ReGenESyS - Reborned GENeric and Expansible SYstem Simulator

2019.0601

Generated by Doxygen 1.8.11

Contents

Chapter 1

ReGenESyS

Reborn Generic and Expansible System Simulator

Genesys is a result of teaching and research activities of Professor Dr. Ing Rafael Luiz Cancian. It began in early 2002 as a way to teach students the basics and simulation techniques of systems implemented by other comercial simulation tools, such as Arena. In Genesys development he replicated all the SIMAN language, used by Arena software, and Genesys has become a clone of that tool, including its graphical interface. Genesys allowed the inclusion of new simulation components through dynamic link libraries and also the parallel execution of simulation models in a distributed environment. The development of Genesys continued until 2007, when the professor stopped teaching systems simulation classes. Ten years later the professor starts again to teach systems simulation classes and to carry out scientific research in the area. So in 2018 Genesys is reborn as ReGenesys, with new language and programming techniques, and even more ambitious goals.

Developed by `rlcancian`

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

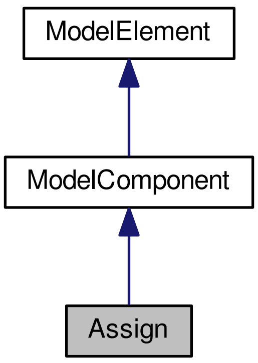
Chapter 5

Class Documentation

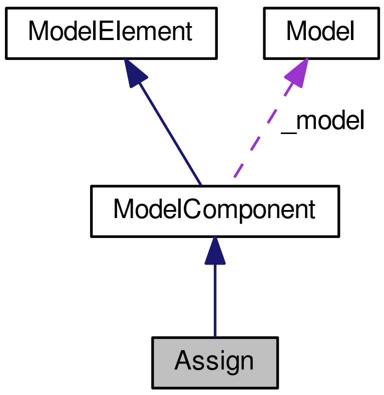
5.1 Assign Class Reference

```
#include <Assign.h>
```

Inheritance diagram for Assign:



Collaboration diagram for Assign:



Classes

- class Assignment

Public Types

- enum DestinationType : int { DestinationType::Attribute =0, DestinationType::Variable =1 }

Public Member Functions

- Assign (Model ∗model)
- Assign (const Assign &orig)
- virtual ∼Assign ()
- virtual std::string show ()
- List< Assignment ∗ > ∗ getAssignments () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.1.1 Member Enumeration Documentation

5.1.1.1 enum **Assign::DestinationType** : int [strong]

Enumerator

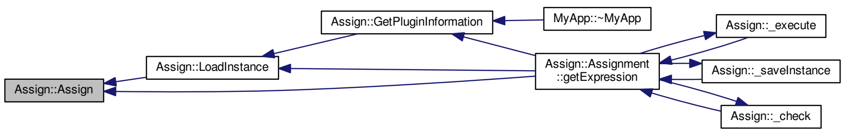
Attribute

Variable

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Assign::Assign (**Model** ∗ *model*)

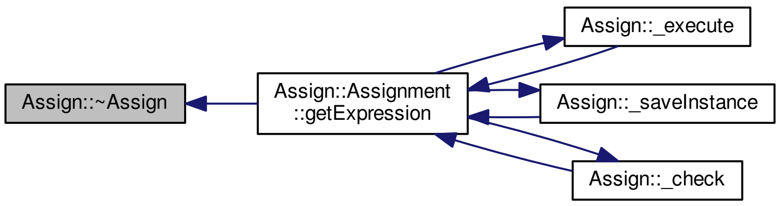
Here is the caller graph for this function:



5.1.2.2 Assign::Assign (const **Assign** & *orig*)

5.1.2.3 Assign::∼Assign ( ) [virtual]

Here is the caller graph for this function:

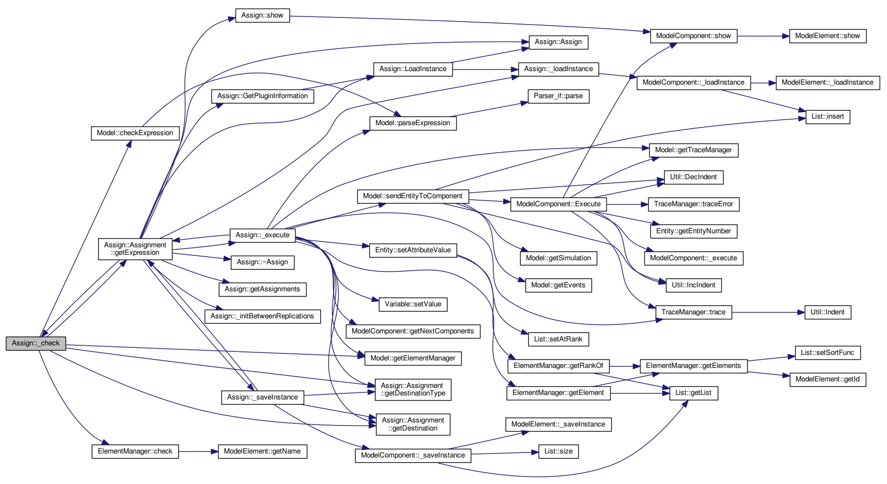


5.1.3 Member Function Documentation

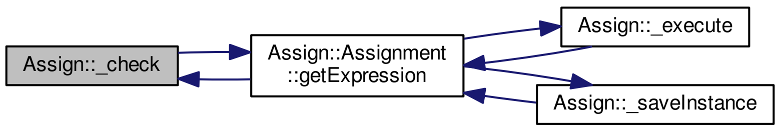
5.1.3.1 bool Assign::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



