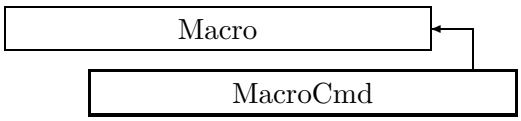


Figure 2.75: Inheritance for class MacroCmd

2.99 class MacroCmd

Encapsulate the buffer for the "archetypes" of all macros.

Type:	Instantiable
Superclasses:	public Macro
Include file:	./MadParser/MacroCmd.hh

Synopsis (including inherited members)

Public members

- **MacroCmd ()**
  (see Section 2.99.1)
- **MacroCmd (const string&,MacroCmd*)**
  (see Section 2.99.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Macro* clone (const string&)**
  Make clone. (see Section 2.98.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the macro command. (see Section 2.99.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.98.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Make a macro instance. (see Section 2.99.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a macro template. (see Section 2.99.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseActuals (Statement&)**
  Parse actual arguments. (see Section 2.98.1)
- **virtual void parseFormals (Statement&)**
  Parse formal arguments. (see Section 2.98.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)

- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.98.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.98.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~MacroCmd ()**
  (see Section 2.99.1)

2.99.1 Detailed descriptions

Public members

- **MacroCmd ()**

- **MacroCmd (const string&,MacroCmd*)**

- **virtual void execute ()**
  Execute the macro command.

- **virtual Object* makeInstance (const string&,Statement&)**
  Make a macro instance.

  Expects parse pointer in the statement to be set on the first argument.

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a macro template.

  Expects parse pointer in the statement to be set on the first argument.

- **virtual ~MacroCmd ()**

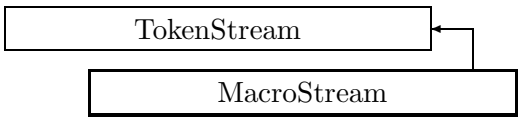


Figure 2.76: Inheritance for class MacroStream

2.100 class MacroStream

An input buffer for macro commands.

Type:	Instantiable
Superclasses:	public TokenStream
Include file:	./MadParser/MacroStream.hh

Synopsis (including inherited members)

Public members

- **MacroStream (const string&)**
  Constructor. (see Section 2.100.1)
- **typedef std::list<Token> TokenList**
  (see Section 2.100.1)
- **void append (Token&)**
  Append a token to the stream. (see Section 2.100.1)
- **virtual Token readToken ()**
  Read a token from the stream. (see Section 2.100.1)
- **void start ()**
  Reset stream to start. (see Section 2.100.1)
- **virtual ~MacroStream ()**
  (see Section 2.100.1)

2.100.1 Detailed descriptions

Public members

- **MacroStream (const string&)**
  Constructor.

  Assign the macro name as a buffer name.
- **typedef std::list<Token> TokenList**
- **void append (Token&)**
  Append a token to the stream.
- **virtual Token readToken ()**
  Read a token from the stream.

- **void start ()**
  Reset stream to start.
- **virtual ~MacroStream ()**

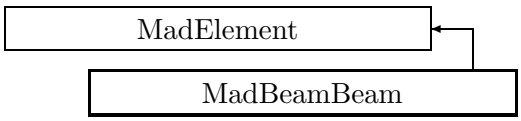


Figure 2.77: Inheritance for class MadBeamBeam

2.101 class MadBeamBeam

The BEAMBEAM element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadBeamBeam.hh

Synopsis (including inherited members)

Public members

- **MadBeamBeam ()**
  The attributes of class MadBeamBeam. Exemplar constructor. (see Section 2.101.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadBeamBeam* clone (const string&)**
  Make clone. (see Section 2.101.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.101.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC beam-beam element. (see Section 2.101.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadBeamBeam ()**
  (see Section 2.101.1)

2.101.1 Detailed descriptions

Public members

- **MadBeamBeam ()**
  The attributes of class MadBeamBeam. Exemplar constructor.
- **virtual MadBeamBeam* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC beam-beam element.
- **virtual ~MadBeamBeam ()**

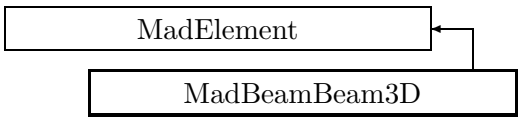


Figure 2.78: Inheritance for class MadBeamBeam3D

2.102 class MadBeamBeam3D

The BEAMINT element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadBeamBeam3D.hh

Synopsis (including inherited members)

Public members

- **MadBeamBeam3D ()**
  The attributes of class MadBeamBeam3D. Exemplar constructor. (see Section 2.102.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadBeamBeam3D* clone (const string&)**
  Make clone. (see Section 2.102.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.102.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC element. (see Section 2.102.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadBeamBeam3D ()**
  (see Section 2.102.1)

2.102.1 Detailed descriptions

Public members

- **MadBeamBeam3D ()**
  The attributes of class MadBeamBeam3D. Exemplar constructor.
- **virtual MadBeamBeam3D* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC element.
- **virtual ~MadBeamBeam3D ()**

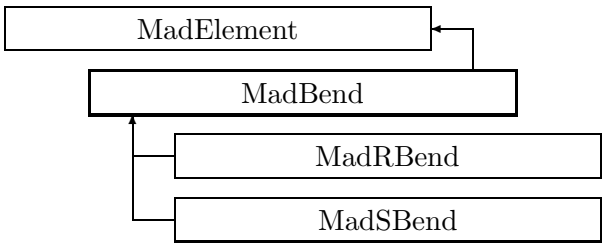


Figure 2.79: Inheritance for class MadBend

2.103 class MadBend

Base class for all bending magnets.
This class factors out the special behaviour for the DOOM interface and the printing in MAD-8 format, as well as the bend attributes.

Type:	abstract
Superclasses:	public MadElement
Include file:	./Elements/MadBend.hh

Synopsis (including inherited members)

Public members

- **MadBend (const char*,const char*)**
  The attribute numbers for MAD bend magnets. Exemplar constructor. (see Section 2.103.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write bend magnet to DOOM data base. (see Section 2.103.1)

- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)
- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)

- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the bend magnet. (see Section 2.103.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)

- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadBend ()**
  (see Section 2.103.1)

Protected members

- **MadBend (const string&,MadBend*)**
  Clone constructor. (see Section 2.103.1)
- **static std::map<string,OwnPtr<AttCell> > attributeRegistry**
  The registry for named attributes. (see Section 2.108.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)
- **static void printAttribute (std::ostream&,const string&,const string&,int&)**
  Print an attribute with a MAD-8 name (as an expression). (see Section 2.108.1)

- **static void printAttribute (std::ostream&,const string&,double,int&)**
  Print an attribute with a MAD-8 name (as a constant). (see Section 2.108.1)
- **static void printMultipoleStrength (std::ostream&,int,int&,const string&,const string&,const Attribute&,const Attribute&,const Attribute&)**
  Print multipole components in MAD-8 format. (see Section 2.108.1)
- **static AttCell* registerRealAttribute (const string&)**
  Register a "real" element attribute. (see Section 2.108.1)
- **static AttCell* registerStringAttribute (const string&)**
  Register a "string" element attribute. (see Section 2.108.1)

2.103.1 Detailed descriptions

Public members

- **MadBend (const char*,const char*)**
  The attribute numbers for MAD bend magnets. Exemplar constructor.
- **virtual void doomPut (DoomWriter&)const**
  Write bend magnet to DOOM data base.

  Special version required for write only.
- **virtual void print (std::ostream&)const**
  Print the bend magnet.

  Handle printing in MAD-8 format.
- **virtual ~MadBend ()**

Protected members

- **MadBend (const string&,MadBend*)**
  Clone constructor.

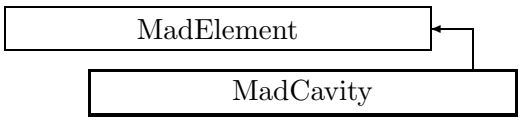


Figure 2.80: Inheritance for class MadCavity

2.104 class MadCavity

The RFCAVITY element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadCavity.hh

Synopsis (including inherited members)

Public members

- **MadCavity ()**
  The attributes of class MadCavity. Exemplar constructor. (see Section 2.104.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadCavity* clone (const string&)**
  Make clone. (see Section 2.104.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.104.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC cavity. (see Section 2.104.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadCavity ()**
  (see Section 2.104.1)

2.104.1 Detailed descriptions

Public members

- **MadCavity ()**
  The attributes of class MadCavity. Exemplar constructor.
- **virtual MadCavity* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC cavity.
- **virtual ~MadCavity ()**

2.105 class MadData

The global MAD structure.
The MAD object holds all global data required for a MAD execution.

Type:	Instantiable
Include file:	./AbstractObjects/MadData.hh

Synopsis (including inherited members)

Public members

- **struct ClearReference**
  Clear Reference. (see Section 2.105.1)
- **MadData ()**
  Constructor. (see Section 2.105.1)
- **void apply (const ObjectFunction&)**
  Apply a function to all objects. (see Section 2.105.1)
- **void create (Object*)**
  Create new object. (see Section 2.105.1)
- **void define (Object*)**
  Define a new object. (see Section 2.105.1)
- **void erase (const string&)**
  Delete existing entry. (see Section 2.105.1)
- **Object* find (const string&)**
  Find entry. (see Section 2.105.1)
- **double getP0 ()const**
  Return value of global reference momentum. (see Section 2.105.1)
- **void makeDirty (Object*)**
  Invalidate expressions. (see Section 2.105.1)
- **void printNames (std::ostream&,const string&)**
  Print all objects. (see Section 2.105.1)
- **void printTitle (std::ostream&)**
  Print the page title. (see Section 2.105.1)
- **void registerExpression (AttributeBase*)**
  Register expression. (see Section 2.105.1)
- **void registerTable (Table*)**
  Register table. (see Section 2.105.1)
- **void setP0 (ValueDefinition*)**
  Set the global momentum. (see Section 2.105.1)

- **void storeTitle (const string&)**
  Store the page title. (see Section 2.105.1)
- **void unregisterExpression (AttributeBase*)**
  Unregister expression. (see Section 2.105.1)
- **void unregisterTable (Table*)**
  Unregister table. (see Section 2.105.1)
- **void update ()**
  Update all objects. (see Section 2.105.1)
- **~MadData ()**
  (see Section 2.105.1)

2.105.1 Detailed descriptions

Public members

- **struct ClearReference**
  Clear Reference.

  This functor is used to clear the reference count stored in an object.
- **MadData ()**
  Constructor.

  Initialise MAD execution.
- **void apply (const ObjectFunction&)**
  Apply a function to all objects.

  Loop over the directory and apply the given functor object to each object in turn.
- **void create (Object*)**
  Create new object.

  No replacement is allowed; if an object with the same name exists, throw **MadException**.
- **void define (Object*)**
  Define a new object.

  Replacement is allowed; however **MadException** is thrown, if the replacement cannot be done.
- **void erase (const string&)**
  Delete existing entry.

  Identified by **name**.
- **Object* find (const string&)**
  Find entry.

  Identified by **name**.
- **double getP0 ()const**
  Return value of global reference momentum.

- **void makeDirty (Object*)**
  Invalidate expressions.

  Force re-evaluation of all expressions before next command is executed. Also set the **modified** flag in **object**, if not NULL.

- **void printNames (std::ostream&,const string&)**
  Print all objects.

  Loop over the directory and print each object whose name matches the regular expression **pattern**.

- **void printTitle (std::ostream&)**
  Print the page title.

- **void registerExpression (AttributeBase*)**
  Register expression.

  Registered expressions are invalidated to be recomputed when any object in the directory is changed or replaced.

- **void registerTable (Table*)**
  Register table.

  Register the table **t**. Registered tables are invalidated to be refilled when an object on which they depend is changed or replaced.

- **void setP0 (ValueDefinition*)**
  Set the global momentum.

- **void storeTitle (const string&)**
  Store the page title.

- **void unregisterExpression (AttributeBase*)**
  Unregister expression.

- **void unregisterTable (Table*)**
  Unregister table.

- **void update ()**
  Update all objects.

  Loop over the directory and notify all objects to update themselves.

- **~MadData ()**

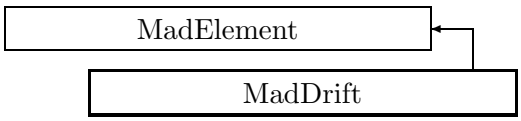


Figure 2.81: Inheritance for class MadDrift

2.106 class MadDrift

The DRIFT element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadDrift.hh

Synopsis (including inherited members)

Public members

- **MadDrift ()**
  Exemplar constructor. (see Section 2.106.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadDrift* clone (const string&)**
  Make clone. (see Section 2.106.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **virtual bool isDrift ()const**
  Test for drift. (see Section 2.106.1)

- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)

- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC drift. (see Section 2.106.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadDrift ()**
  (see Section 2.106.1)

2.106.1 Detailed descriptions

Public members

- **MadDrift ()**
  Exemplar constructor.
- **virtual MadDrift* clone (const string&)**
  Make clone.
- **virtual bool isDrift ()const**
  Test for drift.

  Return true.
- **virtual void update ()**
  Update the embedded CLASSIC drift.
- **virtual ~MadDrift ()**

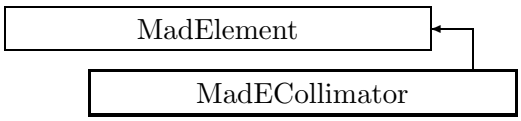


Figure 2.82: Inheritance for class MadECollimator

2.107 class MadECollimator

The ECOLLIMATOR element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadECollimator.hh

Synopsis (including inherited members)

Public members

- **MadECollimator ()**
  The attributes of class MadECollimator. Exemplar constructor. (see Section 2.107.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadECollimator* clone (const string&)**
  Make clone. (see Section 2.107.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.107.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC collimator. (see Section 2.107.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadECollimator ()**
  (see Section 2.107.1)

2.107.1 Detailed descriptions

Public members

- **MadECollimator ()**
  The attributes of class MadECollimator. Exemplar constructor.
- **virtual MadECollimator* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC collimator.
- **virtual ~MadECollimator ()**

2.108 class MadElement

Base class for all beam line elements.
This class factors out all special behaviour for the DOOM interface and the printing in MAD-8 format, as well as some common attributes. It defines a registry for attribute cells, used in the ATTLIST command. The class MadElement command stores all defined attribute values in this registry, and the ATTLIST command can look them up to build up a print line.

Type:	abstract
Superclasses:	public Element
Include file:	./Elements/MadElement.hh

Synopsis (including inherited members)

Public members

- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)
- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)

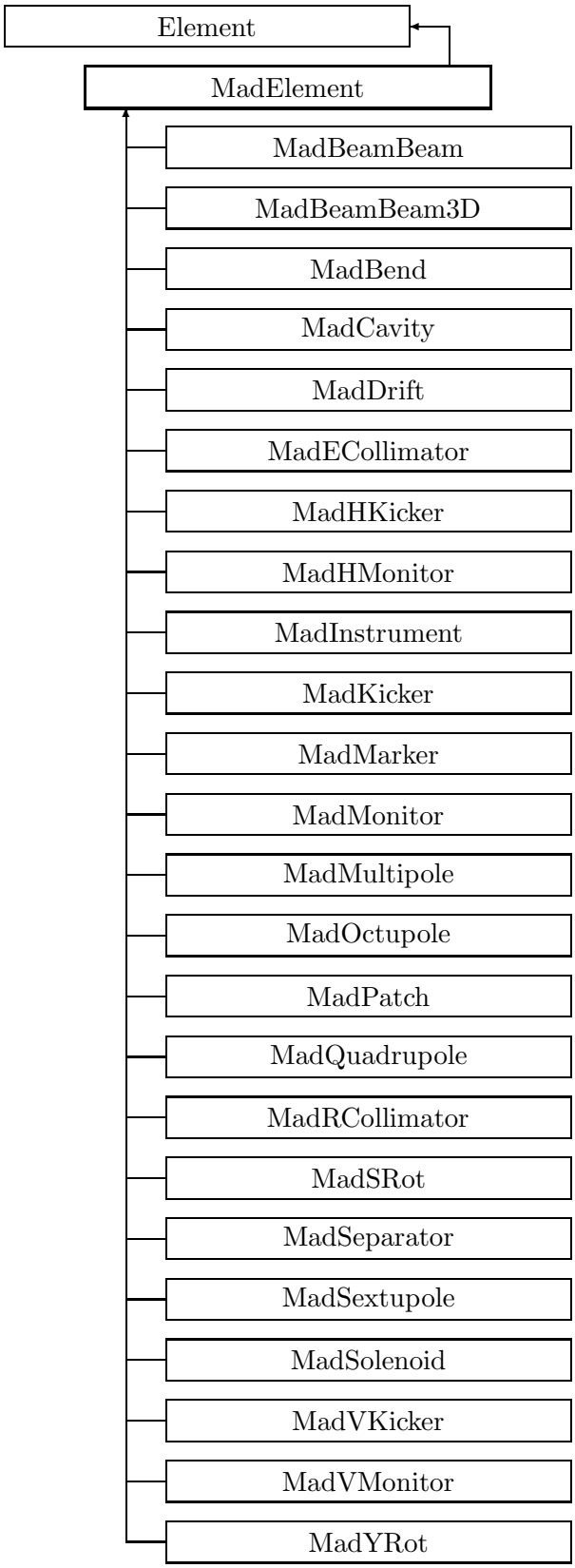


Figure 2.83: Inheritance for class MadElement

- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)

- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)

- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadElement ()**
  (see Section 2.108.1)

Protected members

- **MadElement (int,const char*,const char*)**
  Exemplar constructor. (see Section 2.108.1)
- **MadElement (const string&,MadElement*)**
  Clone constructor. (see Section 2.108.1)
- **static std::map<string,OwnPtr<AttCell> > attributeRegistry**
  The registry for named attributes. (see Section 2.108.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)
- **static void printAttribute (std::ostream&,const string&,const string&,int&)**
  Print an attribute with a MAD-8 name (as an expression). (see Section 2.108.1)
- **static void printAttribute (std::ostream&,const string&,double,int&)**
  Print an attribute with a MAD-8 name (as a constant). (see Section 2.108.1)
- **static void printMultipoleStrength (std::ostream&,int,int&,const string&,const string&,const Attribute&,const Attribute&,const Attribute&)**
  Print multipole components in MAD-8 format. (see Section 2.108.1)
- **static AttCell* registerRealAttribute (const string&)**
  Register a "real" element attribute. (see Section 2.108.1)
- **static AttCell* registerStringAttribute (const string&)**
  Register a "string" element attribute. (see Section 2.108.1)

2.108.1 Detailed descriptions

Public members

- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command.

- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base.

  Access the element attributes via the DOOM index table.

- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base.

  Access the element attributes via the DOOM index table.

- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.

- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute.

  Return a pointer to the AttCell for a named attribute.

- **virtual double getLength ()const**
  Return element length.

- **const string getTypeName ()const**
  Return the element's type name.

- **virtual void parse (Statement&)**
  Parse the element.

  This special version for elements handles unknown attributes by appending them to the attribute list.

- **virtual void print (std::ostream&)const**
  Print the object.

  This special version handles special printing in MAD-8 format.

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute.

- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute.

- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC.

- **virtual ~MadElement ()**

Protected members

- **MadElement (int,const char*,const char*)**
  Exemplar constructor.
- **MadElement (const string&,MadElement*)**
  Clone constructor.
- **static std::map<string,OwnPtr<AttCell> > attributeRegistry**
  The registry for named attributes.
- **static void printAttribute (std::ostream&,const string&,const string&,int&)**
  Print an attribute with a MAD-8 name (as an expression).
- **static void printAttribute (std::ostream&,const string&,double,int&)**
  Print an attribute with a MAD-8 name (as a constant).
- **static void printMultipoleStrength (std::ostream&,int,int&,const string&,const string&,const Attribute&,const Attribute&,const Attribute&)**
  Print multipole components in MAD-8 format.

  This function is accessible to all multipole-like elements (RBend, SBend, Quadrupole, Sextupole, Octupole, Multipole).
- **static AttCell* registerRealAttribute (const string&)**
  Register a "real" element attribute.

  A registered attribute can be listed by the ATTLIST command.
- **static AttCell* registerStringAttribute (const string&)**
  Register a "string" element attribute.

  A registered attribute can be listed by the ATTLIST command.

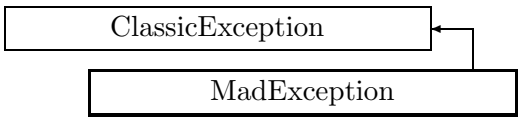


Figure 2.84: Inheritance for class MadException

2.109 class MadException

The base class for all MAD exceptions.

Type:	Instantiable
Superclasses:	public ClassicException
Include file:	./Utilities/MadException.hh

Synopsis (including inherited members)

Public members

- **MadException (const string&,const string&)**
  The usual constructor. (see Section 2.109.1)
- **MadException (const MadException&)**
  (see Section 2.109.1)
- **virtual ~MadException ()**
  (see Section 2.109.1)

2.109.1 Detailed descriptions

Public members

- **MadException (const string&,const string&)**
  The usual constructor.

  Arguments:

  meth the name of the method or function detecting the exception

  msg the message string identifying the exception
- **MadException (const MadException&)**
- **virtual ~MadException ()**

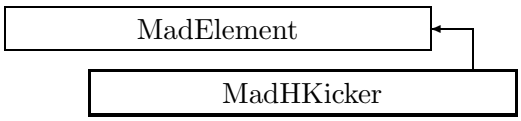


Figure 2.85: Inheritance for class MadHKicker

2.110 class MadHKicker

The HKICKER element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadHKicker.hh

Synopsis (including inherited members)

Public members

- **MadHKicker ()**
  The attributes of class MadHKicker. Exemplar constructor. (see Section 2.110.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadHKicker* clone (const string&)**
  Make clone. (see Section 2.110.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from DOOM data base. (see Section 2.110.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to DOOM data base. (see Section 2.110.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.110.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC corrector. (see Section 2.110.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadHKicker ()**
  (see Section 2.110.1)

2.110.1 Detailed descriptions

Public members

- **MadHKicker ()**
  The attributes of class MadHKicker. Exemplar constructor.
- **virtual MadHKicker* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read element from DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write element to DOOM data base.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC corrector.
- **virtual ~MadHKicker ()**

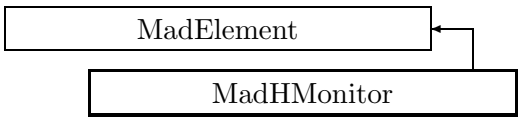


Figure 2.86: Inheritance for class MadHMonitor

2.111 class MadHMonitor

The HMONITOR element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadHMonitor.hh

Synopsis (including inherited members)

Public members

- **MadHMonitor ()**
  Exemplar constructor. (see Section 2.111.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadHMonitor* clone (const string&)**
  Make clone. (see Section 2.111.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC monitor. (see Section 2.111.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadHMonitor ()**
  (see Section 2.111.1)

2.111.1 Detailed descriptions

Public members

- **MadHMonitor ()**
  Exemplar constructor.
- **virtual MadHMonitor* clone (const string&)**
  Make clone.
- **virtual void update ()**
  Update the embedded CLASSIC monitor.
- **virtual ~MadHMonitor ()**

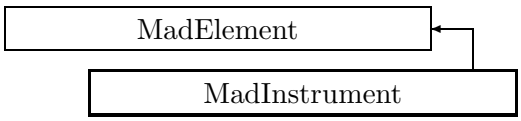


Figure 2.87: Inheritance for class MadInstrument

2.112 class MadInstrument

The INSTRUMENT element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadInstrument.hh

Synopsis (including inherited members)

Public members

- **MadInstrument ()**
  Exemplar constructor. (see Section 2.112.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadInstrument* clone (const string&)**
  Make clone. (see Section 2.112.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC drift. (see Section 2.112.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadInstrument ()**
  (see Section 2.112.1)

2.112.1 Detailed descriptions

Public members

- **MadInstrument ()**
  Exemplar constructor.
- **virtual MadInstrument* clone (const string&)**
  Make clone.
- **virtual void update ()**
  Update the embedded CLASSIC drift.
- **virtual ~MadInstrument ()**

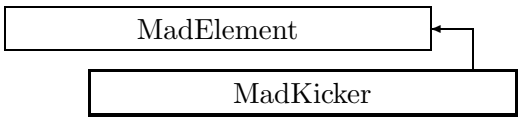


Figure 2.88: Inheritance for class MadKicker

2.113 class MadKicker

The KICKER element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadKicker.hh

Synopsis (including inherited members)

Public members

- **MadKicker ()**
  The attributes of class MadKicker. Exemplar constructor. (see Section 2.113.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadKicker* clone (const string&)**
  Make clone. (see Section 2.113.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  (see Section 2.113.1)
- **virtual void doomPut (DoomWriter&)const**
  (see Section 2.113.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.113.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC corrector. (see Section 2.113.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadKicker ()**
  (see Section 2.113.1)

2.113.1 Detailed descriptions

Public members

- **MadKicker ()**
  The attributes of class MadKicker. Exemplar constructor.
- **virtual MadKicker* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
- **virtual void doomPut (DoomWriter&)const**
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC corrector.
- **virtual ~MadKicker ()**

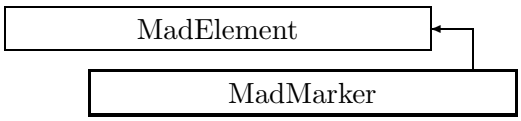


Figure 2.89: Inheritance for class MadMarker

2.114 class MadMarker

The MARKER element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadMarker.hh

Synopsis (including inherited members)

Public members

- **MadMarker ()**
  Exemplar constructor. (see Section 2.114.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadMarker* clone (const string&)**
  Make clone. (see Section 2.114.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the element. (see Section 2.114.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC marker. (see Section 2.114.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadMarker ()**
  (see Section 2.114.1)

2.114.1 Detailed descriptions

Public members

- **MadMarker ()**
  Exemplar constructor.
- **virtual MadMarker* clone (const string&)**
  Make clone.
- **virtual void print (std::ostream&)const**
  Print the element.

  Handle printing in MAD-8 format.
- **virtual void update ()**
  Update the embedded CLASSIC marker.
- **virtual ~MadMarker ()**

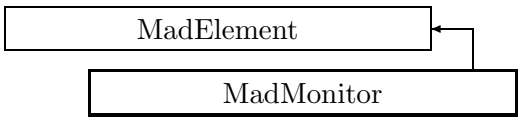


Figure 2.90: Inheritance for class MadMonitor

2.115 class MadMonitor

The MONITOR element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadMonitor.hh

Synopsis (including inherited members)

Public members

- **MadMonitor ()**
  Exemplar constructor. (see Section 2.115.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadMonitor* clone (const string&)**
  Make clone. (see Section 2.115.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC monitor. (see Section 2.115.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadMonitor ()**
  (see Section 2.115.1)

2.115.1 Detailed descriptions

Public members

- **MadMonitor ()**
  Exemplar constructor.
- **virtual MadMonitor* clone (const string&)**
  Make clone.
- **virtual void update ()**
  Update the embedded CLASSIC monitor.
- **virtual ~MadMonitor ()**

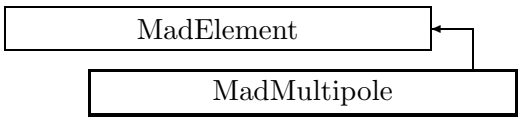


Figure 2.91: Inheritance for class MadMultipole

2.116 class MadMultipole

The MULTIPOLE element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadMultipole.hh

Synopsis (including inherited members)

Public members

- **MadMultipole ()**
  The attributes of class MadMultipole. Exemplar constructor. (see Section 2.116.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadMultipole* clone (const string&)**
  Make clone. (see Section 2.116.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.116.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.116.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.116.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.116.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC multipole. (see Section 2.116.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadMultipole ()**
  (see Section 2.116.1)

2.116.1 Detailed descriptions

Public members

- **MadMultipole ()**
  The attributes of class MadMultipole. Exemplar constructor.
- **virtual MadMultipole* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void print (std::ostream&)const**
  Print the object.

  Handle printing in MAD-8 format.
- **virtual void update ()**
  Update the embedded CLASSIC multipole.
- **virtual ~MadMultipole ()**

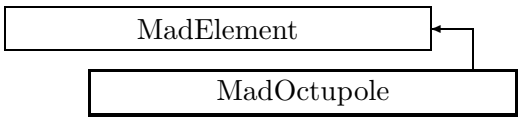


Figure 2.92: Inheritance for class MadOctupole

2.117 class MadOctupole

The OCTUPOLE element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadOctupole.hh

Synopsis (including inherited members)

Public members

- **MadOctupole ()**
  The attributes of class MadOctupole. Exemplar constructor. (see Section 2.117.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadOctupole* clone (const string&)**
  Make clone. (see Section 2.117.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.117.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.117.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the element. (see Section 2.117.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC multipole. (see Section 2.117.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadOctupole ()**
  (see Section 2.117.1)

2.117.1 Detailed descriptions

Public members

- **MadOctupole ()**
  The attributes of class MadOctupole. Exemplar constructor.
- **virtual MadOctupole* clone (const string&)**
  Make clone.
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void print (std::ostream&)const**
  Print the element.

  Handle printing in MAD-8 format.
- **virtual void update ()**
  Update the embedded CLASSIC multipole.
- **virtual ~MadOctupole ()**

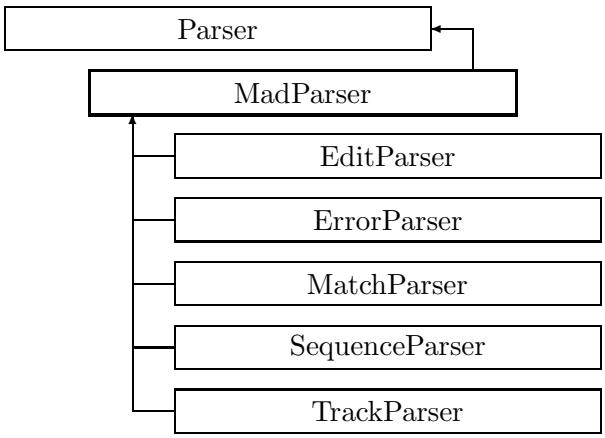


Figure 2.93: Inheritance for class MadParser

2.118 class MadParser

The default parser for MAD-9.
The parser reads a command name and keyword and looks up the keyword in the directory. If it finds an object, it makes a clone with the name read and calls the parser for the cloned object. If that parser succeeds, it calls the clone's execute() function. As required, it first updates the data structure to ensure that everything is up-to-date. Optionally, command execution is also traced.

Type:	Instantiable
Superclasses:	public Parser
Include file:	./MadParser/MadParser.hh

Synopsis (including inherited members)

Public members

- **MadParser ()**
  (see Section 2.118.1)
- **virtual void parse (Statement&)const**
  Parse and execute current statement. (see Section 2.118.1)
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream. (see Section 2.118.1)
- **static Token readToken ()**
  Return next input token. (see Section 2.118.1)
- **virtual void run ()const**
  Read current stream. (see Section 2.118.1)
- **virtual void run (TokenStream*)const**
  Read given stream. (see Section 2.118.1)
- **void stop ()const**
  Set stop flag. (see Section 2.118.1)

- **virtual ~MadParser ()**
  (see Section 2.118.1)

Protected members

- **void execute (Object*,const string&)const**
  Execute or check the current command. (see Section 2.118.1)
- **virtual Object* find (const string&)const**
  Find object by name in the main directory. (see Section 2.118.1)
- **virtual void parseAction (Statement&)const**
  Parse executable command. (see Section 2.118.1)
- **virtual void parseAssign (Statement&)const**
  Parse assignment statement. (see Section 2.118.1)
- **virtual void parseDefine (Statement&)const**
  Parse definition. (see Section 2.118.1)
- **virtual void parseEnd (Statement&)const**
  Check for end of statement. (see Section 2.118.1)
- **virtual void parseMacro (const string&,Statement&)const**
  Parse macro definition or call. (see Section 2.118.1)
- **virtual void printHelp (const string&)const**
  Print help on named command. (see Section 2.118.1)

2.118.1 Detailed descriptions

Public members

- **MadParser ()**

- **virtual void parse (Statement&)const**
  Parse and execute current statement.
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream.
- **static Token readToken ()**
  Return next input token.
- **virtual void run ()const**
  Read current stream.

  Read, parse, and execute statements one at a time.
- **virtual void run (TokenStream*)const**
  Read given stream.

  Switch to given stream, then read, parse, and execute statements one at a time. Used for CALL statements and macros.

- **void stop ()const**
  Set stop flag.

  Causes `run()` to return when the next statement should be read.

- **virtual ~MadParser ()**

Protected members

- **void execute (Object*,const string&)const**
  Execute or check the current command.
- **virtual Object* find (const string&)const**
  Find object by name in the main directory.
- **virtual void parseAction (Statement&)const**
  Parse executable command.
- **virtual void parseAssign (Statement&)const**
  Parse assignment statement.
- **virtual void parseDefine (Statement&)const**
  Parse definition.
- **virtual void parseEnd (Statement&)const**
  Check for end of statement.
- **virtual void parseMacro (const string&,Statement&)const**
  Parse macro definition or call.
- **virtual void printHelp (const string&)const**
  Print help on named command.

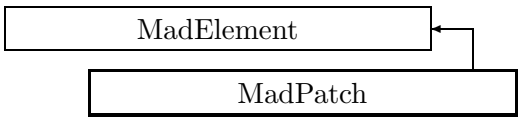


Figure 2.94: Inheritance for class MadPatch

2.119 class MadPatch

The PATCH element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadPatch.hh

Synopsis (including inherited members)

Public members

- **MadPatch ()**
  The attributes of class MadPatch. Exemplar constructor. (see Section 2.119.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadPatch* clone (const string&)**
  Make clone. (see Section 2.119.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isPatch ()const**
  Test for patch. (see Section 2.119.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)

- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC patch. (see Section 2.119.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadPatch ()**
  (see Section 2.119.1)

2.119.1 Detailed descriptions

Public members

- **MadPatch ()**
  The attributes of class MadPatch. Exemplar constructor.
- **virtual MadPatch* clone (const string&)**
  Make clone.
- **virtual bool isPatch ()const**
  Test for patch.

  Return true.
- **virtual void update ()**
  Update the embedded CLASSIC patch.
- **virtual ~MadPatch ()**

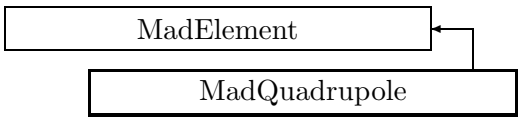


Figure 2.95: Inheritance for class MadQuadrupole

2.120 class MadQuadrupole

The QUADRUPOLE element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadQuadrupole.hh

Synopsis (including inherited members)

Public members

- **MadQuadrupole ()**
  The attributes of class MadQuadrupole. Exemplar constructor. (see Section 2.120.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadQuadrupole* clone (const string&)**
  Make clone. (see Section 2.120.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.120.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.120.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the quadrupole. (see Section 2.120.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC multipole. (see Section 2.120.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadQuadrupole ()**
  (see Section 2.120.1)

2.120.1 Detailed descriptions

Public members

- **MadQuadrupole ()**
  The attributes of class MadQuadrupole. Exemplar constructor.
- **virtual MadQuadrupole* clone (const string&)**
  Make clone.
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void print (std::ostream&)const**
  Print the quadrupole.

  Handle printing in MAD-8 format.
- **virtual void update ()**
  Update the embedded CLASSIC multipole.
- **virtual ~MadQuadrupole ()**

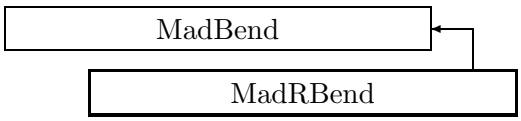


Figure 2.96: Inheritance for class MadRBend

2.121 class MadRBend

The RBEND element.

Type:	Instantiable
Superclasses:	public MadBend
Include file:	./Elements/MadRBend.hh

Synopsis (including inherited members)

Public members

- **MadRBend ()**
  Exemplar constructor. (see Section 2.121.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadRBend* clone (const string&)**
  Make clone. (see Section 2.121.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write bend magnet to DOOM data base. (see Section 2.103.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.121.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the bend magnet. (see Section 2.103.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC bend. (see Section 2.121.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadRBend ()**
  (see Section 2.121.1)

2.121.1 Detailed descriptions

Public members

- **MadRBend ()**
  Exemplar constructor.
- **virtual MadRBend* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC bend.
- **virtual ~MadRBend ()**

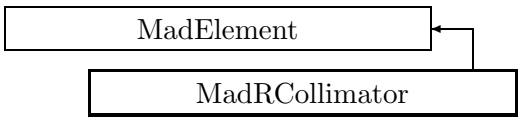


Figure 2.97: Inheritance for class MadRCollimator

2.122 class MadRCollimator

The RCOLLIMATOR element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadRCollimator.hh

Synopsis (including inherited members)

Public members

- **MadRCollimator ()**
  The attributes of class MadRCollimator. Exemplar constructor. (see Section 2.122.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadRCollimator* clone (const string&)**
  Make clone. (see Section 2.122.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.122.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC collimator. (see Section 2.122.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadRCollimator ()**
  (see Section 2.122.1)

2.122.1 Detailed descriptions

Public members

- **MadRCollimator ()**
  The attributes of class MadRCollimator. Exemplar constructor.
- **virtual MadRCollimator* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC collimator.
- **virtual ~MadRCollimator ()**

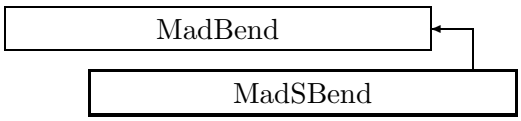


Figure 2.98: Inheritance for class MadSBend

2.123 class MadSBend

The SBEND element.

Type:	Instantiable
Superclasses:	public MadBend
Include file:	./Elements/MadSBend.hh

Synopsis (including inherited members)

Public members

- **MadSBend ()**
  Exemplar constructor. (see Section 2.123.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadSBend* clone (const string&)**
  Make clone. (see Section 2.123.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write bend magnet to DOOM data base. (see Section 2.103.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.123.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the bend magnet. (see Section 2.103.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC bend. (see Section 2.123.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadSBend ()**
  (see Section 2.123.1)

2.123.1 Detailed descriptions

Public members

- **MadSBend ()**
  Exemplar constructor.
- **virtual MadSBend* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC bend.
- **virtual ~MadSBend ()**

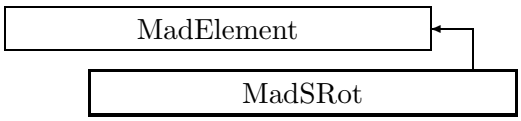


Figure 2.99: Inheritance for class MadSRot

2.124 class MadSRot

The SROT element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadSRot.hh

Synopsis (including inherited members)

Public members

- **MadSRot ()**
  The attributes of class MadSRot. Exemplar constructor. (see Section 2.124.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadSRot* clone (const string&)**
  Make clone. (see Section 2.124.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.124.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC patch. (see Section 2.124.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadSRot ()**
  (see Section 2.124.1)

2.124.1 Detailed descriptions

Public members

- **MadSRot ()**
  The attributes of class MadSRot. Exemplar constructor.
- **virtual MadSRot* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC patch.
- **virtual ~MadSRot ()**

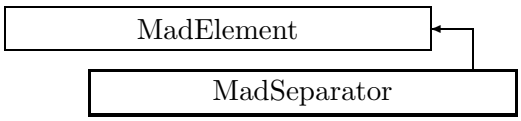


Figure 2.100: Inheritance for class MadSeparator

2.125 class MadSeparator

The ELSEPARATOR element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadSeparator.hh

Synopsis (including inherited members)

Public members

- **MadSeparator ()**
  The attributes of class MadSeparator. Exemplar constructor. (see Section 2.125.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadSeparator* clone (const string&)**
  Make clone. (see Section 2.125.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.125.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC separator. (see Section 2.125.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadSeparator ()**
  (see Section 2.125.1)

2.125.1 Detailed descriptions

Public members

- **MadSeparator ()**
  The attributes of class MadSeparator. Exemplar constructor.
- **virtual MadSeparator* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC separator.
- **virtual ~MadSeparator ()**

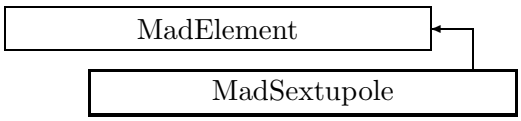


Figure 2.101: Inheritance for class MadSextupole

2.126 class MadSextupole

The SEXTUPOLE element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadSextupole.hh

Synopsis (including inherited members)

Public members

- **MadSextupole ()**
  The attributes of class MadSextupole. Exemplar constructor. (see Section 2.126.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadSextupole* clone (const string&)**
  Make clone. (see Section 2.126.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write the element to the DOOM data base. (see Section 2.126.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.126.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the sextupole. (see Section 2.126.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC multipole. (see Section 2.126.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadSextupole ()**
  (see Section 2.126.1)

2.126.1 Detailed descriptions

Public members

- **MadSextupole ()**
  The attributes of class MadSextupole. Exemplar constructor.
- **virtual MadSextupole* clone (const string&)**
  Make clone.
- **virtual void doomPut (DoomWriter&)const**
  Write the element to the DOOM data base.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void print (std::ostream&)const**
  Print the sextupole.