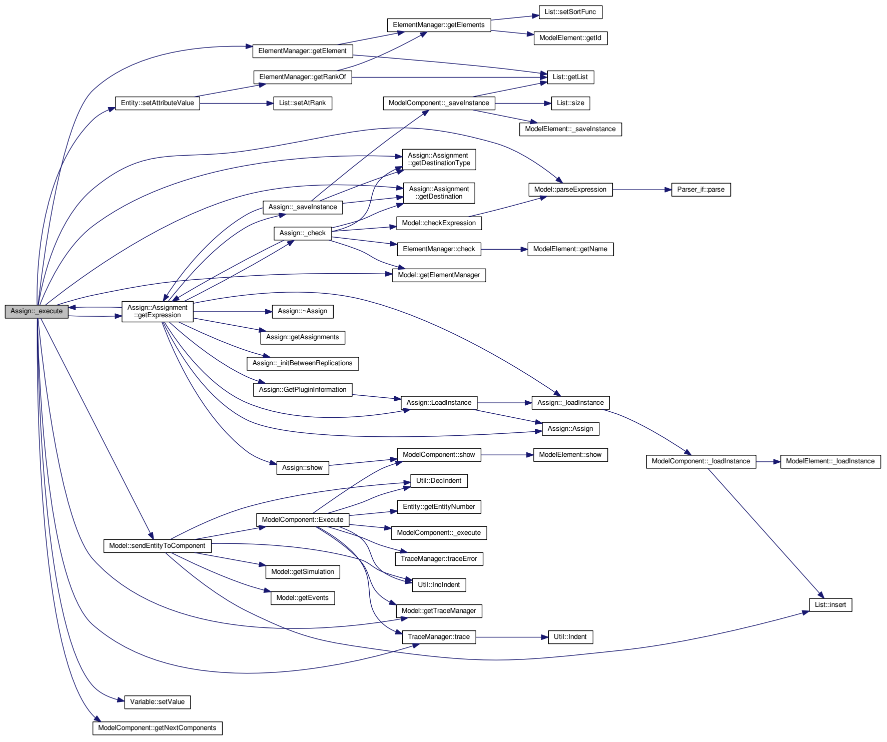
Here is the caller graph for this function:



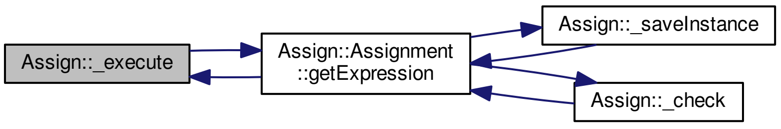
5.1.3.2 void Assign::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

Here is the call graph for this function:



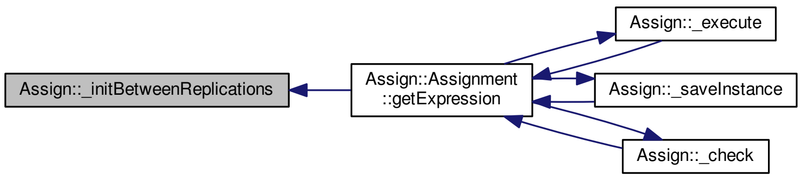
Here is the caller graph for this function:



5.1.3.3 void Assign::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

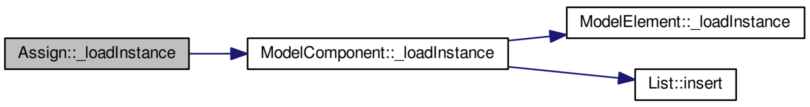
Here is the caller graph for this function:


Assign::_initBetweenReplications
Assign::Assignment ::getExpression
Assign::_execute
Assign::_saveInstance
Assign::_check

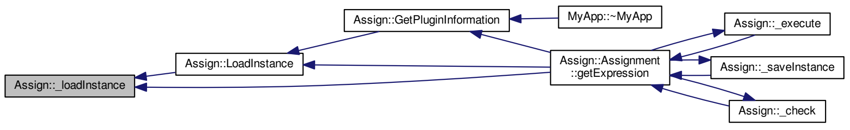
5.1.3.4 bool Assign::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



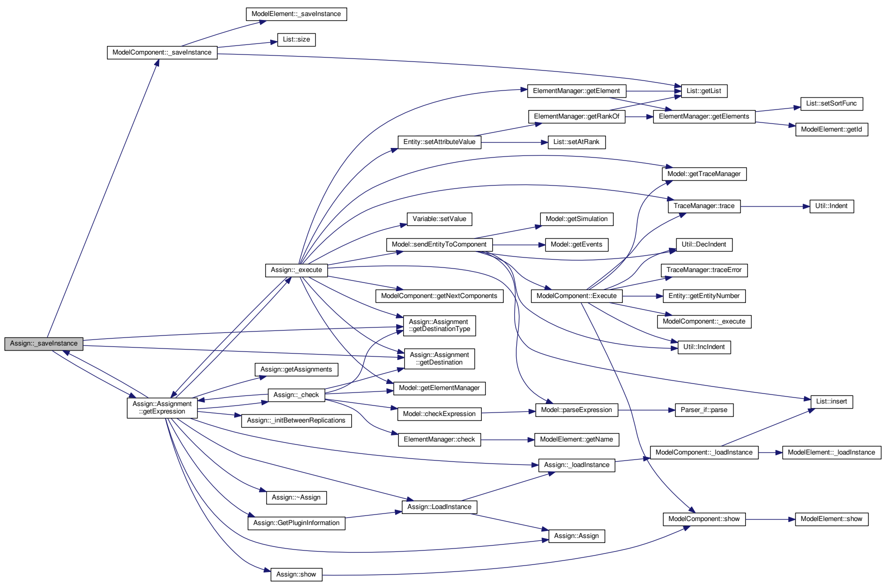
Here is the caller graph for this function:



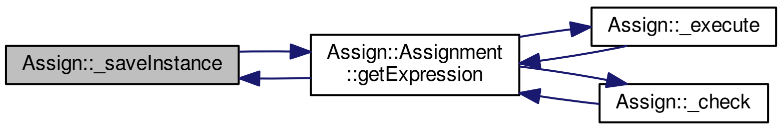
5.1.3.5 std::map< std::string, std::string > * Assign::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:

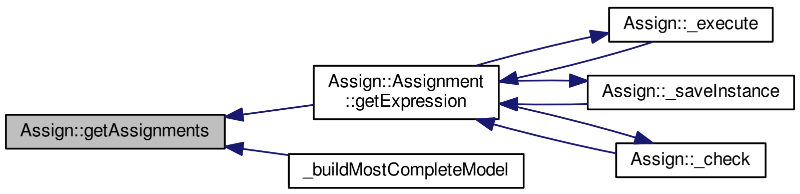


Here is the caller graph for this function:



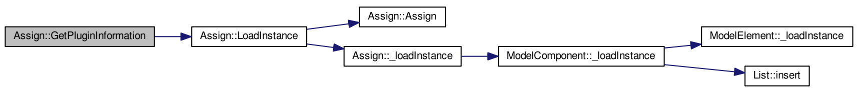
5.1.3.6 List< Assign::Assignment * > * Assign::getAssignments ( ) const

Here is the caller graph for this function:

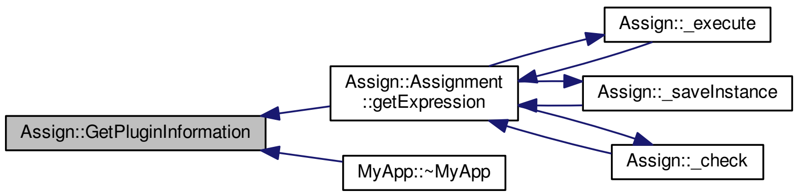


5.1.3.7 PluginInformation * Assign::GetPluginInformation ( ) [static]

Here is the call graph for this function:

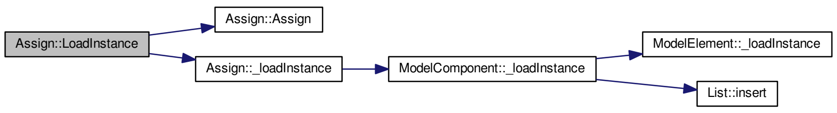

Assign::GetPluginInformation
Assign::LoadInstance
Assign::Assign
Assign::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
List::insert

Here is the caller graph for this function:

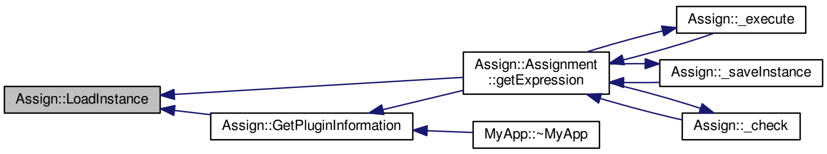


5.1.3.8 ModelComponent ∗ Assign::LoadInstance (Model ∗ *model,* std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:



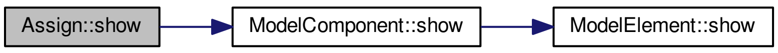
Here is the caller graph for this function:



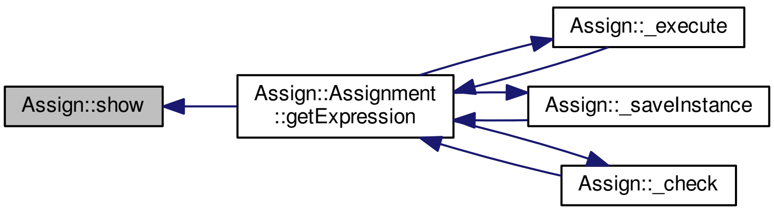
5.1.3.9 std::string Assign::show ( ) [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



Here is the caller graph for this function:



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

- Assign.h
- Assign.cpp

5.2 Assign::Assignment Class Reference

```
#include <Assign.h>
```

Public Member Functions

- Assignment (DestinationType destinationType, std::string destination, std::string expression)
- void setDestination (std::string _destination)
- std::string getDestination () const
- void setDestinationType (DestinationType _destinationType)
- DestinationType getDestinationType () const
- void setExpression (std::string _expression)
- std::string getExpression () const

5.2.1 Detailed Description

While the assign class allows you to perform multiple assignments, the assignment class defines an assignment itself.

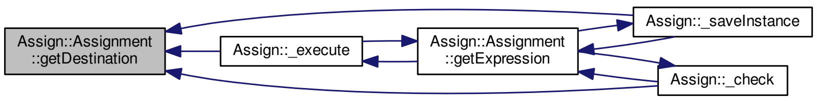
5.2.2 Constructor & Destructor Documentation

5.2.2.1 Assign::Assignment::Assignment (DestinationType *destinationType,* std::string *destination,* std::string *expression*) `[inline]`

5.2.3 Member Function Documentation

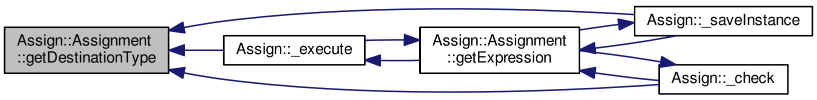
5.2.3.1 std::string Assign::Assignment::getDestination ( ) const `[inline]`

Here is the caller graph for this function:


Assign::Assignment ::getDestination
Assign::_execute
Assign::Assignment ::getExpression
Assign::_saveInstance
Assign::_check

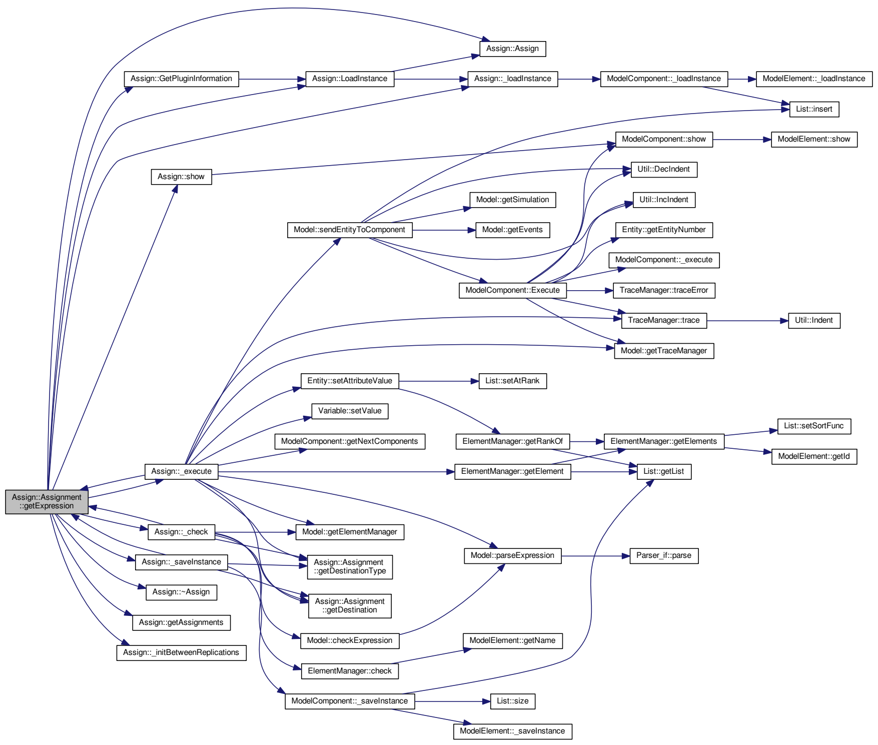
5.2.3.2 DestinationType Assign::Assignment::getDestinationType ( ) const [inline]