  Handle printing in MAD-8 format.
- **virtual void update ()**
  Update the embedded CLASSIC multipole.
- **virtual ~MadSextupole ()**

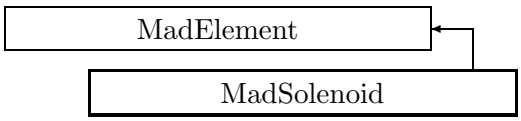


Figure 2.102: Inheritance for class MadSolenoid

2.127 class MadSolenoid

The SOLENOID element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadSolenoid.hh

Synopsis (including inherited members)

Public members

- **MadSolenoid ()**
  The attributes of class MadSolenoid. Exemplar constructor. (see Section 2.127.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadSolenoid* clone (const string&)**
  Make clone. (see Section 2.127.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.127.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC solenoid. (see Section 2.127.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadSolenoid ()**
  (see Section 2.127.1)

2.127.1 Detailed descriptions

Public members

- **MadSolenoid ()**
  The attributes of class MadSolenoid. Exemplar constructor.
- **virtual MadSolenoid* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC solenoid.
- **virtual ~MadSolenoid ()**

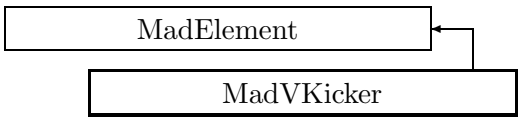


Figure 2.103: Inheritance for class MadVKicker

2.128 class MadVKicker

The VKICKER element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadVKicker.hh

Synopsis (including inherited members)

Public members

- **MadVKicker ()**
  The attributes of class MadVKicker. Exemplar constructor. (see Section 2.128.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadVKicker* clone (const string&)**
  Make clone. (see Section 2.128.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.128.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.128.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.128.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC corrector. (see Section 2.128.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadVKicker ()**
  (see Section 2.128.1)

2.128.1 Detailed descriptions

Public members

- **MadVKicker ()**
  The attributes of class MadVKicker. Exemplar constructor.
- **virtual MadVKicker* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC corrector.
- **virtual ~MadVKicker ()**

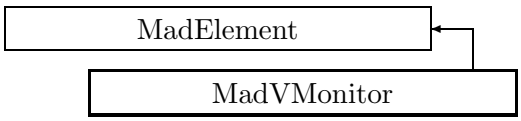


Figure 2.104: Inheritance for class MadVMonitor

2.129 class MadVMonitor

The VMONITOR element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadVMonitor.hh

Synopsis (including inherited members)

Public members

- **MadVMonitor ()**
  Exemplar constructor. (see Section 2.129.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadVMonitor* clone (const string&)**
  Make clone. (see Section 2.129.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.108.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC monitor. (see Section 2.129.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadVMonitor ()**
  (see Section 2.129.1)

2.129.1 Detailed descriptions

Public members

- **MadVMonitor ()**
  Exemplar constructor.
- **virtual MadVMonitor* clone (const string&)**
  Make clone.
- **virtual void update ()**
  Update the embedded CLASSIC monitor.
- **virtual ~MadVMonitor ()**

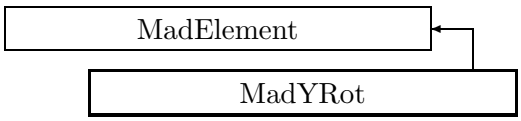


Figure 2.105: Inheritance for class MadYRot

2.130 class MadYRot

The YROT element.

Type:	Instantiable
Superclasses:	public MadElement
Include file:	./Elements/MadYRot.hh

Synopsis (including inherited members)

Public members

- **MadYRot ()**
  The attributes of class MadYRot. Exemplar constructor. (see Section 2.130.1)
- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **enum ValueFlag**
  The common attributes for all elements. Switch for value desired on ATTLIST command. (see Section 2.108.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MadYRot* clone (const string&)**
  Make clone. (see Section 2.130.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read element from the DOOM data base. (see Section 2.108.1)
- **virtual void doomPut (DoomWriter&)const**
  Write element to the DOOM data base. (see Section 2.108.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes. (see Section 2.130.1)

- **static Element* find (const string&)**
  Find named Element. (see Section 2.64.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **static AttCell* findRegisteredAttribute (const string&)**
  Find a registered attribute. (see Section 2.108.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.64.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return arc length from origin to entrance (negative !). (see Section 2.64.1)
- **virtual double getExit (ReferenceType)const**
  Return arc length from origin to exit (positive !). (see Section 2.64.1)
- **virtual double getLength ()const**
  Return element length. (see Section 2.108.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **const string getTypeName ()const**
  Return the element's type name. (see Section 2.108.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)

- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the element. (see Section 2.108.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.108.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **static void setRegisteredAttribute (const string&,double)**
  Store a registered real attribute. (see Section 2.108.1)
- **static void setRegisteredAttribute (const string&,const string&)**
  Store a registered string attribute. (see Section 2.108.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC patch. (see Section 2.130.1)
- **virtual void updateUnknown (ElementBase*)**
  Transmit the "unknown" (not known to MAD) attributes to CLASSIC. (see Section 2.108.1)
- **virtual ~MadYRot ()**
  (see Section 2.130.1)

2.130.1 Detailed descriptions

Public members

- **MadYRot ()**
  The attributes of class MadYRot. Exemplar constructor.
- **virtual MadYRot* clone (const string&)**
  Make clone.
- **virtual void fillRegisteredAttributes (const ElementBase&,ValueFlag)**
  Fill in all registered attributes.
- **virtual void update ()**
  Update the embedded CLASSIC patch.
- **virtual ~MadYRot ()**

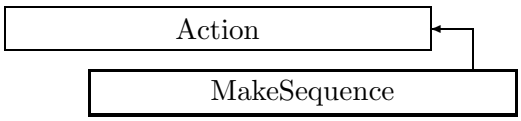


Figure 2.106: Inheritance for class MakeSequence

2.131 class MakeSequence

The MAKESEQ command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./PhysicsActions/MakeSequence.hh

Synopsis (including inherited members)

Public members

- **MakeSequence ()**
  Exemplar constructor. (see Section 2.131.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MakeSequence* clone (const string&)**
  Make clone. (see Section 2.131.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.131.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~MakeSequence ()**
  (see Section 2.131.1)

2.131.1 Detailed descriptions

Public members

- **MakeSequence ()**
  Exemplar constructor.
- **virtual MakeSequence* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~MakeSequence ()**

2.132 class Match

Matching block.
This class encapsulates all data for matching which do not depend on the matching method. It acts as a communication area between the various matching commands.

Type:	Instantiable
Include file:	./Match/Match.hh

Synopsis (including inherited members)

Public members

- **typedef std::list<AbstractFun*> FunList**
  (see Section 2.132.1)
- **Match ()**
  Constructor. (see Section 2.132.1)
- **typedef std::list<AbstractVar*> VarList**
  (see Section 2.132.1)
- **void addFunction (AbstractFun*)**
  Add a set of matching function(s). (see Section 2.132.1)
- **void addVariable (AbstractVar*)**
  Add a matching variable. (see Section 2.132.1)
- **static Match* block**
  The block of match data. (see Section 2.132.1)
- **int countFunctions ()const**
  Return total number of functions. (see Section 2.132.1)
- **int countVariables ()const**
  Get total number of variables. (see Section 2.132.1)
- **void deleteVariable (const string&)**
  Delete a matching variable. (see Section 2.132.1)
- **bool evaluate (const Vector<double>&,Vector<double>&)**
  Evaluate the matching functions. (see Section 2.132.1)
- **AbstractVar* findVariable (const string&)**
  Find a matching variable. (see Section 2.132.1)
- **int getCallCount ()const**
  Return count of function evaluations. (see Section 2.132.1)
- **void getFunctions (Vector<double>&)const**
  Get cached values of matching functions. (see Section 2.132.1)
- **int getPrintLevel ()const**
  Get the flag for printing. (see Section 2.132.1)

- **void getVariables (Vector<double>&)const**
  Get values of matching variables. (see Section 2.132.1)
- **MatchParser parser**
  The parser used during for matching. (see Section 2.132.1)
- **void print (const char*,MatchState)**
  Print the results of minimisation. (see Section 2.132.1)
- **void setPrintLevel (int)**
  Set the flag for printing. (see Section 2.132.1)
- **void setVariables (const Vector<double>&)**
  Set values of matching variables. (see Section 2.132.1)
- **~Match ()**
  (see Section 2.132.1)

2.132.1 Detailed descriptions

Public members

- **typedef std::list<AbstractFun*> FunList**
- **Match ()**
  Constructor.
- **typedef std::list<AbstractVar*> VarList**
- **void addFunction (AbstractFun*)**
  Add a set of matching function(s).
- **void addVariable (AbstractVar*)**
  Add a matching variable.
- **static Match* block**
  The block of match data.
- **int countFunctions ()const**
  Return total number of functions.
- **int countVariables ()const**
  Get total number of variables.
- **void deleteVariable (const string&)**
  Delete a matching variable.

  Identified by name.
- **bool evaluate (const Vector<double>&,Vector<double>&)**
  Evaluate the matching functions.

  Set the matching variables to **x**, cache the function values, and return them in **f**. The boolean return value indicates success (true) or failure (false).

- **AbstractVar* findVariable (const string&)**
  Find a matching variable.

  Identified by name.

- **int getCallCount ()const**
  Return count of function evaluations.

- **void getFunctions (Vector<double>&)const**
  Get cached values of matching functions.

- **int getPrintLevel ()const**
  Get the flag for printing.

- **void getVariables (Vector<double>&)const**
  Get values of matching variables.

- **MatchParser parser**
  The parser used during for matching.

- **void print (const char*,MatchState)**
  Print the results of minimisation.

- **void setPrintLevel (int)**
  Set the flag for printing.

- **void setVariables (const Vector<double>&)**
  Set values of matching variables.

- **~Match ()**

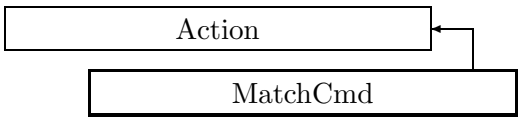


Figure 2.107: Inheritance for class MatchCmd

2.133 class MatchCmd

The MATCH command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/MatchCmd.hh

Synopsis (including inherited members)

Public members

- **MatchCmd ()**
  Exemplar constructor. (see Section 2.133.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MatchCmd* clone (const string&)**
  Make clone. (see Section 2.133.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.133.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~MatchCmd ()**
  (see Section 2.133.1)

2.133.1 Detailed descriptions

Public members

- **MatchCmd ()**
  Exemplar constructor.
- **virtual MatchCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~MatchCmd ()**

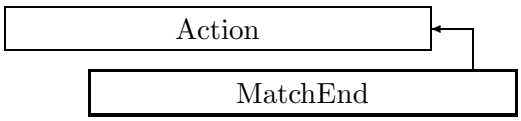


Figure 2.108: Inheritance for class MatchEnd

2.134 class MatchEnd

The ENDMATCH command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/MatchEnd.hh

Synopsis (including inherited members)

Public members

- **MatchEnd ()**
  Exemplar constructor. (see Section 2.134.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MatchEnd* clone (const string&)**
  Make clone. (see Section 2.134.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.134.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~MatchEnd ()**
  (see Section 2.134.1)

2.134.1 Detailed descriptions

Public members

- **MatchEnd ()**
  Exemplar constructor.
- **virtual MatchEnd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~MatchEnd ()**

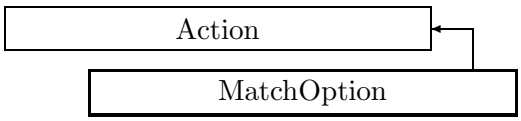


Figure 2.109: Inheritance for class MatchOption

2.135 class MatchOption

The OPTION command for matching.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/MatchOption.hh

Synopsis (including inherited members)

Public members

- **MatchOption ()**
  Exemplar constructor. (see Section 2.135.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MatchOption* clone (const string&)**
  Make clone. (see Section 2.135.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.135.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~MatchOption ()**
  (see Section 2.135.1)

2.135.1 Detailed descriptions

Public members

- **MatchOption ()**
  Exemplar constructor.
- **virtual MatchOption* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~MatchOption ()**

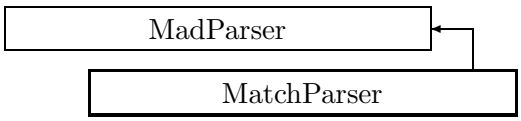


Figure 2.110: Inheritance for class MatchParser

2.136 class MatchParser

The parser used in the MAD match module.
As long as control remains in this class, MAD recognises only the commands allowed in matching mode.

Type:	Instantiable
Superclasses:	public MadParser
Include file:	./Match/MatchParser.hh

Synopsis (including inherited members)

Public members

- **MatchParser ()**
  (see Section 2.136.1)
- **virtual void parse (Statement&)const**
  Parse and execute current statement. (see Section 2.118.1)
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream. (see Section 2.118.1)
- **static Token readToken ()**
  Return next input token. (see Section 2.118.1)
- **virtual void run ()const**
  Read current stream. (see Section 2.118.1)
- **virtual void run (TokenStream*)const**
  Read given stream. (see Section 2.118.1)
- **void stop ()const**
  Set stop flag. (see Section 2.118.1)
- **virtual ~MatchParser ()**
  (see Section 2.136.1)

Protected members

- **void execute (Object*,const string&)const**
  Execute or check the current command. (see Section 2.118.1)
- **virtual Object* find (const string&)const**
  Find object by name in the match command directory. (see Section 2.136.1)

- **virtual void parseAction (Statement&)const**
  Parse executable command. (see Section 2.118.1)
- **virtual void parseAssign (Statement&)const**
  Parse assignment statement. (see Section 2.118.1)
- **virtual void parseDefine (Statement&)const**
  Parse definition. (see Section 2.118.1)
- **virtual void parseEnd (Statement&)const**
  Check for end of statement. (see Section 2.118.1)
- **virtual void parseMacro (const string&,Statement&)const**
  Parse macro definition or call. (see Section 2.118.1)
- **virtual void printHelp (const string&)const**
  Print help on named command. (see Section 2.118.1)

2.136.1 Detailed descriptions

Public members

- **MatchParser ()**

- **virtual ~MatchParser ()**

Protected members

- **virtual Object* find (const string&)const**
  Find object by name in the match command directory.

2.137 template class Matrix <class>

Type:	Instantiable

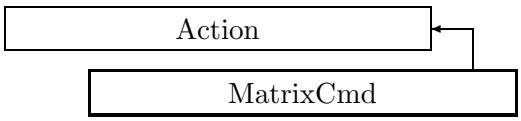


Figure 2.111: Inheritance for class MatrixCmd

2.138 class MatrixCmd

The MATRIX command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Tables/MatrixCmd.hh

Synopsis (including inherited members)

Public members

- **MatrixCmd ()**
  Exemplar constructor. (see Section 2.138.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual MatrixCmd* clone (const string&)**
  Make clone. (see Section 2.138.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.138.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~MatrixCmd ()**
  (see Section 2.138.1)

2.138.1 Detailed descriptions

Public members

- **MatrixCmd ()**
  Exemplar constructor.
- **virtual MatrixCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~MatrixCmd ()**

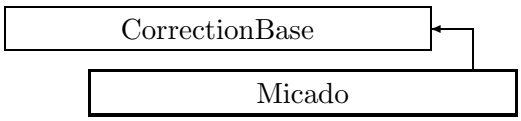


Figure 2.112: Inheritance for class Micado

2.139 class Micado

Class Micado MAD command MICADO.

Type:	Instantiable
Superclasses:	public CorrectionBase
Include file:	./Tables/Micado.hh

Synopsis (including inherited members)

Public members

- **Micado ()**
  Exemplar constructor. (see Section 2.139.1)
- **typedef TBeamline<Row> TLine**
  (see Section 2.38.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Micado* clone (const string&)**
  Make clone. (see Section 2.139.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Check validity of the table definition. (see Section 2.139.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)

- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Micado ()**
  (see Section 2.139.1)

2.139.1 Detailed descriptions

Public members

- **Micado ()**
  Exemplar constructor.
- **virtual Micado* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Check validity of the table definition.

- **virtual ~Micado ()**

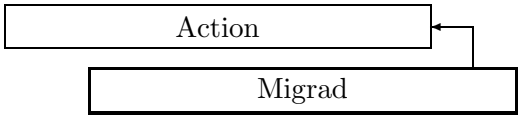


Figure 2.113: Inheritance for class Migrad

2.140 class Migrad

The MIGRAD command.
This class encapsulates a minimisation according to the MIGRAD method, Minimization by a gradient method due to Davidon/Fletcher/Powell (Computer Journal 13, 317, 1970). Also described in

J. F. Bonnans et al., Optimisation Numerique, pp. 44-46. Springer, Berlin, 1997

Algorithm rewritten following the MINUIT package.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/Migrad.hh

Synopsis (including inherited members)

Public members

- **Migrad ()**
  Exemplar constructor. (see Section 2.140.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Migrad* clone (const string&)**
  Make clone. (see Section 2.140.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.140.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)

- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Migrad ()**
  (see Section 2.140.1)

2.140.1 Detailed descriptions

Public members

- **Migrad ()**
  Exemplar constructor.
- **virtual Migrad* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.

- **virtual ~Migrad ()**

2.141 class Multipole

Type:	Instantiable

2.142 class MultipoleWrapper

Type:	Instantiable

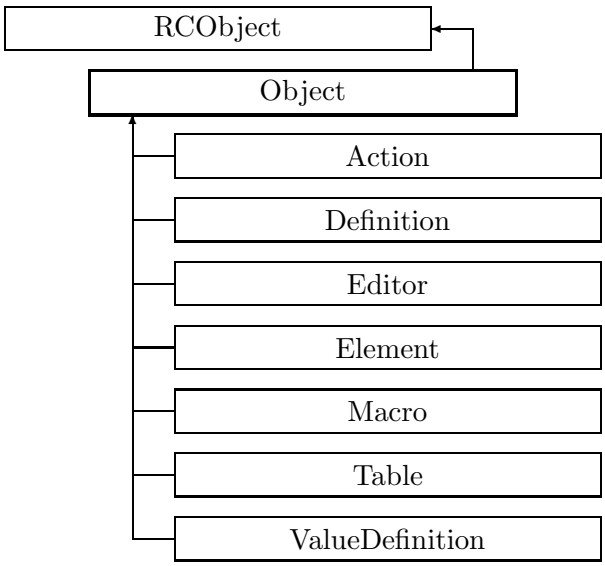


Figure 2.114: Inheritance for class Object

2.143 class Object

The base class for all MAD objects.
Any MAD object which may be generated by parsing a command or definition. All known Objects are referred to via **Pointer<Object>** to make memory management easy.
Objects are linked by name to the MAD directory, which resides in the global object **MadData**. Each Object has a pointer pointing to its parent object. This may be an exemplar object, or a previously generated command or definition object. This mechanism allows to find easily whether an Object belongs to a given class (see isTreeMember()). The class Object has as base class the abstract class RCObject.

Type:	abstract
Superclasses:	public RCObject
Include file:	./AbstractObjects/Object.hh

Synopsis (including inherited members)

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)

- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.143.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.143.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.143.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Object ()**
  (see Section 2.143.1)

Protected members

- **Object (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.143.1)
- **Object (const string&,Object*)**
  Constructor for clones. (see Section 2.143.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.143.1 Detailed descriptions

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed.

  By default redefinition of an existing Object is not allowed.
- **void clear ()**
  Clear the occurrence counter.
- **virtual Object* clone (const string&)**
  Return a clone.

  Call the **clone** constructor to generate a copy of **this** with a new name **name**.
- **void copyAttributes (const Object&)**
  Copy attributes from another object.
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base.
- **virtual void execute ()**
  Execute the command.

  For definitions check validity, for actions execute the action.
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name.

  Find the Object's attribute identified by **name** (version for non-constant object).
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name.

  Find the Object's attribute identified by **name** (version for constant object).

- **const Object* getBaseObject ()const**
  Return the object's base type object.

  Return the top-most ancestor of this object, i. e. the built-in object from which **this** is ultimately derived.

- **virtual const string getCategory ()const**
  Return the object category as a string.

  Is overridden by the (still abstract) sub-classes of Object.

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime().

- **const string& getMadName ()const**
  Return object name.

- **Object* getParent ()const**
  Return parent pointer.

- **int increment ()**
  Increment and return the occurrence counter.

- **bool isBuiltin ()const**
  True, if this is a built-in object.

- **bool isDirty ()const**
  True, if the modified flag is set.

- **bool isFlagged ()const**
  True, if this is flagged by setFlag(true).

- **virtual bool isShared ()const**
  Shared flag.

  If true, this object cannot be cloned to create new elements. If it is an element or a line, all references are to the same object.

- **bool isTreeMember (const Object*)const**
  Test for tree membership.

  Return true, if **this** has been directly or indirectly derived from **subTree**.

- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh).

- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function.

  Construct an instance object from an "archetype" object. The default version throws MadException.

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function.

  Construct an "archetype" object for a MACRO commands. The default version throws MadException.

- **int occurrenceCount ()**
  Return the occurrence counter.

- **virtual void parse (Statement&)**
  Parse the object.

  Parse the statement and fill in the Object's attributes.

- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands.

  This parser allows to use unnamed attributes for command which have only one attribute.

- **virtual void print (std::ostream&)const**
  Print the object.

  Print a MAD-readable image of **this** on the given output stream.

- **virtual void printHelp (std::ostream&)const**
  Print help.

  Print help information for **this** on the given output stream.

- **void registerReference (AttributeBase*)**
  Register a reference to this object.

  Place **a** in the list of references. Whenever **this** is erased or modified, **a** will be notified of the change.

- **virtual void replace (Object*,Object*)**
  Replace references.

  Called for all defined objects, when an object **oldObject** is replaced by a new definition **newObject**. This default version does nothing. Some derived classes may react to the redefinition by invalidating themselves or by replacing reference inside their data.

- **void setDirty (bool)**
  Set/reset the modified flag.

  Set the flag when an object is created and modified, reset it when the object is saved in the DOOM data base.

- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base.

- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for

  flagged objects only.

- **void setMadName (const string&)**
  Set object name.

- **void setParent (Object*)**
  Set parent object.

  This method should normally be called only from Directory.

- **virtual void setShared (bool)**
  Set/reset shared flag.

  If true, this object cannot be cloned to create new elements. If it is an element or a line, all references are to the same object.

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute().

- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object.

  Remove **a** from the list of references. The object **a** will no longer be notified of any change to **this**.

- **virtual void update ()**
  Update this object.

  Called for all defined objects, before an action command is executed. This default version does nothing. Derived classes may use this call to update their internal state. The beam line elements can update their CLASSIC counterpart.

- **virtual ~Object ()**

Protected members

- **Object (int,const char*,const char*)**
  Constructor for exemplars.

  The exemplar object will have the name **name**, and the help text **help**; it initially reserves **size** attributes.

- **Object (const string&,Object*)**
  Constructor for clones.

  The clone object will have the name **name**; it inherits its attributes (shared) from the object ***parent**.

- **bool builtin**
  Built-in flag.

  True, if the object is built-in to MAD-9, i. e. if it is an exemplar or a predefined definition.

- **bool flagged**
  Object flag.

  This flag can be freely set and reset during the execution of a command.

- **bool modified**
  Dirty flag.

  True when **this** is new or modified and should be saved in the DOOM data base.

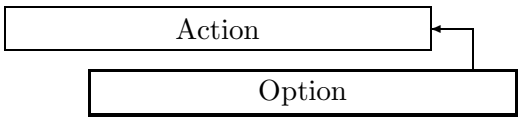


Figure 2.115: Inheritance for class Option

2.144 class Option

The OPTION command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Option.hh

Synopsis (including inherited members)

Public members

- **Option ()**
  Exemplar constructor. (see Section 2.144.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Option* clone (const string&)**
  Make clone. (see Section 2.144.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.144.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Option ()**
  (see Section 2.144.1)

2.144.1 Detailed descriptions

Public members

- **Option ()**
  Exemplar constructor.
- **virtual Option* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Option ()**

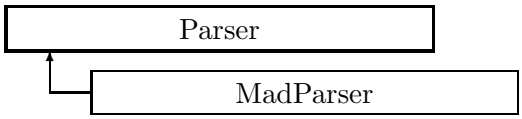


Figure 2.116: Inheritance for class Parser

2.145 class Parser

Type:	Instantiable

2.146 class PartBunch

Type:	Instantiable

2.147 class PartData

Type:	Instantiable

2.148 class Particle

Type:	Instantiable

2.149 class Patch

Type:	Instantiable

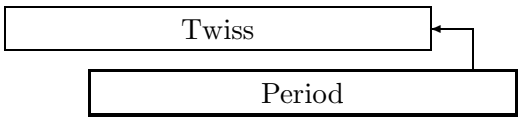


Figure 2.117: Inheritance for class Period

2.150 class Period

The TWISS command.

Type:	Instantiable
Superclasses:	public Twiss
Include file:	./Tables/Period.hh

Synopsis (including inherited members)

Public members

- **struct Cell**
  Descriptor for printing a table cell. (see Section 2.187.1)
- **typedef std::vector<Cell> CellArray**
  An array of cell descriptors. (see Section 2.187.1)
- **Period ()**
  Exemplar constructor. (see Section 2.150.1)
- **class Row**
  Structure for a row of the Twiss table. (see Section 2.205.1)
- **typedef TBeamline<Row> TLine**
  (see Section 2.205.1)
- **TLine::const_iterator begin ()const**
  Access to first row. (see Section 2.205.1)
- **TLine::iterator begin ()**
  Access to first row. (see Section 2.205.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.187.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Period* clone (const string&)**
  Make clone. (see Section 2.150.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)

- **virtual void doomPut (DoomWriter&)const**
  Write the table to the DOOM data base. (see Section 2.205.1)
- **virtual void doomSummary (DoomWriter&)const**
  Fill in summary record. (see Section 2.150.1)
- **TLine::const_iterator end ()const**
  Access to last row. (see Section 2.205.1)
- **TLine::iterator end ()**
  Access to last row. (see Section 2.205.1)
- **virtual void execute ()**
  Check validity of the table definition. (see Section 2.205.1)
- **virtual void fill ()**
  Fill the buffer using the defined algorithm. (see Section 2.150.1)
- **static Table* find (const string&)**
  Find named Table. (see Section 2.187.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **double getALFi (const Row&,int,int)const**
  (see Section 2.205.1)
- **double getALFik (const Row&,int,int)const**
  Mais-Ripken alpha functions. (see Section 2.205.1)
- **double getBETi (const Row&,int,int)const**
  (see Section 2.205.1)
- **double getBETik (const Row&,int,int)const**
  Mais-Ripken beta functions. (see Section 2.205.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **double getCO (const Row&,int,int)const**
  Closed orbit. (see Section 2.205.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.187.1)
- **virtual double getCell (const PlaceRep&,const string&)**
  Return a selected value in a selected row. (see Section 2.205.1)
- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column **col** of this table, limited by **range**. (see Section 2.205.1)

- **FMatrix<double,6,6> getCurlyA ()const**
  Return initial curly A matrix. (see Section 2.205.1)
- **FMatrix<double,6,6> getCurlyA (const Row&)const**
  Curly A map for given row. (see Section 2.205.1)
- **const Row& getCurrent ()const**
  Return current table row in iteration. (see Section 2.205.1)
- **virtual CellArray getDefault ()const**
  Return the default print columns. (see Section 2.205.1)
- **double getDisp (const Row&,int,int)const**
  Dispersion. (see Section 2.205.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **double getET ()const**
  Return emittance for mode 3. (see Section 2.205.1)
- **double getEX ()const**
  Return emittance for mode 1. (see Section 2.205.1)
- **double getEY ()const**
  Return emittance for mode 2. (see Section 2.205.1)
- **double getEigen (const Row&,int,int)const**
  Eigenvectors. (see Section 2.205.1)
- **double getGAMik (const Row&,int,int)const**
  Mais-Ripken gamma functions. (see Section 2.205.1)
- **virtual double getLength ()**
  Return the length of the table. (see Section 2.205.1)
- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline. (see Section 2.205.1)
- **double getMUi (const Row&,int,int)const**
  Three modes, "naive" Twiss functions. (see Section 2.205.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **FMatrix<double,6,6> getMatrix (const Row&)const**
  Accumulated transfer map. (see Section 2.205.1)
- **double getMatrix (const Row&,int,int)const**
  Transfer matrix. (see Section 2.205.1)
- **FVector<double,6> getOrbit ()const**
  Return initial closed orbit. (see Section 2.205.1)

- **FVector<double,6> getOrbit (const Row&)const**
  Get orbit in given row. (see Section 2.205.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row, possible user-defined. (see Section 2.205.1)
- **double getS (const Row&,int,int)const**
  Arc length for given row. (see Section 2.205.1)
- **FMatrix<double,6,6> getSigma ()const**
  Initial envelope (Sigma) matrix. (see Section 2.205.1)
- **FMatrix<double,6,6> getSigma (const Row&)const**
  Envelope (Sigma) matrix for given row. (see Section 2.205.1)
- **double getSigma (const Row&,int,int)const**
  Sigma matrix. (see Section 2.205.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **virtual void invalidate ()**
  Mark this table as invalid, if it is dynamic. (see Section 2.187.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **virtual bool isDependent (const string&)const**
  Check dependency. (see Section 2.205.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return column expression. (see Section 2.205.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table. (see Section 2.150.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual bool matches (Table*)const**
  Check compatibility. (see Section 2.205.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print the table on an ASCII stream. (see Section 2.150.1)
- **void printTableBody (std::ostream&,const CellArray&)const**
  Print the body to this TWISS table. (see Section 2.205.1)
- **void printTableTitle (std::ostream&,const char*)const**
  Print standard information about the TWISS table. (see Section 2.205.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)

- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.187.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.187.1)
- **void tfsBody (std::ostream&,const CellArray&)const**
  Write TFS body for this table. (see Section 2.205.1)
- **void tfsSummary (std::ostream&)const**
  List TFS descriptors for table summary. (see Section 2.205.1)
- **void tfsTableDescriptors (std::ostream&)const**
  Write TFS descriptors existing for all tables. (see Section 2.187.1)
- **virtual void tfsTwissDescriptors (std::ostream&)const**
  Write TFS descriptors for a TWISS table. (see Section 2.150.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Period ()**
  (see Section 2.150.1)

2.150.1 Detailed descriptions

Public members

- **Period ()**
  Exemplar constructor.
- **virtual Period* clone (const string&)**
  Make clone.
- **virtual void doomSummary (DoomWriter&)const**
  Fill in summary record.
- **virtual void fill ()**
  Fill the buffer using the defined algorithm.
- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table.
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print the table on an ASCII stream.

- **virtual void tfsTwissDescriptors (std::ostream&)const**
  Write TFS descriptors for a TWISS table.

  Writes the line and beam info, as well as length and tunes.

- **virtual ~Period ()**

2.151 class PlaceRep

Representation of a place within a beam line or sequence.
This representation holds data to keep track wether the place was already passed when tracking through the line. The place may be one of the following:

"#S", the start of the beam line.

"#E", the end of the beam line.

"SELECTED", all selected positions (flag set in FlaggedElmPtr).

name[occurrence]::name[occurrence]..., denoting a specific place.

Type:	Instantiable
Include file:	./AbstractObjects/PlaceRep.hh

Synopsis (including inherited members)

Public members

- **typedef std::vector<std::pair<string,int> > Data**
  (see Section 2.151.1)
- **PlaceRep ()**
  Default constructor. (see Section 2.151.1)
- **PlaceRep (const string&)**
  Construct default place. (see Section 2.151.1)
- **PlaceRep (const PlaceRep&)**
  (see Section 2.151.1)
- **void append (const string&,int)**
  Add a name/occurrence pair. (see Section 2.151.1)
- **void enter (const FlaggedElmPtr&)const**
  Enter an element or line. (see Section 2.151.1)
- **void initialize ()**
  Initialise data for search. (see Section 2.151.1)
- **bool isActive ()const**
  Return status. (see Section 2.151.1)
- **bool isSelected ()const**
  Return select flag. (see Section 2.151.1)
- **void leave (const FlaggedElmPtr&)const**
  Leave an element or line. (see Section 2.151.1)
- **const PlaceRep& operator= (const PlaceRep&)**
  (see Section 2.151.1)

- **std::ostream& print (std::ostream&)const**
  Print in input format. (see Section 2.151.1)
- **~PlaceRep ()**
  (see Section 2.151.1)

2.151.1 Detailed descriptions

Public members

- **typedef std::vector<std::pair<string,int> > Data**

- **PlaceRep ()**
  Default constructor.

  Constructs undefined place.
- **PlaceRep (const string&)**
  Construct default place.

  Used for places like "#S", "#E", "SELECTED".
- **PlaceRep (const PlaceRep&)**

- **void append (const string&,int)**
  Add a name/occurrence pair.
- **void enter (const FlaggedElmPtr&)const**
  Enter an element or line.

  Sets the internal state to active, when the specified place is entered.
- **void initialize ()**
  Initialise data for search.

  Sets the internal state to the beginning of the line.
- **bool isActive ()const**
  Return status.

  Returns true, if we are at the specified place.
- **bool isSelected ()const**
  Return select flag.

  Returns true, if the place has the value "SELECTED".
- **void leave (const FlaggedElmPtr&)const**
  Leave an element or line.

  Sets the internal state to inactive, when the specified place is left.
- **const PlaceRep& operator= (const PlaceRep&)**

- **std::ostream& print (std::ostream&)const**
  Print in input format.
- **~PlaceRep ()**

2.152 class PlanarArcGeometry

Type:	Instantiable

2.153 class QRSolver

Least-square solution of systems of linear equations.
Given an m by n matrix A, an n by n diagonal matrix D, and an m-vector B, two problem can be solved:

Solve the the system $A * X = B$ in the least squares sense. The first step to solve this problem is:

```
QRSolver solver(A, pivot);
```

The second step is then

```
solver.solveR(X);
```

Solve the the two systems $A * X = B, D * X = 0$ in the least squares sense. The first step to solve this problem is

```
QRSolver solver(A, pivot);
```

The second step is

```
solver.solveS(D, X);
```

The second step can be repeated as many times as required for different diagonal matrices D.

In both cases, the method
`solver.getColNorm(C);`
can be called to return the original column norms of A.

Type:	Instantiable
Include file:	./Algebra/QRSolver.hh

Synopsis (including inherited members)

Public members

- **QRSolver (const Matrix<double>&,const Vector<double>&,bool)**
  Constructor. (see Section 2.153.1)
- **void getColNorm (Array1D<double>&)const**
  Return the original column norms of the matrix A. (see Section 2.153.1)
- **void solveR (Vector<double>&)const**
  Solution of $A * X = B$ in the least-squares sense. (see Section 2.153.1)
- **void solveRT (Vector<double>&)const**
  Pre-multiply the vector V by $R.transpose()^{-1}$. (see Section 2.153.1)
- **void solveS (const Array1D<double>&,double,Vector<double>&)**
  Solution of $A * X = B, D * X = 0$ in the least-squares sense. (see Section 2.153.1)

- **void solveST (Vector<double>&)const**
  Pre-multiply the vector V by $S.transpose()^{-1}$. (see Section 2.153.1)

- **~QRSolver ()**
  (see Section 2.153.1)

2.153.1 Detailed descriptions

Public members

- **QRSolver (const Matrix<double>&,const Vector<double>&,bool)**
  Constructor.

  Determine the QR-factorisation $A = Q * R$ of the matrix A and transform the right-hand side B. If **pivot** is true, then pivot search is done.

  This method uses Householder transformations with optional column pivoting to compute a QR-factorization of the m-by-n matrix A. It determines an orthogonal matrix Q, a permutation matrix P, and an upper trapezoidal matrix R with diagonal elements of nonincreasing magnitude, such that $A * P = Q * R$. The Householder transformation for column k is of the form $I - (1/U(k)) * U * U^T$, where U has zeros in the first $k-1$ positions. The form of this transformation and the method of pivoting first appeared in the corresponding LINPACK subroutine.

- **void getColNorm (Array1D<double>&)const**
  Return the original column norms of the matrix A.

- **void solveR (Vector<double>&)const**
  Solution of $A * X = B$ in the least-squares sense.

- **void solveRT (Vector<double>&)const**
  Pre-multiply the vector V by $R.transpose()^{-1}$.

- **void solveS (const Array1D<double>&,double,Vector<double>&)**
  Solution of $A * X = B, D * X = 0$ in the least-squares sense.

- **void solveST (Vector<double>&)const**
  Pre-multiply the vector V by $S.transpose()^{-1}$.

  Requires prior execution of solveS.

- **~QRSolver ()**

2.154 class RBend

Type:	Instantiable

2.155 class RBendWrapper

Type:	Instantiable

2.156 class RFCavity

Type:	Instantiable

2.157 class RangeRep

Representation of a range within a beam line or sequence.
This representation holds two places, defining the beginning and ending of the range. It keeps track wether we are before, within, or after the range. The value may also be "FULL", denoting the complete beam line.

Type:	Instantiable
Include file:	./AbstractObjects/RangeRep.hh

Synopsis (including inherited members)

Public members

- **RangeRep ()**
  Default constructor. (see Section 2.157.1)
- **RangeRep (PlaceRep&,PlaceRep&)**
  Constructor from two given places. (see Section 2.157.1)
- **RangeRep (const RangeRep&)**
  (see Section 2.157.1)
- **void enter (const FlaggedElmPtr&)const**
  Enter an element or line. (see Section 2.157.1)
- **void initialize ()**
  Initialise data for search. (see Section 2.157.1)
- **bool isActive ()const**
  Test for active range. (see Section 2.157.1)
- **void leave (const FlaggedElmPtr&)const**
  Leave an element or line. (see Section 2.157.1)
- **const RangeRep& operator= (const RangeRep&)**
  (see Section 2.157.1)
- **std::ostream& print (std::ostream&)const**
  Print in input format. (see Section 2.157.1)
- **~RangeRep ()**
  (see Section 2.157.1)

2.157.1 Detailed descriptions

Public members

- **RangeRep ()**
  Default constructor.

  Construct range for full line (value "FULL").
- **RangeRep (PlaceRep&,PlaceRep&)**
  Constructor from two given places.

- **RangeRep (const RangeRep&)**

- **void enter (const FlaggedElmPtr&)const**
  Enter an element or line.

  Sets the internal state to active, when we enter the specified range.

- **void initialize ()**
  Initialise data for search.

  Sets the internal state to the beginning of the beam line.

- **bool isActive ()const**
  Test for active range.

  Return true, if we are within the specified range.

- **void leave (const FlaggedElmPtr&)const**
  Leave an element or line.

  Sets the internal state to inactive, when we leave the specified range.

- **const RangeRep& operator= (const RangeRep&)**

- **std::ostream& print (std::ostream&)const**
  Print in input format.

- **˜RangeRep ()**

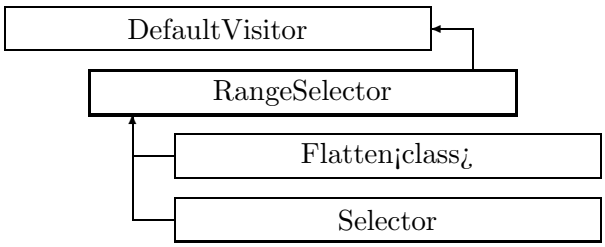


Figure 2.118: Inheritance for class RangeSelector

2.158 class RangeSelector

An abstract visitor which calls the pure virtual method `RangeSelector::handleXXX()` for each element or beamline in range.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Tables/RangeSelector.hh

Synopsis (including inherited members)

Public members

- **RangeSelector (const Beamline&,const RangeRep&)**
  Constructor. (see Section 2.158.1)
- **virtual void execute ()**
  Execute the algorithm. (see Section 2.158.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the visitor to an FlaggedElmPtr. (see Section 2.158.1)
- **virtual ~RangeSelector ()**
  (see Section 2.158.1)

Protected members

- **virtual void handleBeamline (const FlaggedElmPtr&)**
  The operation to be done for beamlines. (see Section 2.158.1)
- **virtual void handleElement (const FlaggedElmPtr&)**
  The operation to be done for elements. (see Section 2.158.1)
- **RangeRep itsRange**
  Working data for range. (see Section 2.158.1)

2.158.1 Detailed descriptions

Public members

- **RangeSelector (const Beamline&,const RangeRep&)**
  Constructor.

Attach visitor to a beamline, remember the range.

- **virtual void execute ()**
  Execute the algorithm.
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the visitor to an FlaggedElmPtr.
- **virtual ~RangeSelector ()**

Protected members

- **virtual void handleBeamline (const FlaggedElmPtr&)**
  The operation to be done for beamlines.

  When overriding, make sure the beamline members are handled.
- **virtual void handleElement (const FlaggedElmPtr&)**
  The operation to be done for elements.
- **RangeRep itsRange**
  Working data for range.

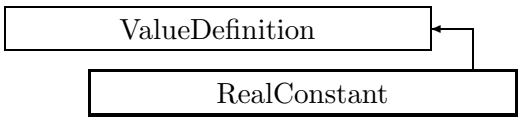


Figure 2.119: Inheritance for class RealConstant

2.159 class RealConstant

The REAL CONSTANT definition.

Type:	Instantiable
Superclasses:	public ValueDefinition
Include file:	./ValueDefinitions/RealConstant.hh

Synopsis (including inherited members)

Public members

- **RealConstant ()**
  Exemplar constructor. (see Section 2.159.1)
- **RealConstant (const string&,RealConstant*,double)**
  Constructor for built-in constants. (see Section 2.159.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.159.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual RealConstant* clone (const string&)**
  Make clone. (see Section 2.159.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read constant from DOOM data base. (see Section 2.159.1)
- **virtual void doomPut (DoomWriter&)const**
  Write constant to DOOM data base. (see Section 2.159.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual bool getBool ()const**
  Return logical value. (see Section 2.208.1)
- **virtual bool getBoolComponent (int)const**
  Return indexed logical value. (see Section 2.208.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.208.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual double getReal ()const**
  Return value. (see Section 2.159.1)
- **virtual double getRealComponent (int)const**
  Return indexed real value. (see Section 2.208.1)
- **virtual string getString ()const**
  Return string value. (see Section 2.208.1)
- **virtual string getStringComponent (int)const**
  Return indexed string value. (see Section 2.208.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the constant. (see Section 2.159.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.208.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.208.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)

- **Attribute& value ()**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **const Attribute& value ()const**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **virtual ~RealConstant ()**
  (see Section 2.159.1)

2.159.1 Detailed descriptions

Public members

- **RealConstant ()**
  Exemplar constructor.
- **RealConstant (const string&,RealConstant*,double)**
  Constructor for built-in constants.
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced.

  Always false for constants.
- **virtual RealConstant* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read constant from DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write constant to DOOM data base.
- **virtual double getReal ()const**
  Return value.
- **virtual void print (std::ostream&)const**
  Print the constant.
- **virtual ~RealConstant ()**

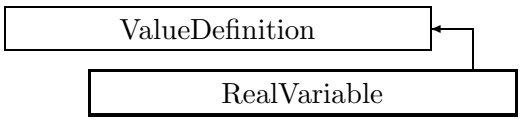


Figure 2.120: Inheritance for class RealVariable

2.160 class RealVariable

The REAL VARIABLE definition.

Type:	Instantiable
Superclasses:	public ValueDefinition
Include file:	./ValueDefinitions/RealVariable.hh

Synopsis (including inherited members)

Public members

- **RealVariable ()**
  Exemplar constructor. (see Section 2.160.1)
- **RealVariable (const string&,RealVariable*,double)**
  Constructor for built-in variables. (see Section 2.160.1)
- **virtual bool canReplaceBy (Object*)**
  Test for allowed replacement. (see Section 2.160.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual RealVariable* clone (const string&)**
  Make clone. (see Section 2.160.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read variable from DOOM data base. (see Section 2.160.1)
- **virtual void doomPut (DoomWriter&)const**
  Write variable to DOOM data base. (see Section 2.160.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual bool getBool ()const**
  Return logical value. (see Section 2.208.1)
- **virtual bool getBoolComponent (int)const**
  Return indexed logical value. (see Section 2.208.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.208.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual double getReal ()const**
  Return value. (see Section 2.160.1)
- **virtual double getRealComponent (int)const**
  Return indexed real value. (see Section 2.208.1)
- **virtual string getString ()const**
  Return string value. (see Section 2.208.1)
- **virtual string getStringComponent (int)const**
  Return indexed string value. (see Section 2.208.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the variable. (see Section 2.160.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.208.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.208.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)

- **Attribute& value ()**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **const Attribute& value ()const**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **virtual ~RealVariable ()**
  (see Section 2.160.1)

2.160.1 Detailed descriptions

Public members

- **RealVariable ()**
  Exemplar constructor.
- **RealVariable (const string&,RealVariable*,double)**
  Constructor for built-in variables.
- **virtual bool canReplaceBy (Object*)**
  Test for allowed replacement.