Here is the caller graph for this function:

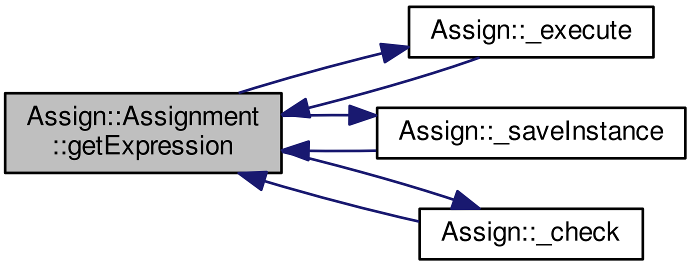


5.2.3.3 std::string Assign::Assignment::getExpression ( ) const [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



5.2.3.4 void Assign::Assignment::setDestination (std::string *_destination*) `[inline]`

5.2.3.5 void Assign::Assignment::setDestinationType (DestinationType *_destinationType*) `[inline]`

5.2.3.6 void Assign::Assignment::setExpression (std::string *_expression*) `[inline]`

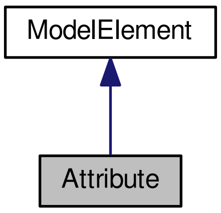
The documentation for this class was generated from the following file:

- Assign.h

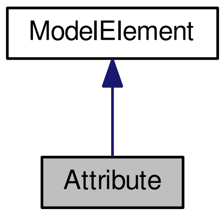
5.3 Attribute Class Reference

```
#include <Attribute.h>
```

Inheritance diagram for Attribute:



Collaboration diagram for Attribute:



Public Member Functions

- Attribute ()
- Attribute (std::string name)
- Attribute (const Attribute &orig)
- virtual ∼Attribute ()
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

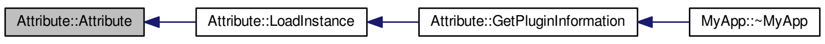
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.3.1 Constructor & Destructor Documentation

5.3.1.1 Attribute::Attribute ( )

Here is the caller graph for this function:



5.3.1.2 Attribute::Attribute (std::string *name*)

5.3.1.3 Attribute::Attribute (const Attribute & *orig*)

5.3.1.4 Attribute::∼Attribute ( ) [virtual]

5.3.2 Member Function Documentation

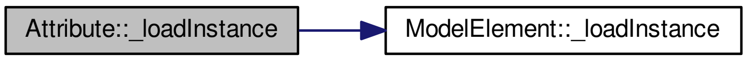
5.3.2.1 bool Attribute::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

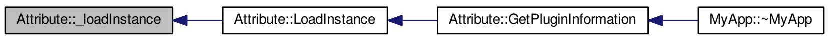
5.3.2.2 bool Attribute::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



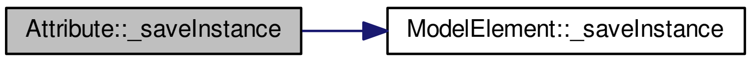
Here is the caller graph for this function:



5.3.2.3 std::map< std::string, std::string > ∗ Attribute::_saveInstance ( ) [protected], [virtual]

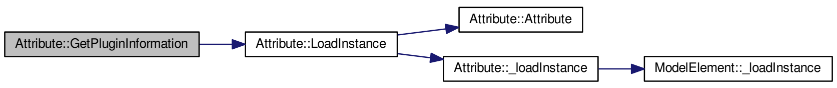
Reimplemented from ModelElement.

Here is the call graph for this function:

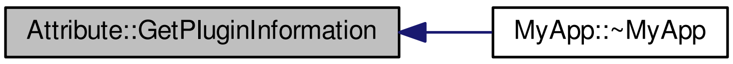


5.3.2.4 PluginInformation ∗ Attribute::GetPluginInformation ( ) [static]

Here is the call graph for this function:

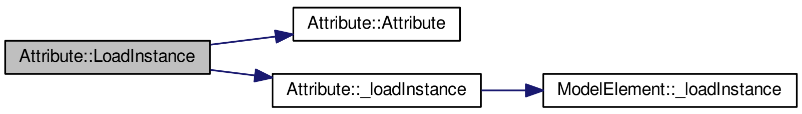


Here is the caller graph for this function:

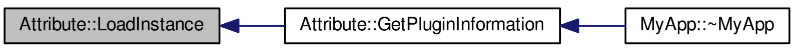


5.3.2.5 ModelElement ∗ Attribute::LoadInstance (ElementManager ∗ *elems*, std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:



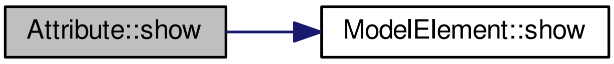
Here is the caller graph for this function:



5.3.2.6 std::string Attribute::show ( ) [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



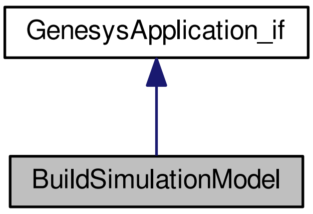
The documentation for this class was generated from the following files:

- Attribute.h
- Attribute.cpp

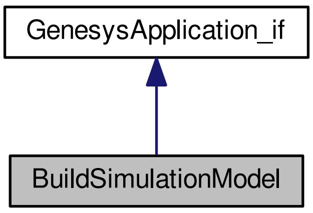
5.4 BuildSimulationModel Class Reference

`#include <BuildSimulationModel.h>`

Inheritance diagram for BuildSimulationModel:



Collaboration diagram for BuildSimulationModel:



Public Member Functions

- BuildSimulationModel ()
- virtual int main (int argc, char ∗∗argv)

5.4.1 Constructor & Destructor Documentation

5.4.1.1 BuildSimulationModel::BuildSimulationModel ( )

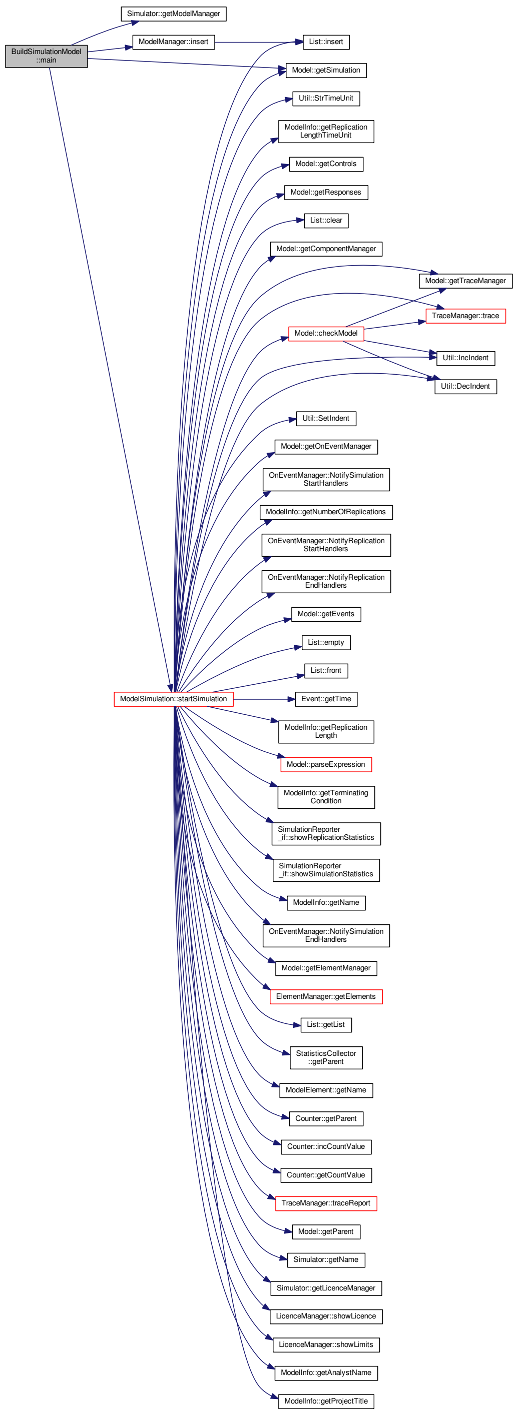
5.4.2 Member Function Documentation

5.4.2.1 int BuildSimulationModel::main (int *argc*, char ∗∗ *argv*) [virtual]

This is the main function of the BuildSimulationModel application. It instanciates the simulator, builds a simulation model and then simulate that model.

Implements GenesysApplication_if.

Here is the call graph for this function:


Simulator::getModelManager
ModelManager::insert
BuildSimulationModel ::main
List::insert
Model::getSimulation
Util::StrTimeUnit
ModelInfo::getReplication LengthTimeUnit
Model::getControls
Model::getResponses
List::clear
Model::getComponentManager
Model::getTraceManager
TraceManager::trace
Model::checkModel
Util::IncIndent
Util::DecIndent
Util::SetIndent
Model::getOnEventManager
OnEventManager::NotifySimulation StartHandlers
ModelInfo::getNumberOfReplications
OnEventManager::NotifyReplication StartHandlers
OnEventManager::NotifyReplication EndHandlers
Model::getEvents
List::empty
List::front
ModelSimulation::startSimulation
Event::getTime
ModelInfo::getReplication Length
Model::parseExpression
ModelInfo::getTerminating Condition
SimulationReporter _if::showReplicationStatistics
SimulationReporter _if::showSimulationStatistics
ModelInfo::getName
OnEventManager::NotifySimulation EndHandlers
Model::getElementManager
ElementManager::getElements
List::getList
StatisticsCollector ::getParent
ModelElement::getName
Counter::getParent
Counter::incCountValue
Counter::getCountValue
TraceManager::traceReport
Model::getParent
Simulator::getName
Simulator::getLicenceManager
LicenceManager::showLicence
LicenceManager::showLimits
ModelInfo::getAnalystName
ModelInfo::getProjectTitle

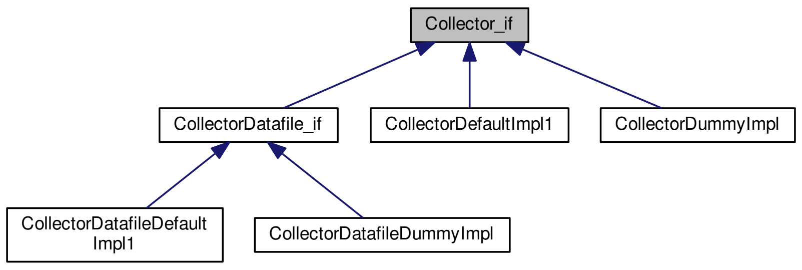
The documentation for this class was generated from the following files:

- BuildSimulationModel.h
- BuildSimulationModel.cpp

5.5 Collector_if Class Reference

```
#include <Collector_if.h>
```

Inheritance diagram for Collector_if:



Public Member Functions

- virtual void clear ()=0
- virtual void addValue (double value)=0
- virtual double getLastValue ()=0
- virtual unsigned long numElements ()=0
- virtual void setAddValueHandler (CollectorAddValueHandler addValueHandler)=0
- virtual void setClearHandler (CollectorClearHandler clearHandler)=0

5.5.1 Detailed Description

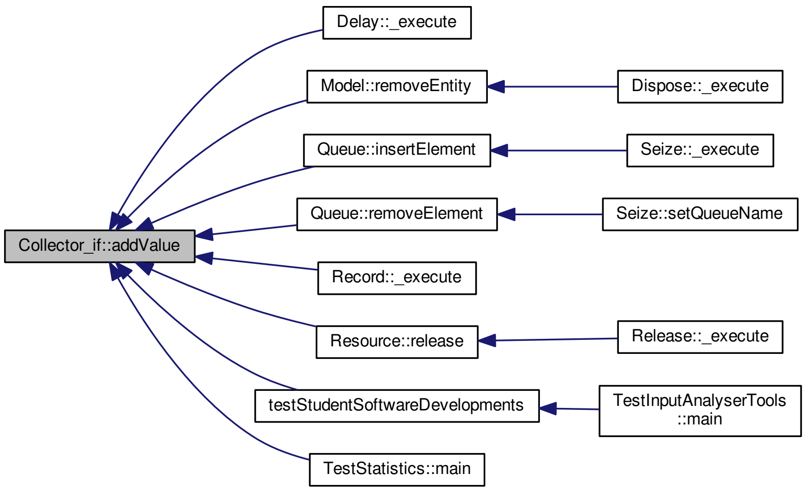
Interface for collecting values of a single stochastic variable. Values collected can be used as base for statistical analysis.