  True, if **rhs** is a real variable.
- **virtual RealVariable* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read variable from DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write variable to DOOM data base.
- **virtual double getReal ()const**
  Return value.
- **virtual void print (std::ostream&)const**
  Print the variable.
- **virtual ~RealVariable ()**

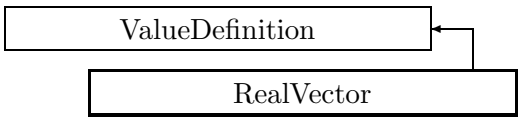


Figure 2.121: Inheritance for class RealVector

2.161 class RealVector

The REAL VECTOR definition.

Type:	Instantiable
Superclasses:	public ValueDefinition
Include file:	./ValueDefinitions/RealVector.hh

Synopsis (including inherited members)

Public members

- **RealVector ()**
  Exemplar constructor. (see Section 2.161.1)
- **virtual bool canReplaceBy (Object*)**
  Test for allowed replacement. (see Section 2.161.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual RealVector* clone (const string&)**
  Make clone. (see Section 2.161.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read the vector from the DOOM data base. (see Section 2.161.1)
- **virtual void doomPut (DoomWriter&)const**
  Write the vector to the DOOM data base. (see Section 2.161.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual bool getBool ()const**
  Return logical value. (see Section 2.208.1)

- **virtual bool getBoolComponent (int)const**
  Return indexed logical value. (see Section 2.208.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.208.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual double getReal ()const**
  Return real value. (see Section 2.208.1)
- **virtual double getRealComponent (int)const**
  Return indexed value. (see Section 2.161.1)
- **virtual string getString ()const**
  Return string value. (see Section 2.208.1)
- **virtual string getStringComponent (int)const**
  Return indexed string value. (see Section 2.208.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the vector. (see Section 2.161.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.208.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.208.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **Attribute& value ()**
  Return the attribute representing the value of the definition. (see Section 2.208.1)

- **const Attribute& value ()const**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **virtual ~RealVector ()**
  (see Section 2.161.1)

2.161.1 Detailed descriptions

Public members

- **RealVector ()**
  Exemplar constructor.
- **virtual bool canReplaceBy (Object*)**
  Test for allowed replacement.

  True, if **rhs** is a real vector.
- **virtual RealVector* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read the vector from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write the vector to the DOOM data base.
- **virtual double getRealComponent (int)const**
  Return indexed value.
- **virtual void print (std::ostream&)const**
  Print the vector.
- **virtual ~RealVector ()**

2.162 class RegularExpression

A regular expression.
This class encapsulates the UNIX regular expressions as defined in regexp(5). It provides a simple interface to the POSIX-compliant regcomp/regexec package.

Type:	Instantiable
Include file:	./Utilities/RegularExpression.hh

Synopsis (including inherited members)

Public members

- **bool OK ()const**
  Check the regular expression for sanity. (see Section 2.162.1)
- **RegularExpression (const string&,bool)**
  Constructor. (see Section 2.162.1)
- **RegularExpression (const RegularExpression&)**
  (see Section 2.162.1)
- **bool match (const string&)const**
  Match a string against the pattern. (see Section 2.162.1)
- **~RegularExpression ()**
  (see Section 2.162.1)

2.162.1 Detailed descriptions

Public members

- **bool OK ()const**
  Check the regular expression for sanity.
- **RegularExpression (const string&,bool)**
  Constructor.

  Construct regular expression from **pattern**. If **ignore** is true, the expression ignores upper/lower case.
- **RegularExpression (const RegularExpression&)**
- **bool match (const string&)const**
  Match a string against the pattern.
- **~RegularExpression ()**

Figure 2.122: Inheritance for class Replacer

2.163 class Replacer

A visitor which replaces all references to named element by a new version.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Lines/Replacer.hh

Synopsis (including inherited members)

Public members

- **Replacer (const Beamline&,const string&,ElementBase*)**
  Constructor. (see Section 2.163.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the visitor to an FlaggedElmPtr. (see Section 2.163.1)
- **virtual ~Replacer ()**
  (see Section 2.163.1)

2.163.1 Detailed descriptions

Public members

- **Replacer (const Beamline&,const string&,ElementBase*)**
  Constructor.

  Attach visitor to a beamline, remember the range.
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the visitor to an FlaggedElmPtr.
- **virtual ~Replacer ()**

2.164 class SBend

Type:	Instantiable

2.165 class SBendWrapper

Type:	Instantiable

2.166 class SFunction

Functions of arc length.
This class handles the position funcitions SI(), SC(), and SO().

Type:	Instantiable
Include file:	./Expressions/SFunction.hh

Synopsis (including inherited members)

Public members

- **SFunction ()**
  Default constructor. (see Section 2.166.1)
- **static double arcCtr ()**
  Return arc length at center SC(). (see Section 2.166.1)
- **static double arcIn ()**
  Return arc length at entrance SI(). (see Section 2.166.1)
- **static double arcOut ()**
  Return arc length at exit SO(). (see Section 2.166.1)
- **void reset ()**
  Reset the arc length to zero. (see Section 2.166.1)
- **void update (double)**
  Advance position by element length. (see Section 2.166.1)
- **~SFunction ()**
  Destructor. (see Section 2.166.1)

2.166.1 Detailed descriptions

Public members

- **SFunction ()**
  Default constructor.

  This constructor resets the arc length and registers **this** as the current arc length function. Only one such function may be active at any time.
- **static double arcCtr ()**
  Return arc length at center SC().
- **static double arcIn ()**
  Return arc length at entrance SI().
- **static double arcOut ()**
  Return arc length at exit SO().
- **void reset ()**
  Reset the arc length to zero.

- **void update (double)**
  Advance position by element length.
- **~SFunction ()**
  Destructor.

  Unregister **this** as the current arc length function.

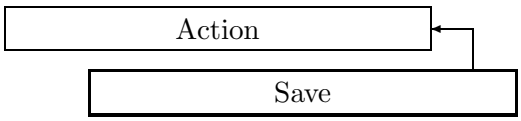


Figure 2.123: Inheritance for class Save

2.167 class Save

The SAVE command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Save.hh

Synopsis (including inherited members)

Public members

- **Save ()**
  Exemplar constructor. (see Section 2.167.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Save* clone (const string&)**
  Make clone. (see Section 2.167.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.167.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.167.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Save ()**
  (see Section 2.167.1)

2.167.1 Detailed descriptions

Public members

- **Save ()**
  Exemplar constructor.
- **virtual Save* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Save ()**

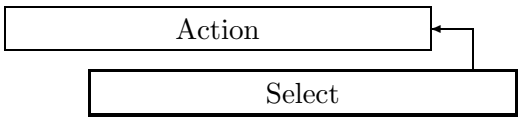


Figure 2.124: Inheritance for class Select

2.168 class Select

The SELECT command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Select.hh

Synopsis (including inherited members)

Public members

- **Select ()**
  Exemplar constructor. (see Section 2.168.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Select* clone (const string&)**
  Make clone. (see Section 2.168.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.168.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Select ()**
  (see Section 2.168.1)

2.168.1 Detailed descriptions

Public members

- **Select ()**
  Exemplar constructor.
- **virtual Select* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Select ()**

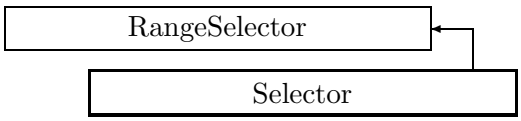


Figure 2.125: Inheritance for class Selector

2.169 class Selector

Set selection flags for a given range in a beam line.

Type:	Instantiable
Superclasses:	public RangeSelector
Include file:	./Tables/Selector.hh

Synopsis (including inherited members)

Public members

- **Selector (const Beamline&,const RangeRep&,const string&,const string&,const string&)**
  Constructor. (see Section 2.169.1)
- **virtual void execute ()**
  Execute the selection. (see Section 2.169.1)
- **int getCount ()const**
  Return the count of selected elements. (see Section 2.169.1)
- **virtual void visitFlaggedElmPtr (const FlaggedElmPtr&)**
  Apply the visitor to an FlaggedElmPtr. (see Section 2.158.1)
- **virtual ~Selector ()**
  (see Section 2.169.1)

Protected members

- **virtual void handleBeamline (const FlaggedElmPtr&)**
  The operation to be done for beamlines. (see Section 2.158.1)
- **virtual void handleElement (const FlaggedElmPtr&)**
  The operation to be done for elements. (see Section 2.169.1)
- **RangeRep itsRange**
  Working data for range. (see Section 2.158.1)

2.169.1 Detailed descriptions

Public members

- **Selector (const Beamline&,const RangeRep&,const string&,const string&,const string&)**
  Constructor.

Attach visitor to **bl**. Remember range **range**, class name **cName**, type name **tName**, and pattern string **pString**.

- **virtual void execute ()**
  Execute the selection.
- **int getCount ()const**
  Return the count of selected elements.
- **virtual ~Selector ()**

Protected members

- **virtual void handleElement (const FlaggedElmPtr&)**
  The operation to be done for elements.

2.170 class Separator

Type:	Instantiable

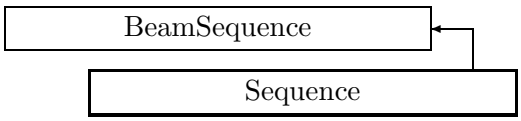


Figure 2.126: Inheritance for class Sequence

2.171 class Sequence

The SEQUENCE definition.

Type:	Instantiable
Superclasses:	public BeamSequence
Include file:	./Lines/Sequence.hh

Synopsis (including inherited members)

Public members

- **enum ReferenceType**
  Reference for element positioning. (see Section 2.64.1)
- **Sequence ()**
  Exemplar constructor. (see Section 2.171.1)
- **typedef TBeamline<SequenceMember> TLine**
  The type of a sequence line. (see Section 2.171.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.64.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Sequence* clone (const string&)**
  Make clone. (see Section 2.171.1)
- **virtual Sequence* copy (const string&)**
  Make copy of the sequence line. (see Section 2.171.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read the sequence from the DOOM data base. (see Section 2.171.1)
- **virtual void doomPut (DoomWriter&)const**
  Write the sequence to the DOOM data base. (see Section 2.171.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual TLine* fetchLine ()const**
  Return the embedded CLASSIC beam line. (see Section 2.171.1)

- **static BeamSequence* find (const string&)**
  Find a BeamSequence by name. (see Section 2.27.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.27.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **inline ElementBase* getElement ()const**
  Return the embedded CLASSIC element. (see Section 2.64.1)
- **virtual double getEntrance (ReferenceType)const**
  Return the arc length from origin to entrance. (see Section 2.171.1)
- **virtual double getExit (ReferenceType)const**
  Return the arc length from origin to exit. (see Section 2.171.1)
- **virtual double getLength ()const**
  Return sequence length. (see Section 2.171.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **ReferenceType getReference ()const**
  Return the reference type flag. (see Section 2.171.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)

- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a sequence template. (see Section 2.171.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse sequence. (see Section 2.171.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print sequence. (see Section 2.171.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references to elements. (see Section 2.171.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **inline void setElement (ElementBase*)**
  Assign new CLASSIC element. (see Section 2.64.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set shared flag. (see Section 2.64.1)

- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.64.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.64.1)
- **void storeLine (TLine&)**
  Store sequence line. (see Section 2.171.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update the embedded CLASSIC beam line. (see Section 2.171.1)
- **virtual ~Sequence ()**
  (see Section 2.171.1)

2.171.1 Detailed descriptions

Public members

- **Sequence ()**
  Exemplar constructor.
- **typedef TBeamline<SequenceMember> TLine**
  The type of a sequence line.
- **virtual Sequence* clone (const string&)**
  Make clone.

  The new object is an empty sequence, it will be filled in by the parser.
- **virtual Sequence* copy (const string&)**
  Make copy of the sequence line.
- **virtual void doomGet (const DoomReader&)**
  Read the sequence from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write the sequence to the DOOM data base.
- **virtual TLine* fetchLine ()const**
  Return the embedded CLASSIC beam line.

  The result it the ideal line.
- **virtual double getEntrance (ReferenceType)const**
  Return the arc length from origin to entrance.
- **virtual double getExit (ReferenceType)const**
  Return the arc length from origin to exit.
- **virtual double getLength ()const**
  Return sequence length.

- **ReferenceType getReference ()const**
  Return the reference type flag.

  "ENTRY", "CENTRE", or "EXIT".

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a sequence template.

  Arguments:

  > Name to be given to the template.
  >
  > Token stream to be read for the sequence line.
  >
  > Statement to be read for the arguments.

- **virtual void parse (Statement&)**
  Parse sequence.

- **virtual void print (std::ostream&)const**
  Print sequence.

- **virtual void replace (Object*,Object*)**
  Replace references to elements.

- **void storeLine (TLine&)**
  Store sequence line.

  Assign to the underlying ideal line and re-insert the drifts.

- **virtual void update ()**
  Update the embedded CLASSIC beam line.

  Recompute drift lengths.

- **virtual ~Sequence ()**

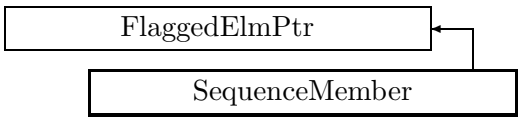


Figure 2.127: Inheritance for class SequenceMember

2.172 class SequenceMember

A member of a SEQUENCE.

Type:	Instantiable
Superclasses:	public FlaggedElmPtr
Include file:	./Lines/SequenceMember.hh

Synopsis (including inherited members)

Public members

- **enum MemberType**
  (see Section 2.172.1)
- **enum PositionType**
  (see Section 2.172.1)
- **SequenceMember ()**
  (see Section 2.172.1)
- **SequenceMember (const SequenceMember&)**
  (see Section 2.172.1)
- **PositionType itsFlag**
  Flag word. (see Section 2.172.1)
- **double itsPosition**
  The position attribute ("AT" or "DRIFT"). (see Section 2.172.1)
- **MemberType itsType**
  Type word. (see Section 2.172.1)
- **void setLength (double)**
  Store the drift length for a generated drift. (see Section 2.172.1)
- **~SequenceMember ()**
  (see Section 2.172.1)

2.172.1 Detailed descriptions

Public members

- **enum MemberType**

- **enum PositionType**

- **SequenceMember ()**

- **SequenceMember (const SequenceMember&)**

- **PositionType itsFlag**
  Flag word.

  Specific to the sequence parser, tells how the element is related to the preceding one.

- **double itsPosition**
  The position attribute ("AT" or "DRIFT").

- **MemberType itsType**
  Type word.

  Tells how the element is defined.

- **void setLength (double)**
  Store the drift length for a generated drift.

- **~SequenceMember ()**

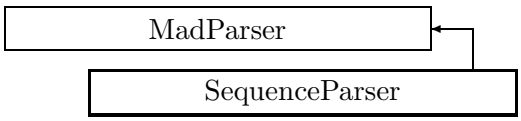


Figure 2.128: Inheritance for class SequenceParser

2.173 class SequenceParser

The parser for SEQUENCE members.
Recognises the format for all possible SEQUENCE members.

Type:	Instantiable
Superclasses:	public MadParser
Include file:	./Lines/SequenceParser.hh

Synopsis (including inherited members)

Public members

- **typedef std::list<Reference> RefList**
  (see Section 2.173.1)
- **SequenceParser (Sequence*)**
  Constructor. (see Section 2.173.1)
- **typedef Sequence::TLine TLine**
  (see Section 2.173.1)
- **virtual void parse (Statement&)const**
  Parse sequence member. (see Section 2.173.1)
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream. (see Section 2.118.1)
- **static Token readToken ()**
  Return next input token. (see Section 2.118.1)
- **virtual void run ()const**
  Read current stream. (see Section 2.118.1)
- **virtual void run (TokenStream*)const**
  Read given stream. (see Section 2.118.1)
- **void stop ()const**
  Set stop flag. (see Section 2.118.1)
- **virtual ~SequenceParser ()**
  (see Section 2.173.1)

2.173.1 Detailed descriptions

Public members

- **typedef std::list<Reference> RefList**

- **SequenceParser (Sequence*)**
  Constructor.

  Assign the current sequence being parsed.

- **typedef Sequence::TLine TLine**

- **virtual void parse (Statement&)const**
  Parse sequence member.

- **virtual ~SequenceParser ()**

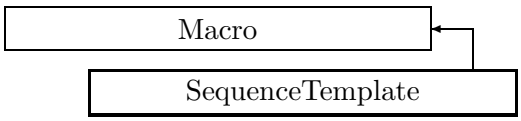


Figure 2.129: Inheritance for class SequenceTemplate

2.174 class SequenceTemplate

An "archetype" for a SEQUENCE with arguments.

Type:	Instantiable
Superclasses:	public Macro
Include file:	./Lines/SequenceTemplate.hh

Synopsis (including inherited members)

Public members

- **SequenceTemplate ()**
  (see Section 2.174.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual SequenceTemplate* clone (const string&)**
  Make clone. (see Section 2.174.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.98.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Make line instance. (see Section 2.174.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a sequence template. (see Section 2.174.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **void parse (TokenStream&,Statement&)**
  Parse the sequence template. (see Section 2.174.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseActuals (Statement&)**
  Parse actual arguments. (see Section 2.98.1)
- **virtual void parseFormals (Statement&)**
  Parse formal arguments. (see Section 2.98.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)

- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.98.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.98.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~SequenceTemplate ()**
  (see Section 2.174.1)

2.174.1 Detailed descriptions

Public members

- **SequenceTemplate ()**

- **virtual SequenceTemplate* clone (const string&)**
  Make clone.

  Throw MadException, since the template cannot be cloned.

- **virtual Object* makeInstance (const string&,Statement&)**
  Make line instance.

  The instance gets the name **name**, and its actual arguments are read from **stat**.

- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Make a sequence template.

  Return NULL, since one cannot make a template from a template.

- **void parse (TokenStream&,Statement&)**
  Parse the sequence template.

- **virtual ~SequenceTemplate ()**

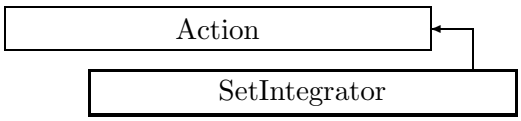


Figure 2.130: Inheritance for class SetIntegrator

2.175 class SetIntegrator

The SETINT command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./PhysicsActions/SetIntegrator.hh

Synopsis (including inherited members)

Public members

- **SetIntegrator ()**
  Exemplar constructor. (see Section 2.175.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual SetIntegrator* clone (const string&)**
  Make clone. (see Section 2.175.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.175.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~SetIntegrator ()**
  (see Section 2.175.1)

2.175.1 Detailed descriptions

Public members

- **SetIntegrator ()**
  Exemplar constructor.
- **virtual SetIntegrator* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~SetIntegrator ()**

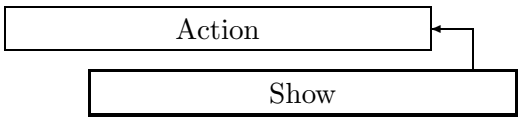


Figure 2.131: Inheritance for class Show

2.176 class Show

The SHOW command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Show.hh

Synopsis (including inherited members)

Public members

- **Show ()**
  Exemplar constructor. (see Section 2.176.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Show* clone (const string&)**
  Make clone. (see Section 2.176.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.176.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.176.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Show ()**
  (see Section 2.176.1)

2.176.1 Detailed descriptions

Public members

- **Show ()**
  Exemplar constructor.
- **virtual Show* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Show ()**

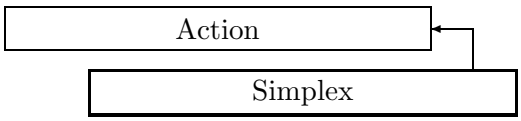


Figure 2.132: Inheritance for class Simplex

2.177 class Simplex

The SIMPLEX command.
This class encapsulates a minimisation according to the SIMPLEX method taken from the MINUIT package.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/Simplex.hh

Synopsis (including inherited members)

Public members

- **Simplex ()**
  Exemplar constructor. (see Section 2.177.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Simplex* clone (const string&)**
  Make clone. (see Section 2.177.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.177.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)

- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)

- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Simplex ()**
  (see Section 2.177.1)

2.177.1 Detailed descriptions

Public members

- **Simplex ()**
  Exemplar constructor.
- **virtual Simplex* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Simplex ()**

2.178 class Solenoid

Type:	Instantiable

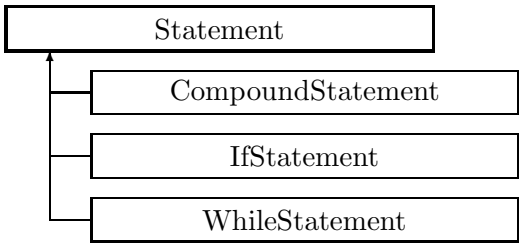


Figure 2.133: Inheritance for class Statement

2.179 class Statement

Type:	Instantiable

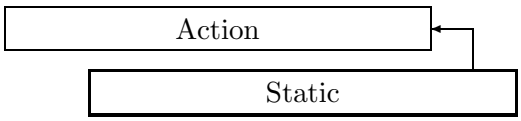


Figure 2.134: Inheritance for class Static

2.180 class Static

The STATIC command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./PhysicsActions/Static.hh

Synopsis (including inherited members)

Public members

- **Static ()**
  Exemplar constructor. (see Section 2.180.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Static* clone (const string&)**
  Make clone. (see Section 2.180.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.180.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Static ()**
  (see Section 2.180.1)

2.180.1 Detailed descriptions

Public members

- **Static ()**
  Exemplar constructor.
- **virtual Static* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Static ()**

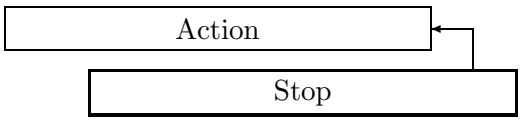


Figure 2.135: Inheritance for class Stop

2.181 class Stop

The STOP command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Stop.hh

Synopsis (including inherited members)

Public members

- **Stop ()**
  Exemplar constructor. (see Section 2.181.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Stop* clone (const string&)**
  Make clone. (see Section 2.181.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.181.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Stop ()**
  (see Section 2.181.1)

2.181.1 Detailed descriptions

Public members

- **Stop ()**
  Exemplar constructor.
- **virtual Stop* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Stop ()**

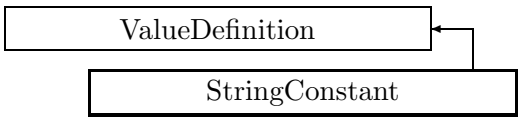


Figure 2.136: Inheritance for class StringConstant

2.182 class StringConstant

The STRING CONSTANT definition.

Type:	Instantiable
Superclasses:	public ValueDefinition
Include file:	./ValueDefinitions/StringConstant.hh

Synopsis (including inherited members)

Public members

- **StringConstant ()**
  Exemplar constructor. (see Section 2.182.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.182.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual StringConstant* clone (const string&)**
  Make clone. (see Section 2.182.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read the constant from the DOOM data base. (see Section 2.182.1)
- **virtual void doomPut (DoomWriter&)const**
  Write the constant to the DOOM data base. (see Section 2.182.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual bool getBool ()const**
  Return logical value. (see Section 2.208.1)

- **virtual bool getBoolComponent (int)const**
  Return indexed logical value. (see Section 2.208.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.208.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual double getReal ()const**
  Return real value. (see Section 2.208.1)
- **virtual double getRealComponent (int)const**
  Return indexed real value. (see Section 2.208.1)
- **virtual string getString ()const**
  Return value. (see Section 2.182.1)
- **virtual string getStringComponent (int)const**
  Return indexed string value. (see Section 2.208.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the constant. (see Section 2.182.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.208.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.208.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **Attribute& value ()**
  Return the attribute representing the value of the definition. (see Section 2.208.1)

- **const Attribute& value ()const**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **virtual ~StringConstant ()**
  (see Section 2.182.1)

2.182.1 Detailed descriptions

Public members

- **StringConstant ()**
  Exemplar constructor.
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced.

  True, if **rhs** is a string constant.
- **virtual StringConstant* clone (const string&)**
  Make clone.
- **virtual void doomGet (const DoomReader&)**
  Read the constant from the DOOM data base.
- **virtual void doomPut (DoomWriter&)const**
  Write the constant to the DOOM data base.
- **virtual string getString ()const**
  Return value.
- **virtual void print (std::ostream&)const**
  Print the constant.
- **virtual ~StringConstant ()**

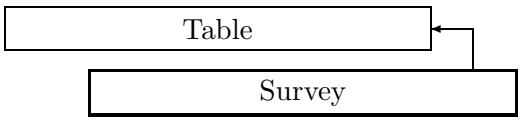


Figure 2.137: Inheritance for class Survey

2.183 class Survey

The SURVEY command.

Type:	Instantiable
Superclasses:	public Table
Include file:	./Tables/Survey.hh

Synopsis (including inherited members)

Public members

- **struct Cell**
  Descriptor for printing a table cell. (see Section 2.187.1)
- **typedef std::vector<Cell> CellArray**
  An array of cell descriptors. (see Section 2.187.1)
- **class Row**
  The class for one row of the survey table. (see Section 2.183.1)
- **Survey ()**
  Exemplar constructor. (see Section 2.183.1)
- **typedef TBeamline<Row> TLine**
  (see Section 2.183.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.187.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Survey* clone (const string&)**
  Make clone. (see Section 2.183.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Store the table in the DOOM data base. (see Section 2.183.1)
- **virtual void execute ()**
  Check validity of survey definition. (see Section 2.183.1)

- **virtual void fill ()**
  Fill the buffer using the survey algorithm. (see Section 2.183.1)
- **static Table* find (const string&)**
  Find named Table. (see Section 2.187.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.187.1)
- **virtual double getCell (const PlaceRep&,const string&)**
  Return a selected value in a selected row. (see Section 2.183.1)
- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column **col** of this table, limited by **range**. (see Section 2.183.1)
- **const Row& getCurrent ()const**
  Return current row of table. (see Section 2.183.1)
- **virtual CellArray getDefault ()const**
  Return the default print columns. (see Section 2.183.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **virtual double getLength ()**
  Return the length of the table. (see Section 2.183.1)
- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline. (see Section 2.183.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **const Euclid3D& getMap (const Row&)const**
  Position and orientation of local system. (see Section 2.183.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **double getPhi (const Row&,int,int)const**
  Rotation about X. (see Section 2.183.1)
- **double getPsi (const Row&,int,int)const**
  Rotation about Z. (see Section 2.183.1)

- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row, possible user-defined. (see Section 2.183.1)
- **double getS (const Row&,int,int)const**
  Arc length for given row. (see Section 2.183.1)
- **double getTheta (const Row&,int,int)const**
  Rotation about Y. (see Section 2.183.1)
- **double getW (const Row&,int,int)const**
  Local axis vectors. (see Section 2.183.1)
- **double getX (const Row&,int,int)const**
  X component of displacement. (see Section 2.183.1)
- **double getY (const Row&,int,int)const**
  Y component of displacement. (see Section 2.183.1)
- **double getZ (const Row&,int,int)const**
  Z component of displacement. (see Section 2.183.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **virtual void invalidate ()**
  Mark this table as invalid, if it is dynamic. (see Section 2.187.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **virtual bool isDependent (const string&)const**
  Find dependency. (see Section 2.183.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return column. (see Section 2.183.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)

- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table. (see Section 2.183.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual bool matches (Table*)const**
  Check compatibility. (see Section 2.183.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print list for the table. (see Section 2.183.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.187.1)

- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.187.1)
- **void tfsTableDescriptors (std::ostream&)const**
  Write TFS descriptors existing for all tables. (see Section 2.187.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Survey ()**
  (see Section 2.183.1)

2.183.1 Detailed descriptions

Public members

- **class Row**
  The class for one row of the survey table.
- **Survey ()**
  Exemplar constructor.
- **typedef TBeamline<Row> TLine**
- **virtual Survey* clone (const string&)**
  Make clone.
- **virtual void doomPut (DoomWriter&)const**
  Store the table in the DOOM data base.
- **virtual void execute ()**
  Check validity of survey definition.
- **virtual void fill ()**
  Fill the buffer using the survey algorithm.
- **virtual double getCell (const PlaceRep&,const string&)**
  Return a selected value in a selected row.
- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column col of this table, limited by range.
- **const Row& getCurrent ()const**
  Return current row of table.
- **virtual CellArray getDefault ()const**
  Return the default print columns.
- **virtual double getLength ()**
  Return the length of the table.

- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline.
- **const Euclid3D& getMap (const Row&)const**
  Position and orientation of local system.
- **double getPhi (const Row&,int,int)const**
  Rotation about X.
- **double getPsi (const Row&,int,int)const**
  Rotation about Z.
- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row, possible user-defined.
- **double getS (const Row&,int,int)const**
  Arc length for given row.
- **double getTheta (const Row&,int,int)const**
  Rotation about Y.
- **double getW (const Row&,int,int)const**
  Local axis vectors.

  First index (1 ... 3) is coordinate, second index (1 ... 3) is vector.
- **double getX (const Row&,int,int)const**
  X component of displacement.
- **double getY (const Row&,int,int)const**
  Y component of displacement.
- **double getZ (const Row&,int,int)const**
  Z component of displacement.
- **virtual bool isDependent (const string&)const**
  Find dependency.

  Return true, if this table depends on the named object.
- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return column.

  Return an expression which denotes the selected column, identified by its name.
- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table.
- **virtual bool matches (Table*)const**
  Check compatibility.

  True, if **rhs** is a survey table.
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print list for the table.
- **virtual ~Survey ()**

2.184 class Surveyor

Type:	Instantiable

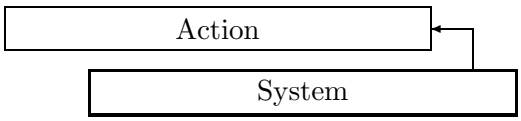


Figure 2.138: Inheritance for class System

2.185 class System

The SYSTEM command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/System.hh

Synopsis (including inherited members)

Public members

- **System ()**
  Exemplar constructor. (see Section 2.185.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual System* clone (const string&)**
  Make clone. (see Section 2.185.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.185.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.185.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~System ()**
  (see Section 2.185.1)

2.185.1 Detailed descriptions

Public members

- **System ()**
  Exemplar constructor.
- **virtual System* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~System ()**

2.186 template class TValue <class>

Type:	Instantiable

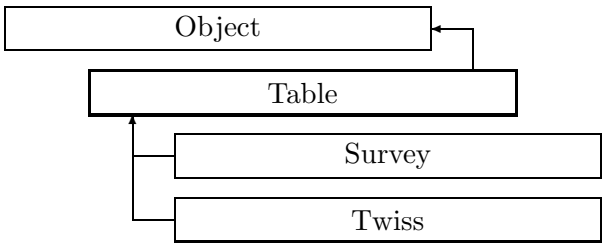


Figure 2.139: Inheritance for class Table

2.187 class Table

The base class for all MAD tables.
It implements the common behaviour of tables, it can also be used via dynamic casting to determine whether an object represents a table.

Type:	abstract
Superclasses:	public Object
Include file:	./AbstractObjects/Table.hh

Synopsis (including inherited members)

Public members

- **struct Cell**
  Descriptor for printing a table cell. (see Section 2.187.1)
- **typedef std::vector<Cell> CellArray**
  An array of cell descriptors. (see Section 2.187.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.187.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual void fill ()**
  Refill the buffer. (see Section 2.187.1)

- **static Table* find (const string&)**
  Find named Table. (see Section 2.187.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.187.1)
- **virtual double getCell (const PlaceRep&,const string&)**
  Return value in selected table cell. (see Section 2.187.1)
- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column **col** of this table, limited by **range**. (see Section 2.187.1)
- **virtual CellArray getDefault ()const**
  Return the default print columns. (see Section 2.187.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **virtual double getLength ()**
  Return the length of the table. (see Section 2.187.1)
- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline. (see Section 2.187.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row. (see Section 2.187.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **virtual void invalidate ()**
  Mark this table as invalid, if it is dynamic. (see Section 2.187.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **virtual bool isDependent (const string&)const**
  Find out if table depends on the object identified by **name**. (see Section 2.187.1)

- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return expression to compute the value in a column of the table. (see Section 2.187.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table. (see Section 2.187.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual bool matches (Table*)const**
  Check that **rhs** is of same type as **this**. (see Section 2.187.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print list for the table. (see Section 2.187.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)

- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.187.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.187.1)
- **void tfsTableDescriptors (std::ostream&)const**
  Write TFS descriptors existing for all tables. (see Section 2.187.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Table ()**
  (see Section 2.187.1)

Protected members

- **Table (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.187.1)
- **Table (const string&,Table*)**
  Constructor for clones. (see Section 2.187.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool dynamic**
  Flag dynamic table. (see Section 2.187.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)

- **bool modified**
  Dirty flag. (see Section 2.143.1)
- **bool refill**
  Refill flag. (see Section 2.187.1)

2.187.1 Detailed descriptions

Public members

- **struct Cell**
  Descriptor for printing a table cell.
- **typedef std::vector<Cell> CellArray**
  An array of cell descriptors.
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced.

  A table cannot be replaced, thus this method never returns true.
- **virtual void fill ()**
  Refill the buffer.

  If **refill** is true, call the table algorithm to fill the buffer.
- **static Table* find (const string&)**
  Find named Table.

  If a table with name **name** exists, return a pointer to that table. If no such table exists, throw MadException.
- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "TABLE".
- **virtual double getCell (const PlaceRep&,const string&)**
  Return value in selected table cell.

  Return the value stored in the table cell identified by the row at **row** and the column name **col**.
- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column col of this table, limited by range.
- **virtual CellArray getDefault ()const**
  Return the default print columns.

  Returns an array of column descriptors, which, when applied to a row of the table, gives the default print columns for this table.
- **virtual double getLength ()**
  Return the length of the table.

  Returns the geometric length of the underlying beam line.

- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline.

  Returns the CLASSIC beamline representing the table. The data of the table are attached to each position in the line.

- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row.

  Returns the values stored in the row specified by the first argument, with columns selected by the names stored in the second argument.

- **virtual void invalidate ()**
  Mark this table as invalid, if it is dynamic.

- **virtual bool isDependent (const string&)const**
  Find out if table depends on the object identified by name.

  Must be overridden in derived classes.

- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return expression to compute the value in a column of the table.

  The row is identified by setting a "current" row pointer in **listTFS()** or **printTable()**. The expression then computes the value in the column identified by the argument.

- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table.

- **virtual bool matches (Table*)const**
  Check that rhs is of same type as this.

- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print list for the table.

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always true for tables.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always true for tables.

- **void tfsTableDescriptors (std::ostream&)const**
  Write TFS descriptors existing for all tables.

- **virtual ~Table ()**

Protected members

- **Table (int,const char*,const char*)**
  Constructor for exemplars.

- **Table (const string&,Table*)**
  Constructor for clones.

- **bool dynamic**
  Flag dynamic table.

  If true, the table is dynamic. If it is also invalidated, it will be refilled when fill() is called.

- **bool refill**
  Refill flag.

  If true, the table has been invalidated. If it is also dynammic, it will be refilled when fill() is called.

2.188 class TableRowRep

Representation of a table row reference.

Type:	Instantiable
Include file:	./AbstractObjects/TableRowRep.hh

Synopsis (including inherited members)

Public members

- **TableRowRep ()**
  Default constructor. (see Section 2.188.1)
- **TableRowRep (const string&,const PlaceRep&)**
  Constructor. (see Section 2.188.1)
- **TableRowRep (const TableRowRep&)**
  (see Section 2.188.1)
- **PlaceRep getPosition ()const**
  Return the row position representation. (see Section 2.188.1)
- **const string& getTabName ()const**
  Return the table name. (see Section 2.188.1)
- **const TableRowRep& operator= (const TableRowRep&)**
  (see Section 2.188.1)
- **std::ostream& print (std::ostream&)const**
  Print in input format. (see Section 2.188.1)
- **~TableRowRep ()**
  (see Section 2.188.1)

2.188.1 Detailed descriptions

Public members

- **TableRowRep ()**
  Default constructor.

  Constructs undefined reference.
- **TableRowRep (const string&,const PlaceRep&)**
  Constructor.

  Construct reference to the row identified by **row** of the table with name **name**.
- **TableRowRep (const TableRowRep&)**
- **PlaceRep getPosition ()const**
  Return the row position representation.

- **const string& getTabName ()const**
  Return the table name.
- **const TableRowRep& operator= (const TableRowRep&)**
- **std::ostream& print (std::ostream&)const**
  Print in input format.
- **~TableRowRep ()**

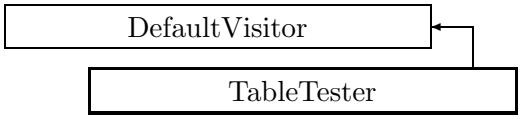


Figure 2.140: Inheritance for class TableTester

2.189 class TableTester

Test dependency of a table.
Visitor for testing for dependency of a table on a name.

Type:	Instantiable
Superclasses:	public DefaultVisitor
Include file:	./Tables/TableTester.hh

Synopsis (including inherited members)

Public members

- **TableTester (const Beamline&,const string&)**
  Constructor. (see Section 2.189.1)
- **virtual void applyDefault (const ElementBase&)**
  Apply default operation. (see Section 2.189.1)
- **~TableTester ()**
  (see Section 2.189.1)

2.189.1 Detailed descriptions

Public members

- **TableTester (const Beamline&,const string&)**
  Constructor.

  Attach visitor to **bl**, remember the name **name**.
- **virtual void applyDefault (const ElementBase&)**
  Apply default operation.

  Throw an exception, if the contained element has the given name.
- **~TableTester ()**

Figure 2.141: Inheritance for class ThickMapper

2.190 class ThickMapper

Build a map using a finite-length lens for each elements.
Multipole-like elements are done by expanding the Lie series.
Phase space coordinates numbering:

number	name	unit
0	x	metres
1	p_x/p_r	1
2	y	metres
3	p_y/p_r	1
4	$v * delta_t$	metres
5	$delta_p/p_r$	1

Where p_r is the constant reference momentum defining the reference frame velocity, m is the rest mass of the particles, and v is the instantaneous velocity of the particle.
Other units used:

quantity	unit
reference momentum	electron-volts
velocity	metres/second
accelerating voltage	volts
separator voltage	volts
frequencies	hertz
phase lags	$2 * pi$

Approximations used:

All elements are represented by maps for finite-length elements. For multipole-like elements the Lie series is used.

Geometric transformations ignore rotations about transverse axes and translations along the design orbit and truncate after second order.

Beam-beam elements are two-dimensional, and the second moment <x,y> of the opposite bunches vanish.

Type:	Instantiable
Superclasses:	public Mapper
Include file:	./Algorithms/ThickMapper.hh

Synopsis (including inherited members)

Public members

- **ThickMapper (const Beamline&,const PartData&,bool,bool)**
  Constructor. (see Section 2.190.1)

- **virtual void visitBeamBeam (const BeamBeam&)**
  Apply the algorithm to a BeamBeam. (see Section 2.190.1)
- **virtual void visitCollimator (const Collimator&)**
  Apply the algorithm to a collimator. (see Section 2.190.1)
- **virtual void visitCorrector (const Corrector&)**
  Apply the algorithm to a Corrector. (see Section 2.190.1)
- **virtual void visitDiagnostic (const Diagnostic&)**
  Apply the algorithm to a Diagnostic. (see Section 2.190.1)
- **virtual void visitDrift (const Drift&)**
  Apply the algorithm to a Drift. (see Section 2.190.1)
- **virtual void visitLambertson (const Lambertson&)**
  Apply the algorithm to a Lambertson. (see Section 2.190.1)
- **virtual void visitMarker (const Marker&)**
  Apply the algorithm to a Marker. (see Section 2.190.1)
- **virtual void visitMonitor (const Monitor&)**
  Apply the algorithm to a Monitor. (see Section 2.190.1)
- **virtual void visitMultipole (const Multipole&)**
  Apply the algorithm to a Multipole. (see Section 2.190.1)
- **virtual void visitRBend (const RBend&)**
  Apply the algorithm to a RBend. (see Section 2.190.1)
- **virtual void visitRFCavity (const RFCavity&)**
  Apply the algorithm to a RFCavity. (see Section 2.190.1)
- **virtual void visitRFQuadrupole (const RFQuadrupole&)**
  Apply the algorithm to a RFQuadrupole. (see Section 2.190.1)
- **virtual void visitSBend (const SBend&)**
  Apply the algorithm to a SBend. (see Section 2.190.1)
- **virtual void visitSeparator (const Separator&)**
  Apply the algorithm to a Separator. (see Section 2.190.1)
- **virtual void visitSeptum (const Septum&)**
  Apply the algorithm to a Septum. (see Section 2.190.1)
- **virtual void visitSolenoid (const Solenoid&)**
  Apply the algorithm to a Solenoid. (see Section 2.190.1)
- **virtual ~ThickMapper ()**
  (see Section 2.190.1)

2.190.1 Detailed descriptions

Public members

- **ThickMapper (const Beamline&,const PartData&,bool,bool)**
  Constructor.

  The beam line to be tracked is "bl". The particle reference data are taken from "data". If **revBeam** is true, the beam runs from s = C to s = 0. If **revTrack** is true, we track against the beam.

- **virtual void visitBeamBeam (const BeamBeam&)**
  Apply the algorithm to a BeamBeam.

- **virtual void visitCollimator (const Collimator&)**
  Apply the algorithm to a collimator.

- **virtual void visitCorrector (const Corrector&)**
  Apply the algorithm to a Corrector.

- **virtual void visitDiagnostic (const Diagnostic&)**
  Apply the algorithm to a Diagnostic.

- **virtual void visitDrift (const Drift&)**
  Apply the algorithm to a Drift.

- **virtual void visitLambertson (const Lambertson&)**
  Apply the algorithm to a Lambertson.

- **virtual void visitMarker (const Marker&)**
  Apply the algorithm to a Marker.

- **virtual void visitMonitor (const Monitor&)**
  Apply the algorithm to a Monitor.

- **virtual void visitMultipole (const Multipole&)**
  Apply the algorithm to a Multipole.

- **virtual void visitRBend (const RBend&)**
  Apply the algorithm to a RBend.

- **virtual void visitRFCavity (const RFCavity&)**
  Apply the algorithm to a RFCavity.

- **virtual void visitRFQuadrupole (const RFQuadrupole&)**
  Apply the algorithm to a RFQuadrupole.