5.5.2 Member Function Documentation

5.5.2.1 virtual void Collector_if::addValue (double *value*) `[pure virtual]`

Implemented in CollectorDatafileDefaultImpl1, CollectorDummyImpl, CollectorDatafileDummyImpl, and Collector←DefaultImpl1.

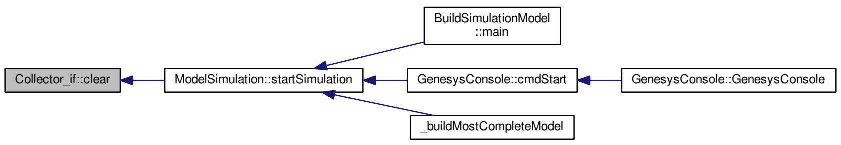
Here is the caller graph for this function:



5.5.2.2 virtual void Collector_if::clear ( ) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, CollectorDummyImpl, CollectorDatafileDummyImpl, and Collector←DefaultImpl1.

Here is the caller graph for this function:


BuildSimulationModel ::main
Collector_if::clear
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

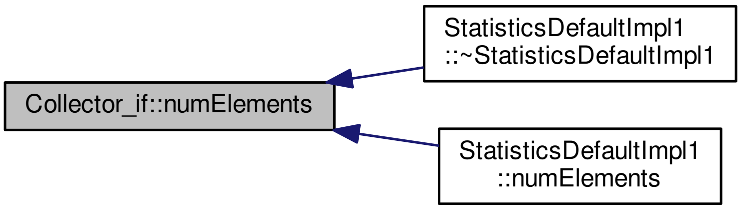
5.5.2.3 virtual double Collector_if::getLastValue ( ) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, CollectorDummyImpl, CollectorDatafileDummyImpl, and Collector←DefaultImpl1.

5.5.2.4 virtual unsigned long Collector_if::numElements ( ) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, CollectorDummyImpl, CollectorDatafileDummyImpl, and Collector←DefaultImpl1.

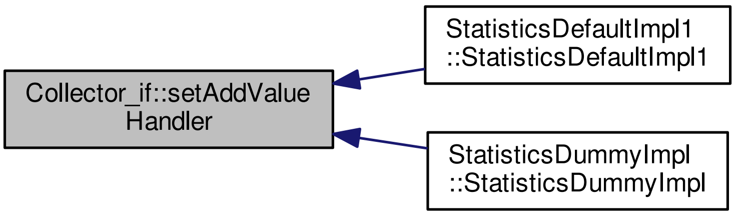
Here is the caller graph for this function:


Collector_if::numElements
StatisticsDefaultImpl1 ::~StatisticsDefaultImpl1
StatisticsDefaultImpl1 ::numElements

5.5.2.5 virtual void Collector_if::setAddValueHandler (**CollectorAddValueHandler** *addValueHandler*) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, CollectorDatafileDummyImpl, CollectorDummyImpl, and Collector←DefaultImpl1.

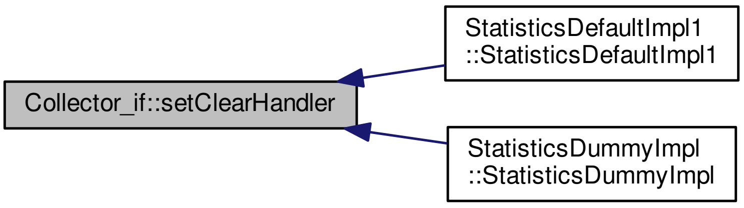
Here is the caller graph for this function:


Collector_if::setAddValue Handler
StatisticsDefaultImpl1 ::StatisticsDefaultImpl1
StatisticsDummyImpl ::StatisticsDummyImpl

5.5.2.6 virtual void Collector_if::setClearHandler (**CollectorClearHandler** *clearHandler*) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, CollectorDatafileDummyImpl, CollectorDummyImpl, and Collector←DefaultImpl1.

Here is the caller graph for this function:


Collector_if::setClearHandler
StatisticsDefaultImpl1 ::StatisticsDefaultImpl1
StatisticsDummyImpl ::StatisticsDummyImpl

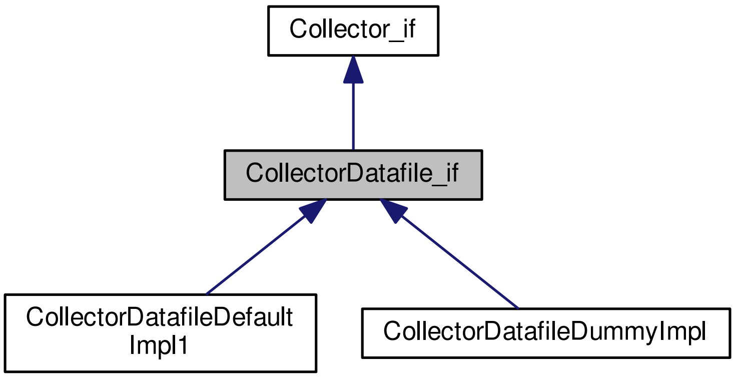
The documentation for this class was generated from the following file:

- Collector_if.h

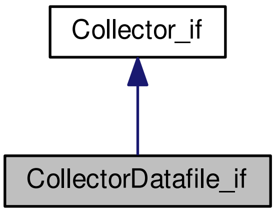
5.6 CollectorDatafile_if Class Reference

```
#include <CollectorDatafile_if.h>
```

Inheritance diagram for CollectorDatafile_if:



Collaboration diagram for CollectorDatafile_if:



Public Member Functions

- virtual double getValue (unsigned int rank)=0
- virtual void seekFirstValue ()=0
- virtual double getNextValue ()=0
- virtual std::string getDataFilename ()=0
- virtual void setDataFilename (std::string filename)=0

5.6.1 Detailed Description

Interface for collecting values of a stochastic variable that will be stores in a datafile.