- **virtual void visitSBend (const SBend&)**
  Apply the algorithm to a SBend.

- **virtual void visitSeparator (const Separator&)**
  Apply the algorithm to a Separator.

- **virtual void visitSeptum (const Septum&)**
  Apply the algorithm to a Septum.

- **virtual void visitSolenoid (const Solenoid&)**
  Apply the algorithm to a Solenoid.
- **virtual ~ThickMapper ()**

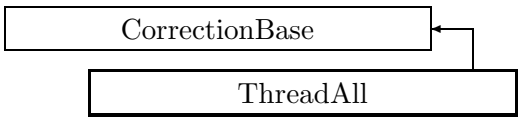


Figure 2.142: Inheritance for class ThreadAll

2.191 class ThreadAll

Class ThreadAll MAD command THREADALL.

Type:	Instantiable
Superclasses:	public CorrectionBase
Include file:	./Tables/ThreadAll.hh

Synopsis (including inherited members)

Public members

- **typedef TBeamline<Row> TLine**
  (see Section 2.38.1)
- **ThreadAll ()**
  Exemplar constructor. (see Section 2.191.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ThreadAll* clone (const string&)**
  Make clone. (see Section 2.191.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Check validity of the table definition. (see Section 2.191.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)

- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ThreadAll ()**
  (see Section 2.191.1)

2.191.1 Detailed descriptions

Public members

- **ThreadAll ()**
  Exemplar constructor.
- **virtual ThreadAll* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Check validity of the table definition.

- **virtual ~ThreadAll ()**

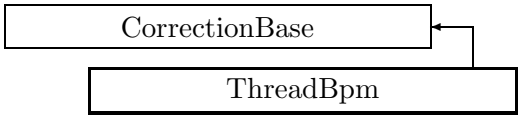


Figure 2.143: Inheritance for class ThreadBpm

2.192 class ThreadBpm

Class ThreadBpm MAD command THREADBPM.

Type:	Instantiable
Superclasses:	public CorrectionBase
Include file:	./Tables/ThreadBpm.hh

Synopsis (including inherited members)

Public members

- **typedef TBeamline<Row> TLine**
  (see Section 2.38.1)
- **ThreadBpm ()**
  Exemplar constructor. (see Section 2.192.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual ThreadBpm* clone (const string&)**
  Make clone. (see Section 2.192.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Check validity of the table definition. (see Section 2.192.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)

- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)

- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~ThreadBpm ()**
  (see Section 2.192.1)

2.192.1 Detailed descriptions

Public members

- **ThreadBpm ()**
  Exemplar constructor.
- **virtual ThreadBpm* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Check validity of the table definition.

- **virtual ~ThreadBpm ()**

2.193 class Timer

Timer class.
Encapsulates the UNIX wall time functions. Get Calendar date or time.

Type:	Instantiable
Include file:	./Utilities/Timer.hh

Synopsis (including inherited members)

Public members

- **Timer ()**
  Constructor. (see Section 2.193.1)
- **string date ()const**
  Return date. (see Section 2.193.1)
- **string time ()const**
  Return time. (see Section 2.193.1)
- **˜Timer ()**
  (see Section 2.193.1)

2.193.1 Detailed descriptions

Public members

- **Timer ()**
  Constructor.

  Store the system clock time.
- **string date ()const**
  Return date.
- **string time ()const**
  Return time.
- **˜Timer ()**

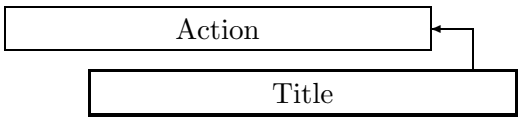


Figure 2.144: Inheritance for class Title

2.194 class Title

The TITLE command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Title.hh

Synopsis (including inherited members)

Public members

- **Title ()**
  Exemplar constructor. (see Section 2.194.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Title* clone (const string&)**
  Make clone. (see Section 2.194.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.194.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.194.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Title ()**
  (see Section 2.194.1)

2.194.1 Detailed descriptions

Public members

- **Title ()**
  Exemplar constructor.
- **virtual Title* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Title ()**

2.195 class Token

Type:	Instantiable

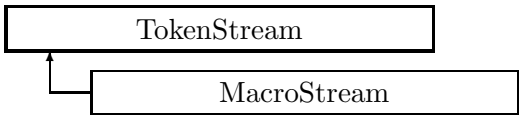


Figure 2.145: Inheritance for class TokenStream

2.196 class TokenStream

Type:	Instantiable

2.197 class Track

Hold data for tracking.
Acts as a communication area between the various tracking commands.

Type:	Instantiable
Include file:	./Track/Track.hh

Synopsis (including inherited members)

Public members

- **Track (BeamSequence*,const PartData&)**
  (see Section 2.197.1)
- **static Track* block**
  The block of track data. (see Section 2.197.1)
- **PartBunch bunch**
  The particle bunch to be tracked. (see Section 2.197.1)
- **TrackParser parser**
  The parser used during tracking. (see Section 2.197.1)
- **PartData reference**
  The reference data. (see Section 2.197.1)
- **BeamSequence* use**
  The lattice to be tracked through. (see Section 2.197.1)
- **~Track ()**
  (see Section 2.197.1)

2.197.1 Detailed descriptions

Public members

- **Track (BeamSequence*,const PartData&)**

- **static Track* block**
  The block of track data.
- **PartBunch bunch**
  The particle bunch to be tracked.
- **TrackParser parser**
  The parser used during tracking.
- **PartData reference**
  The reference data.
- **BeamSequence* use**
  The lattice to be tracked through.

- **~Track ()**

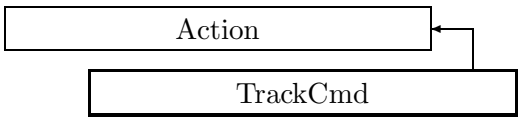


Figure 2.146: Inheritance for class TrackCmd

2.198 class TrackCmd

The TRACK command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Track/TrackCmd.hh

Synopsis (including inherited members)

Public members

- **TrackCmd ()**
  Exemplar constructor. (see Section 2.198.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual TrackCmd* clone (const string&)**
  Make clone. (see Section 2.198.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.198.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~TrackCmd ()**
  (see Section 2.198.1)

2.198.1 Detailed descriptions

Public members

- **TrackCmd ()**
  Exemplar constructor.
- **virtual TrackCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~TrackCmd ()**

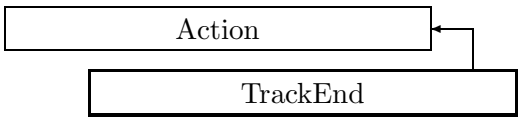


Figure 2.147: Inheritance for class TrackEnd

2.199 class TrackEnd

The ENDTRACK command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Track/TrackEnd.hh

Synopsis (including inherited members)

Public members

- **TrackEnd ()**
  Exemplar constructor. (see Section 2.199.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual TrackEnd* clone (const string&)**
  Make clone. (see Section 2.199.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.199.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~TrackEnd ()**
  (see Section 2.199.1)

2.199.1 Detailed descriptions

Public members

- **TrackEnd ()**
  Exemplar constructor.
- **virtual TrackEnd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~TrackEnd ()**

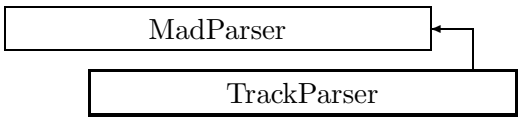


Figure 2.148: Inheritance for class TrackParser

2.200 class TrackParser

The parser class used by the MAD tracking module.
As long as control remains in this class, MAD recognises only the commands allowed in tracking mode.

Type:	Instantiable
Superclasses:	public MadParser
Include file:	./Track/TrackParser.hh

Synopsis (including inherited members)

Public members

- **TrackParser ()**
  (see Section 2.200.1)
- **virtual void parse (Statement&)const**
  Parse and execute current statement. (see Section 2.118.1)
- **virtual Statement* readStatement (TokenStream*)const**
  Read complete statement from a token stream. (see Section 2.118.1)
- **static Token readToken ()**
  Return next input token. (see Section 2.118.1)
- **virtual void run ()const**
  Read current stream. (see Section 2.118.1)
- **virtual void run (TokenStream*)const**
  Read given stream. (see Section 2.118.1)
- **void stop ()const**
  Set stop flag. (see Section 2.118.1)
- **virtual ~TrackParser ()**
  (see Section 2.200.1)

Protected members

- **void execute (Object*,const string&)const**
  Execute or check the current command. (see Section 2.118.1)
- **virtual Object* find (const string&)const**
  Find object by name in the track command directory. (see Section 2.200.1)

- **virtual void parseAction (Statement&)const**
  Parse executable command. (see Section 2.118.1)
- **virtual void parseAssign (Statement&)const**
  Parse assignment statement. (see Section 2.118.1)
- **virtual void parseDefine (Statement&)const**
  Parse definition. (see Section 2.118.1)
- **virtual void parseEnd (Statement&)const**
  Check for end of statement. (see Section 2.118.1)
- **virtual void parseMacro (const string&,Statement&)const**
  Parse macro definition or call. (see Section 2.118.1)
- **virtual void printHelp (const string&)const**
  Print help on named command. (see Section 2.118.1)

2.200.1 Detailed descriptions

Public members

- **TrackParser ()**

- **virtual ~TrackParser ()**

Protected members

- **virtual Object* find (const string&)const**
  Find object by name in the track command directory.

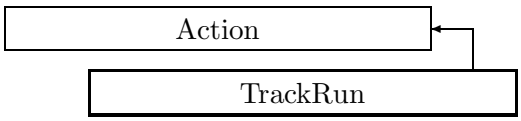


Figure 2.149: Inheritance for class TrackRun

2.201 class TrackRun

The RUN command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Track/TrackRun.hh

Synopsis (including inherited members)

Public members

- **TrackRun ()**
  Exemplar constructor. (see Section 2.201.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual TrackRun* clone (const string&)**
  Make clone. (see Section 2.201.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.201.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~TrackRun ()**
  (see Section 2.201.1)

2.201.1 Detailed descriptions

Public members

- **TrackRun ()**
  Exemplar constructor.
- **virtual TrackRun* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~TrackRun ()**

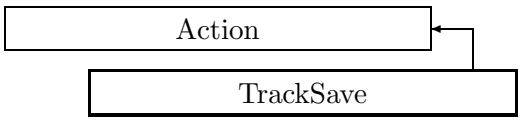


Figure 2.150: Inheritance for class TrackSave

2.202 class TrackSave

The TSAVE command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Track/TrackSave.hh

Synopsis (including inherited members)

Public members

- **TrackSave ()**
  Exemplar constructor. (see Section 2.202.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual TrackSave* clone (const string&)**
  Make clone. (see Section 2.202.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.202.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~TrackSave ()**
  (see Section 2.202.1)

2.202.1 Detailed descriptions

Public members

- **TrackSave ()**
  Exemplar constructor.
- **virtual TrackSave* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~TrackSave ()**

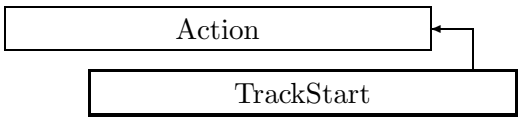


Figure 2.151: Inheritance for class TrackStart

2.203 class TrackStart

The START command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Track/TrackStart.hh

Synopsis (including inherited members)

Public members

- **TrackStart ()**
  Exemplar constructor. (see Section 2.203.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual TrackStart* clone (const string&)**
  Make clone. (see Section 2.203.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.203.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~TrackStart ()**
  (see Section 2.203.1)

2.203.1 Detailed descriptions

Public members

- **TrackStart ()**
  Exemplar constructor.
- **virtual TrackStart* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~TrackStart ()**

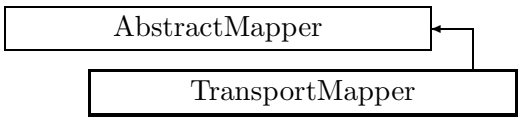


Figure 2.152: Inheritance for class TransportMapper

2.204 class TransportMapper

Build a map using a transport map for each elements.
Phase space coordinates numbering:

number	name	unit
0	x	metres
1	p_x/p_r	1
2	y	metres
3	p_y/p_r	1
4	v*delta_t	metres
5	delta_p/p_r	1

Where p_r is the constant reference momentum defining the reference frame velocity, m is the rest mass of the particles, and v is the instantaneous velocity of the particle.
Other units used:

quantity	unit
reference momentum	electron-volts
velocity	metres/second
accelerating voltage	volts
separator voltage	volts
frequencies	hertz
phase lags	$2 * pi$

Approximations used:

All elements are represented by maps for finite-length elements.

Terms beyond second-order in orbit coordinates are ignored.

Geometric transformations ignore rotations about transverse axes and translations along the design orbit and truncate after second order.

Beam-beam elements are two-dimensional, and the second moment <x,y> of the opposite bunches vanish.

Type:	Instantiable
Superclasses:	public AbstractMapper
Include file:	./Algorithms/TransportMapper.hh

Synopsis (including inherited members)

Public members

- **TransportMapper (const Beamline&,const PartData&,bool,bool)**
  Constructor. (see Section 2.204.1)

- **virtual void getMap (TransportMap<double,6>&)const**
  Return the transport part of the accumulated map. (see Section 2.204.1)
- **virtual void getMap (FVps<double,6>&)const**
  Return the full map accumulated so far. (see Section 2.204.1)
- **virtual void setMap (const TransportMap<double,6>&)**
  Reset the transport part of the accumulated map for restart. (see Section 2.204.1)
- **virtual void setMap (const FVps<double,6>&)**
  Reset the full map for restart. (see Section 2.204.1)
- **virtual void visitAlignWrapper (const AlignWrapper&)**
  Apply the algorithm to an offset beamline object wrapper. (see Section 2.204.1)
- **virtual void visitBeamBeam (const BeamBeam&)**
  Apply the algorithm to a BeamBeam. (see Section 2.204.1)
- **virtual void visitCollimator (const Collimator&)**
  Apply the algorithm to a collimator. (see Section 2.204.1)
- **virtual void visitComponent (const Component&)**
  Apply the algorithm to an arbitrary component. (see Section 2.204.1)
- **virtual void visitCorrector (const Corrector&)**
  Apply the algorithm to a Corrector. (see Section 2.204.1)
- **virtual void visitDiagnostic (const Diagnostic&)**
  Apply the algorithm to a Diagnostic. (see Section 2.204.1)
- **virtual void visitDrift (const Drift&)**
  Apply the algorithm to a Drift. (see Section 2.204.1)
- **virtual void visitLambertson (const Lambertson&)**
  Apply the algorithm to a Lambertson. (see Section 2.204.1)
- **virtual void visitMapIntegrator (const MapIntegrator&)**
  Apply the algorithm to an integrator capable of mapping. (see Section 2.204.1)
- **virtual void visitMarker (const Marker&)**
  Apply the algorithm to a Marker. (see Section 2.204.1)
- **virtual void visitMonitor (const Monitor&)**
  Apply the algorithm to a Monitor. (see Section 2.204.1)
- **virtual void visitMultipole (const Multipole&)**
  Apply the algorithm to a Multipole. (see Section 2.204.1)
- **virtual void visitPatch (const Patch&)**
  Apply the algorithm to a patch. (see Section 2.204.1)
- **virtual void visitRBend (const RBend&)**
  Apply the algorithm to a RBend. (see Section 2.204.1)

- **virtual void visitRFCavity (const RFCavity&)**
  Apply the algorithm to a RFCavity. (see Section 2.204.1)
- **virtual void visitRFQuadrupole (const RFQuadrupole&)**
  Apply the algorithm to a RFQuadrupole. (see Section 2.204.1)
- **virtual void visitSBend (const SBend&)**
  Apply the algorithm to a SBend. (see Section 2.204.1)
- **virtual void visitSeparator (const Separator&)**
  Apply the algorithm to a Separator. (see Section 2.204.1)
- **virtual void visitSeptum (const Septum&)**
  Apply the algorithm to a Septum. (see Section 2.204.1)
- **virtual void visitSolenoid (const Solenoid&)**
  Apply the algorithm to a Solenoid. (see Section 2.204.1)
- **virtual ~TransportMapper ()**
  (see Section 2.204.1)

2.204.1 Detailed descriptions

Public members

- **TransportMapper (const Beamline&,const PartData&,bool,bool)**
  Constructor.

  The beam line to be tracked is **bl**. The particle reference data are taken from **data**. If **revBeam** is true, the beam runs from s = C to s = 0. If **revTrack** is true, we track against the beam.
- **virtual void getMap (TransportMap<double,6>&)const**
  Return the transport part of the accumulated map.
- **virtual void getMap (FVps<double,6>&)const**
  Return the full map accumulated so far.
- **virtual void setMap (const TransportMap<double,6>&)**
  Reset the transport part of the accumulated map for restart.
- **virtual void setMap (const FVps<double,6>&)**
  Reset the full map for restart.
- **virtual void visitAlignWrapper (const AlignWrapper&)**
  Apply the algorithm to an offset beamline object wrapper.
- **virtual void visitBeamBeam (const BeamBeam&)**
  Apply the algorithm to a BeamBeam.
- **virtual void visitCollimator (const Collimator&)**
  Apply the algorithm to a collimator.

- **virtual void visitComponent (const Component&)**
  Apply the algorithm to an arbitrary component.

  This override calls the component to track the map.

- **virtual void visitCorrector (const Corrector&)**
  Apply the algorithm to a Corrector.

- **virtual void visitDiagnostic (const Diagnostic&)**
  Apply the algorithm to a Diagnostic.

- **virtual void visitDrift (const Drift&)**
  Apply the algorithm to a Drift.

- **virtual void visitLambertson (const Lambertson&)**
  Apply the algorithm to a Lambertson.

- **virtual void visitMapIntegrator (const MapIntegrator&)**
  Apply the algorithm to an integrator capable of mapping.

- **virtual void visitMarker (const Marker&)**
  Apply the algorithm to a Marker.

- **virtual void visitMonitor (const Monitor&)**
  Apply the algorithm to a Monitor.

- **virtual void visitMultipole (const Multipole&)**
  Apply the algorithm to a Multipole.

- **virtual void visitPatch (const Patch&)**
  Apply the algorithm to a patch.

- **virtual void visitRBend (const RBend&)**
  Apply the algorithm to a RBend.

- **virtual void visitRFCavity (const RFCavity&)**
  Apply the algorithm to a RFCavity.

- **virtual void visitRFQuadrupole (const RFQuadrupole&)**
  Apply the algorithm to a RFQuadrupole.

- **virtual void visitSBend (const SBend&)**
  Apply the algorithm to a SBend.

- **virtual void visitSeparator (const Separator&)**
  Apply the algorithm to a Separator.

- **virtual void visitSeptum (const Septum&)**
  Apply the algorithm to a Septum.

- **virtual void visitSolenoid (const Solenoid&)**
  Apply the algorithm to a Solenoid.

- **virtual ~TransportMapper ()**

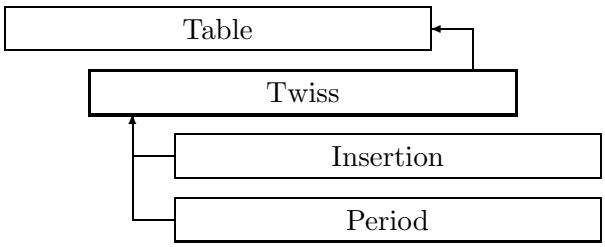


Figure 2.153: Inheritance for class Twiss

2.205 class Twiss

Class Twiss Abstract base class for table buffers holding lattice function.