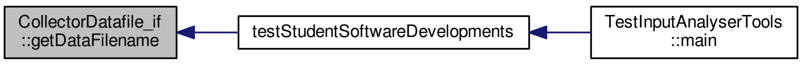
5.6.2 Member Function Documentation

5.6.2.1 virtual std::string CollectorDatafile_if::getDataFilename ( ) [pure virtual]

Get the next value in the file and advances the pointer

Implemented in CollectorDatafileDefaultImpl1, and CollectorDatafileDummyImpl.

Here is the caller graph for this function:


CollectorDatafile_if ::getDataFilename
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.6.2.2 virtual double CollectorDatafile_if::getNextValue ( ) [pure virtual]

Set the pointer to the first value in the file

Implemented in CollectorDatafileDefaultImpl1, and CollectorDatafileDummyImpl.

5.6.2.3 virtual double CollectorDatafile_if::getValue (unsigned int *rank*) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, and CollectorDatafileDummyImpl.

5.6.2.4 virtual void CollectorDatafile_if::seekFirstValue ( ) [pure virtual]

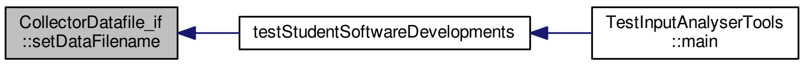
Get a value from a specific position

Implemented in CollectorDatafileDefaultImpl1, and CollectorDatafileDummyImpl.

5.6.2.5 virtual void CollectorDatafile_if::setDataFilename (std::string *filename*) [pure virtual]

Implemented in CollectorDatafileDefaultImpl1, and CollectorDatafileDummyImpl.

Here is the caller graph for this function:


CollectorDatafile_if ::setDataFilename
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

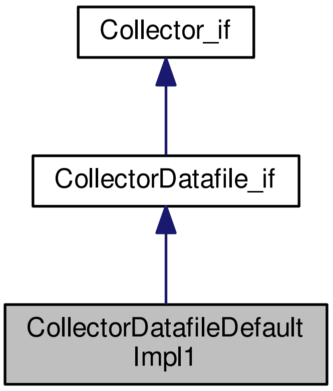
The documentation for this class was generated from the following file:

- CollectorDatafile_if.h

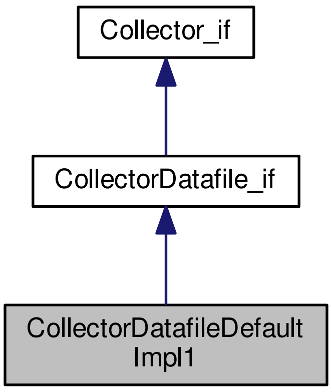
5.7 CollectorDatafileDefaultImpl1 Class Reference

```
#include <CollectorDatafileDefaultImpl1.h>
```

Inheritance diagram for CollectorDatafileDefaultImpl1:



Collaboration diagram for CollectorDatafileDefaultImpl1:



Public Member Functions

- CollectorDatafileDefaultImpl1 ()
- CollectorDatafileDefaultImpl1 (const CollectorDatafileDefaultImpl1 &orig)
- virtual ∼CollectorDatafileDefaultImpl1 ()
- void clear ()
- void addValue (double value)
- double getLastValue ()
- unsigned long numElements ()
- double getValue (unsigned int num)
- double getNextValue ()
- void seekFirstValue ()
- std::string getDataFilename ()
- void setDataFilename (std::string filename)
- void setAddValueHandler (CollectorAddValueHandler addValueHandler)
- void setClearHandler (CollectorClearHandler clearHandler)

5.7.1 Constructor & Destructor Documentation

5.7.1.1 CollectorDatafileDefaultImpl1::CollectorDatafileDefaultImpl1 ( )

5.7.1.2 CollectorDatafileDefaultImpl1::CollectorDatafileDefaultImpl1 (const CollectorDatafileDefaultImpl1 & *orig*)

5.7.1.3 CollectorDatafileDefaultImpl1::∼CollectorDatafileDefaultImpl1 ( ) [virtual]

5.7.2 Member Function Documentation

5.7.2.1 void CollectorDatafileDefaultImpl1::addValue (double *value*) [virtual]

Implements Collector_if.

5.7.2.2 void CollectorDatafileDefaultImpl1::clear ( ) [virtual]

Implements Collector_if.

5.7.2.3 std::string CollectorDatafileDefaultImpl1::getDataFilename ( ) [virtual]

Get the next value in the file and advances the pointer

Implements CollectorDatafile_if.

5.7.2.4 double CollectorDatafileDefaultImpl1::getLastValue ( ) [virtual]

Implements Collector_if.

5.7.2.5 double CollectorDatafileDefaultImpl1::getNextValue ( ) [virtual]

Set the pointer to the first value in the file

Implements CollectorDatafile_if.

5.7.2.6 double CollectorDatafileDefaultImpl1::getValue (unsigned int *num*) [virtual]

Implements CollectorDatafile_if.

5.7.2.7 unsigned long CollectorDatafileDefaultImpl1::numElements ( ) [virtual]

Implements Collector_if.

5.7.2.8 void CollectorDatafileDefaultImpl1::seekFirstValue ( ) [virtual]

Get a value from a specific position

Implements CollectorDatafile_if.

5.7.2.9 void CollectorDatafileDefaultImpl1::setAddValueHandler (CollectorAddValueHandler *addValueHandler*) [virtual]

Implements Collector_if.

5.7.2.10 void CollectorDatafileDefaultImpl1::setClearHandler (CollectorClearHandler *clearHandler*) [virtual]

Implements Collector_if.

5.7.2.11 void CollectorDatafileDefaultImpl1::setDataFilename (std::string *filename*) `[virtual]`

Implements CollectorDatafile_if.