Type:	abstract
Superclasses:	public Table
Include file:	./Tables/Twiss.hh

Synopsis (including inherited members)

Public members

- **struct Cell**
  Descriptor for printing a table cell. (see Section 2.187.1)
- **typedef std::vector<Cell> CellArray**
  An array of cell descriptors. (see Section 2.187.1)
- **class Row**
  Structure for a row of the Twiss table. (see Section 2.205.1)
- **typedef TBeamline<Row> TLine**
  (see Section 2.205.1)
- **TLine::const_iterator begin ()const**
  Access to first row. (see Section 2.205.1)
- **TLine::iterator begin ()**
  Access to first row. (see Section 2.205.1)
- **virtual bool canReplaceBy (Object*)**
  Test if object can be replaced. (see Section 2.187.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)

- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write the table to the DOOM data base. (see Section 2.205.1)
- **virtual void doomSummary (DoomWriter&)const**
  Fill in summary record. (see Section 2.205.1)
- **TLine::const_iterator end ()const**
  Access to last row. (see Section 2.205.1)
- **TLine::iterator end ()**
  Access to last row. (see Section 2.205.1)
- **virtual void execute ()**
  Check validity of the table definition. (see Section 2.205.1)
- **virtual void fill ()**
  Refill the buffer. (see Section 2.187.1)
- **static Table* find (const string&)**
  Find named Table. (see Section 2.187.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **double getALFi (const Row&,int,int)const**
  (see Section 2.205.1)
- **double getALFik (const Row&,int,int)const**
  Mais-Ripken alpha functions. (see Section 2.205.1)
- **double getBETi (const Row&,int,int)const**
  (see Section 2.205.1)
- **double getBETik (const Row&,int,int)const**
  Mais-Ripken beta functions. (see Section 2.205.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **double getCO (const Row&,int,int)const**
  Closed orbit. (see Section 2.205.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.187.1)
- **virtual double getCell (const PlaceRep&,const string&)**
  Return a selected value in a selected row. (see Section 2.205.1)

- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column **col** of this table, limited by **range**. (see Section 2.205.1)
- **FMatrix<double,6,6> getCurlyA ()const**
  Return initial curly A matrix. (see Section 2.205.1)
- **FMatrix<double,6,6> getCurlyA (const Row&)const**
  Curly A map for given row. (see Section 2.205.1)
- **const Row& getCurrent ()const**
  Return current table row in iteration. (see Section 2.205.1)
- **virtual CellArray getDefault ()const**
  Return the default print columns. (see Section 2.205.1)
- **double getDisp (const Row&,int,int)const**
  Dispersion. (see Section 2.205.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **double getET ()const**
  Return emittance for mode 3. (see Section 2.205.1)
- **double getEX ()const**
  Return emittance for mode 1. (see Section 2.205.1)
- **double getEY ()const**
  Return emittance for mode 2. (see Section 2.205.1)
- **double getEigen (const Row&,int,int)const**
  Eigenvectors. (see Section 2.205.1)
- **double getGAMik (const Row&,int,int)const**
  Mais-Ripken gamma functions. (see Section 2.205.1)
- **virtual double getLength ()**
  Return the length of the table. (see Section 2.205.1)
- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline. (see Section 2.205.1)
- **double getMUi (const Row&,int,int)const**
  Three modes, "naive" Twiss functions. (see Section 2.205.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **FMatrix<double,6,6> getMatrix (const Row&)const**
  Accumulated transfer map. (see Section 2.205.1)
- **double getMatrix (const Row&,int,int)const**
  Transfer matrix. (see Section 2.205.1)

- **FVector<double,6> getOrbit ()const**
  Return initial closed orbit. (see Section 2.205.1)
- **FVector<double,6> getOrbit (const Row&)const**
  Get orbit in given row. (see Section 2.205.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row, possible user-defined. (see Section 2.205.1)
- **double getS (const Row&,int,int)const**
  Arc length for given row. (see Section 2.205.1)
- **FMatrix<double,6,6> getSigma ()const**
  Initial envelope (Sigma) matrix. (see Section 2.205.1)
- **FMatrix<double,6,6> getSigma (const Row&)const**
  Envelope (Sigma) matrix for given row. (see Section 2.205.1)
- **double getSigma (const Row&,int,int)const**
  Sigma matrix. (see Section 2.205.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **virtual void invalidate ()**
  Mark this table as invalid, if it is dynamic. (see Section 2.187.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **virtual bool isDependent (const string&)const**
  Check dependency. (see Section 2.205.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return column expression. (see Section 2.205.1)

- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual void makeTFS (std::ostream&,const CellArray&)const**
  Write TFS file for this table. (see Section 2.187.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual bool matches (Table*)const**
  Check compatibility. (see Section 2.205.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **virtual void printTable (std::ostream&,const CellArray&)const**
  Print list for the table. (see Section 2.187.1)
- **void printTableBody (std::ostream&,const CellArray&)const**
  Print the body to this TWISS table. (see Section 2.205.1)
- **void printTableTitle (std::ostream&,const char*)const**
  Print standard information about the TWISS table. (see Section 2.205.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)

- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.187.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.187.1)
- **void tfsBody (std::ostream&,const CellArray&)const**
  Write TFS body for this table. (see Section 2.205.1)
- **void tfsSummary (std::ostream&)const**
  List TFS descriptors for table summary. (see Section 2.205.1)
- **void tfsTableDescriptors (std::ostream&)const**
  Write TFS descriptors existing for all tables. (see Section 2.187.1)
- **virtual void tfsTwissDescriptors (std::ostream&)const**
  Write TFS descriptors for a TWISS table. (see Section 2.205.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Twiss ()**
  (see Section 2.205.1)

Protected members

- **Twiss (int,const char*,const char*)**
  The common attributes for all objects having the "TWISS" interface. Exemplar constructor. (see Section 2.205.1)
- **Twiss (const string&,Twiss*)**
  Clone constructor. (see Section 2.205.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **FMatrix<double,6,6> curly_A**
  The initial curly A matrix. (see Section 2.205.1)
- **bool dynamic**
  Flag dynamic table. (see Section 2.187.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)

- **bool modified**
  Dirty flag. (see Section 2.143.1)
- **static const int numColumns**
  Number of table columns. (see Section 2.205.1)
- **FVector<double,6> orbit**
  The initial closed orbit. (see Section 2.205.1)
- **bool refill**
  Refill flag. (see Section 2.187.1)

2.205.1 Detailed descriptions

Public members

- **class Row**
  Structure for a row of the Twiss table.
- **typedef TBeamline<Row> TLine**
- **TLine::const_iterator begin ()const**
  Access to first row.

  Version for constant table.
- **TLine::iterator begin ()**
  Access to first row.

  Version for non-constant table.
- **virtual void doomPut (DoomWriter&)const**
  Write the table to the DOOM data base.
- **virtual void doomSummary (DoomWriter&)const**
  Fill in summary record.
- **TLine::const_iterator end ()const**
  Access to last row.

  Version for constant table.
- **TLine::iterator end ()**
  Access to last row.

  Version for non-constant table.
- **virtual void execute ()**
  Check validity of the table definition.
- **double getALFi (const Row&,int,int)const**

- **double getALFik (const Row&,int,int)const**
  Mais-Ripken alpha functions.

  First index (0 ... 2) is plane, second index (0 ... 2) is mode.

- **double getBETi (const Row&,int,int)const**

- **double getBETik (const Row&,int,int)const**
  Mais-Ripken beta functions.

  First index (0 ... 2) is plane, second index (0 ... 2) is mode.

- **double getCO (const Row&,int,int)const**
  Closed orbit.

  Index (0 ... 5) is plane.

- **virtual double getCell (const PlaceRep&,const string&)**
  Return a selected value in a selected row.

- **virtual std::vector<double> getColumn (const RangeRep&,const string&)**
  Return column col of this table, limited by range.

- **FMatrix<double,6,6> getCurlyA ()const**
  Return initial curly A matrix.

- **FMatrix<double,6,6> getCurlyA (const Row&)const**
  Curly A map for given row.

- **const Row& getCurrent ()const**
  Return current table row in iteration.

- **virtual CellArray getDefault ()const**
  Return the default print columns.

- **double getDisp (const Row&,int,int)const**
  Dispersion.

  Index (0 ... 5) is plane.

- **double getET ()const**
  Return emittance for mode 3.

- **double getEX ()const**
  Return emittance for mode 1.

- **double getEY ()const**
  Return emittance for mode 2.

- **double getEigen (const Row&,int,int)const**
  Eigenvectors.

  First index (0 ... 5) is plane, second index (0 ... 5) is column.

- **double getGAMik (const Row&,int,int)const**
  Mais-Ripken gamma functions.

  First index (0 ... 2) is plane, second index (0 ... 2) is mode.

- **virtual double getLength ()**
  Return the length of the table.

- **virtual const Beamline* getLine ()const**
  Return embedded CLASSIC beamline.

- **double getMUi (const Row&,int,int)const**
  Three modes, "naive" Twiss functions.

  Index (0 ... 2) is mode.

- **FMatrix<double,6,6> getMatrix (const Row&)const**
  Accumulated transfer map.

- **double getMatrix (const Row&,int,int)const**
  Transfer matrix.

  First index (0 ... 5) is row, second index (0 ... 5) is column.

- **FVector<double,6> getOrbit ()const**
  Return initial closed orbit.

- **FVector<double,6> getOrbit (const Row&)const**
  Get orbit in given row.

- **virtual std::vector<double> getRow (const PlaceRep&,const std::vector<string>&)**
  Return a table row, possible user-defined.

- **double getS (const Row&,int,int)const**
  Arc length for given row.

- **FMatrix<double,6,6> getSigma ()const**
  Initial envelope (Sigma) matrix.

- **FMatrix<double,6,6> getSigma (const Row&)const**
  Envelope (Sigma) matrix for given row.

- **double getSigma (const Row&,int,int)const**
  Sigma matrix.

  Both indices (0 ... 5) refer to planes.

- **virtual bool isDependent (const string&)const**
  Check dependency.

  Return true, if this table depends on the named object.

- **virtual Expressions::PtrToScalar<double> makeColumnExpression (const string&)const**
  Return column expression.

  Return an expression which denotes a column of the twiss table, identified by its name.

- **virtual bool matches (Table*)const**
  Check compatibility.

  True, if **rhs** is derived from **Twiss**.

- **void printTableBody (std::ostream&,const CellArray&)const**
  Print the body to this TWISS table.

- **void printTableTitle (std::ostream&,const char*)const**
  Print standard information about the TWISS table.

- **void tfsBody (std::ostream&,const CellArray&)const**
  Write TFS body for this table.

- **void tfsSummary (std::ostream&)const**
  List TFS descriptors for table summary.

  Writes the maximum and r.m.s. orbit and dispersion.

- **virtual void tfsTwissDescriptors (std::ostream&)const**
  Write TFS descriptors for a TWISS table.

  Writes the line and beam info, as well as length and tunes.

- **virtual ~Twiss ()**

Protected members

- **Twiss (int,const char*,const char*)**
  The common attributes for all objects having the "TWISS" interface. Exemplar constructor.

  Must be accessible to classes Insertion and Period.

- **Twiss (const string&,Twiss*)**
  Clone constructor.

- **FMatrix<double,6,6> curly_A**
  The initial curly A matrix.

- **static const int numColumns**
  Number of table columns.

- **FVector<double,6> orbit**
  The initial closed orbit.

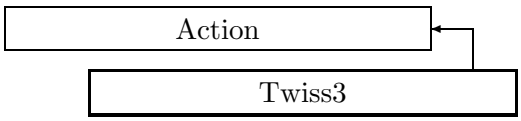


Figure 2.154: Inheritance for class Twiss3

2.206 class Twiss3

The TWISS3 command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Tables/Twiss3.hh

Synopsis (including inherited members)

Public members

- **Twiss3 ()**
  Exemplar constructor. (see Section 2.206.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Twiss3* clone (const string&)**
  Make clone. (see Section 2.206.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.206.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Twiss3 ()**
  (see Section 2.206.1)

2.206.1 Detailed descriptions

Public members

- **Twiss3 ()**
  Exemplar constructor.
- **virtual Twiss3* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~Twiss3 ()**

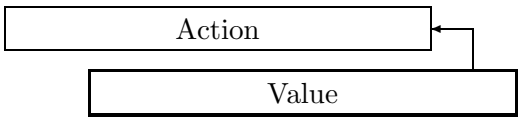


Figure 2.155: Inheritance for class Value

2.207 class Value

The VALUE command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/Value.hh

Synopsis (including inherited members)

Public members

- **Value ()**
  Exemplar constructor. (see Section 2.207.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Value* clone (const string&)**
  Make clone. (see Section 2.207.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.207.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.207.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~Value ()**
  (see Section 2.207.1)

2.207.1 Detailed descriptions

Public members

- **Value ()**
  Exemplar constructor.
- **virtual Value* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~Value ()**

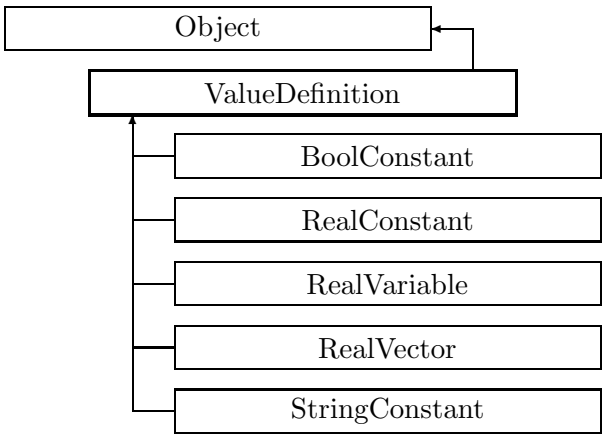


Figure 2.156: Inheritance for class ValueDefinition

2.208 class ValueDefinition

The base class for all MAD value definitions.
It implements the common behaviour of values, it can also be used via dynamic casting to determine whether an object represents a value.

Type:	abstract
Superclasses:	public Object
Include file:	./AbstractObjects/ValueDefinition.hh

Synopsis (including inherited members)

Public members

- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.143.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual Object* clone (const string&)**
  Return a clone. (see Section 2.143.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.143.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)

- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual bool getBool ()const**
  Return logical value. (see Section 2.208.1)
- **virtual bool getBoolComponent (int)const**
  Return indexed logical value. (see Section 2.208.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.208.1)
- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **virtual double getReal ()const**
  Return real value. (see Section 2.208.1)
- **virtual double getRealComponent (int)const**
  Return indexed real value. (see Section 2.208.1)
- **virtual string getString ()const**
  Return string value. (see Section 2.208.1)
- **virtual string getStringComponent (int)const**
  Return indexed string value. (see Section 2.208.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)

- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)
- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.208.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.208.1)

- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **Attribute& value ()**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **const Attribute& value ()const**
  Return the attribute representing the value of the definition. (see Section 2.208.1)
- **virtual ~ValueDefinition ()**
  (see Section 2.208.1)

Protected members

- **ValueDefinition (int,const char*,const char*)**
  Constructor for exemplars. (see Section 2.208.1)
- **ValueDefinition (const string&,ValueDefinition*)**
  Constructor for clones. (see Section 2.208.1)
- **bool builtin**
  Built-in flag. (see Section 2.143.1)
- **bool flagged**
  Object flag. (see Section 2.143.1)
- **const string itsPrefix**
  The declaration prefix. (see Section 2.208.1)
- **bool modified**
  Dirty flag. (see Section 2.143.1)

2.208.1 Detailed descriptions

Public members

- **virtual bool getBool ()const**
  Return logical value.

  The default version throws MadException.
- **virtual bool getBoolComponent (int)const**
  Return indexed logical value.

  The default version throws MadException.
- **virtual const string getCategory ()const**
  Return the object category as a string.

  Return the string "VARIABLE".

- **virtual double getReal ()const**
  Return real value.

  The default version throws MadException.

- **virtual double getRealComponent (int)const**
  Return indexed real value.

  The default version throws MadException.

- **virtual string getString ()const**
  Return string value.

  The default version throws MadException.

- **virtual string getStringComponent (int)const**
  Return indexed string value.

  The default version throws MadException.

- **virtual bool shouldTrace ()const**
  Trace flag.

  If true, the object's execute() function should be traced. Always false for value definitions.

- **virtual bool shouldUpdate ()const**
  Update flag.

  If true, the data structure should be updated before calling execute(). Always false for value definitions.

- **Attribute& value ()**
  Return the attribute representing the value of the definition.

  Version for non-constant object.

- **const Attribute& value ()const**
  Return the attribute representing the value of the definition.

  Version for constant object.

- **virtual ~ValueDefinition ()**

Protected members

- **ValueDefinition (int,const char*,const char*)**
  Constructor for exemplars.

- **ValueDefinition (const string&,ValueDefinition*)**
  Constructor for clones.

- **const string itsPrefix**
  The declaration prefix.

  A string representing the value type in MAD-9 input format: "BOOL", "REAL", "REAL CONST", "REAL VECTOR", etc.

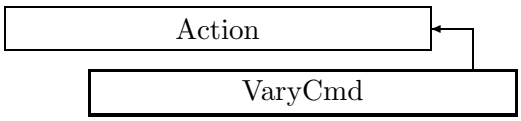


Figure 2.157: Inheritance for class VaryCmd

2.209 class VaryCmd

The VARY command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./Match/VaryCmd.hh

Synopsis (including inherited members)

Public members

- **VaryCmd ()**
  Exemplar constructor. (see Section 2.209.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual VaryCmd* clone (const string&)**
  Make clone. (see Section 2.209.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.209.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse the object. (see Section 2.143.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~VaryCmd ()**
  (see Section 2.209.1)

2.209.1 Detailed descriptions

Public members

- **VaryCmd ()**
  Exemplar constructor.
- **virtual VaryCmd* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual ~VaryCmd ()**

2.210 template class Vector <class>

Type:	Instantiable

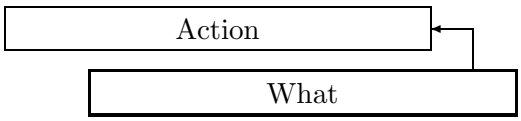


Figure 2.158: Inheritance for class What

2.211 class What

The WHAT command.

Type:	Instantiable
Superclasses:	public Action
Include file:	./BasicActions/What.hh

Synopsis (including inherited members)

Public members

- **What ()**
  Exemplar constructor. (see Section 2.211.1)
- **virtual bool canReplaceBy (Object*)**
  Test if replacement is allowed. (see Section 2.8.1)
- **void clear ()**
  Clear the occurrence counter. (see Section 2.143.1)
- **virtual What* clone (const string&)**
  Make clone. (see Section 2.211.1)
- **void copyAttributes (const Object&)**
  Copy attributes from another object. (see Section 2.143.1)
- **virtual void doomGet (const DoomReader&)**
  Read an Object from the DOOM data base. (see Section 2.143.1)
- **virtual void doomPut (DoomWriter&)const**
  Write an Object to the DOOM data base. (see Section 2.143.1)
- **virtual void execute ()**
  Execute the command. (see Section 2.211.1)
- **virtual Attribute* findAttribute (const string&)**
  Find an attribute by name. (see Section 2.143.1)
- **virtual const Attribute* findAttribute (const string&)const**
  Find an attribute by name. (see Section 2.143.1)
- **const Object* getBaseObject ()const**
  Return the object's base type object. (see Section 2.143.1)
- **virtual const string getCategory ()const**
  Return the object category as a string. (see Section 2.8.1)

- **double getDoomTime ()const**
  Return the time stamp stored by setDoomTime(). (see Section 2.143.1)
- **const string& getMadName ()const**
  Return object name. (see Section 2.143.1)
- **Object* getParent ()const**
  Return parent pointer. (see Section 2.143.1)
- **int increment ()**
  Increment and return the occurrence counter. (see Section 2.143.1)
- **bool isBuiltin ()const**
  True, if **this** is a built-in object. (see Section 2.143.1)
- **bool isDirty ()const**
  True, if the **modified** flag is set. (see Section 2.143.1)
- **bool isFlagged ()const**
  True, if **this** is flagged by setFlag(true). (see Section 2.143.1)
- **virtual bool isShared ()const**
  Shared flag. (see Section 2.143.1)
- **bool isTreeMember (const Object*)const**
  Test for tree membership. (see Section 2.143.1)
- **std::vector<Attribute> itsAttr**
  The object attributes (see Attribute.hh). (see Section 2.143.1)
- **virtual Object* makeInstance (const string&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **virtual Object* makeTemplate (const string&,TokenStream&,Statement&)**
  Macro handler function. (see Section 2.143.1)
- **int occurrenceCount ()**
  Return the occurrence counter. (see Section 2.143.1)
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command). (see Section 2.211.1)
- **virtual void parseShortcut (Statement&)**
  Parser for single-attribute commands. (see Section 2.143.1)
- **virtual void print (std::ostream&)const**
  Print the object. (see Section 2.143.1)
- **virtual void printHelp (std::ostream&)const**
  Print help. (see Section 2.143.1)
- **void registerReference (AttributeBase*)**
  Register a reference to this object. (see Section 2.143.1)

- **virtual void replace (Object*,Object*)**
  Replace references. (see Section 2.143.1)
- **void setDirty (bool)**
  Set/reset the **modified** flag. (see Section 2.143.1)
- **void setDoomTime (double)**
  Store the time stamp read from the DOOM data base. (see Section 2.143.1)
- **void setFlag (bool)**
  Flag/unflag this object, e. g. to control output of objects for (see Section 2.143.1)
- **void setMadName (const string&)**
  Set object name. (see Section 2.143.1)
- **void setParent (Object*)**
  Set parent object. (see Section 2.143.1)
- **virtual void setShared (bool)**
  Set/reset shared flag. (see Section 2.143.1)
- **virtual bool shouldTrace ()const**
  Trace flag. (see Section 2.8.1)
- **virtual bool shouldUpdate ()const**
  Update flag. (see Section 2.8.1)
- **void unregisterReference (AttributeBase*)**
  Unegister a reference to this object. (see Section 2.143.1)
- **virtual void update ()**
  Update this object. (see Section 2.143.1)
- **virtual ~What ()**
  (see Section 2.211.1)

2.211.1 Detailed descriptions

Public members

- **What ()**
  Exemplar constructor.
- **virtual What* clone (const string&)**
  Make clone.
- **virtual void execute ()**
  Execute the command.
- **virtual void parse (Statement&)**
  Parse command (special for one-attribute command).
- **virtual ~What ()**

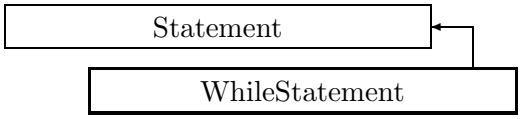


Figure 2.159: Inheritance for class WhileStatement

2.212 class WhileStatement

While statement.
A statement of the form "WHILE (<condition>) <statement>".

Type:	Instantiable
Superclasses:	public Statement
Include file:	./MadParser/WhileStatement.hh

Synopsis (including inherited members)

Public members

- **WhileStatement (const Parser&,TokenStream&)**
  Constructor. (see Section 2.212.1)
- **virtual void execute (const Parser&)**
  Execute. (see Section 2.212.1)
- **virtual ~WhileStatement ()**
  (see Section 2.212.1)

2.212.1 Detailed descriptions

Public members

- **WhileStatement (const Parser&,TokenStream&)**
  Constructor.

  Parse the statement on the given token stream, using the given parser.
- **virtual void execute (const Parser&)**
  Execute.

  Use the given parser to execute the controlled statements.
- **virtual ~WhileStatement ()**

Bibliography

[1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.