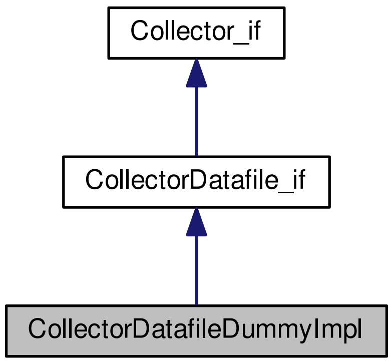
The documentation for this class was generated from the following files:

- CollectorDatafileDefaultImpl1.h
- CollectorDatafileDefaultImpl1.cpp

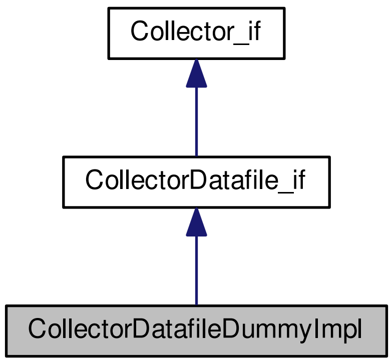
5.8 CollectorDatafileDummyImpl Class Reference

`#include <CollectorDatafileDummyImpl.h>`

Inheritance diagram for CollectorDatafileDummyImpl:



Collaboration diagram for CollectorDatafileDummyImpl:



Public Member Functions

- CollectorDatafileDummyImpl ()
- CollectorDatafileDummyImpl (const CollectorDatafileDummyImpl &orig)
- ∼CollectorDatafileDummyImpl ()
- void clear ()
- void addValue (double value)
- double getLastValue ()
- unsigned long numElements ()
- double getValue (unsigned int num)
- double getNextValue ()
- void seekFirstValue ()
- std::string getDataFilename ()
- void setDataFilename (std::string filename)
- void setAddValueHandler (CollectorAddValueHandler addValueHandler)
- void setClearHandler (CollectorClearHandler clearHandler)

5.8.1 Constructor & Destructor Documentation

5.8.1.1 CollectorDatafileDummyImpl::CollectorDatafileDummyImpl ( )

5.8.1.2 CollectorDatafileDummyImpl::CollectorDatafileDummyImpl (const CollectorDatafileDummyImpl & *orig*)

5.8.1.3 CollectorDatafileDummyImpl::∼CollectorDatafileDummyImpl ( )

5.8.2 Member Function Documentation

5.8.2.1 void CollectorDatafileDummyImpl::addValue (double *value*) [virtual]

Implements Collector_if.

5.8.2.2 void CollectorDatafileDummyImpl::clear ( ) [virtual]

Implements Collector_if.

5.8.2.3 std::string CollectorDatafileDummyImpl::getDataFilename ( ) [virtual]

Get the next value in the file and advances the pointer

Implements CollectorDatafile_if.

5.8.2.4 double CollectorDatafileDummyImpl::getLastValue ( ) [virtual]

Implements Collector_if.

5.8.2.5 double CollectorDatafileDummyImpl::getNextValue ( ) [virtual]

Set the pointer to the first value in the file

Implements CollectorDatafile_if.

5.8.2.6 double CollectorDatafileDummyImpl::getValue (unsigned int *num*) [virtual]

Implements CollectorDatafile_if.

5.8.2.7 unsigned long CollectorDatafileDummyImpl::numElements ( ) [virtual]

Implements Collector_if.

5.8.2.8 void CollectorDatafileDummyImpl::seekFirstValue ( ) [virtual]

Get a value from a specific position

Implements CollectorDatafile_if.

5.8.2.9 void CollectorDatafileDummyImpl::setAddValueHandler (CollectorAddValueHandler *addValueHandler*) [virtual]

Implements Collector_if.

5.8.2.10 void CollectorDatafileDummyImpl::setClearHandler (CollectorClearHandler *clearHandler*) [virtual]

Implements Collector_if.

5.8.2.11 void CollectorDatafileDummyImpl::setDataFilename (std::string *filename*) [virtual]

Implements CollectorDatafile_if.

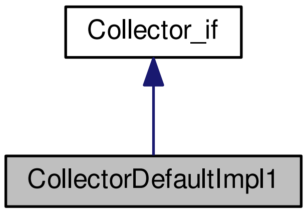
The documentation for this class was generated from the following files:

- CollectorDatafileDummyImpl.h
- CollectorDatafileDummyImpl.cpp

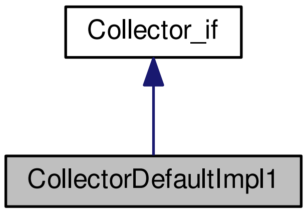
5.9 CollectorDefaultImpl1 Class Reference

```
#include <CollectorDefaultImpl1.h>
```

Inheritance diagram for CollectorDefaultImpl1:



Collaboration diagram for CollectorDefaultImpl1:



Public Member Functions

- CollectorDefaultImpl1 ()
- CollectorDefaultImpl1 (const CollectorDefaultImpl1 &orig)
- virtual ∼CollectorDefaultImpl1 ()
- void clear ()
- void addValue (double value)
- double getLastValue ()
- unsigned long numElements ()
- void setAddValueHandler (CollectorAddValueHandler addValueHandler)
- void setClearHandler (CollectorClearHandler clearHandler)

5.9.1 Constructor & Destructor Documentation

5.9.1.1 CollectorDefaultImpl1::CollectorDefaultImpl1 ( )

5.9.1.2 CollectorDefaultImpl1::CollectorDefaultImpl1 (const CollectorDefaultImpl1 & *orig*)

5.9.1.3 CollectorDefaultImpl1::∼CollectorDefaultImpl1 ( ) `[virtual]`

5.9.2 Member Function Documentation

5.9.2.1 void CollectorDefaultImpl1::addValue (double *value*) `[virtual]`

Implements Collector_if.

5.9.2.2 void CollectorDefaultImpl1::clear ( ) `[virtual]`

Implements Collector_if.

5.9.2.3 double CollectorDefaultImpl1::getLastValue ( ) `[virtual]`

Implements Collector_if.

5.9.2.4 unsigned long CollectorDefaultImpl1::numElements ( ) `[virtual]`

Implements Collector_if.

5.9.2.5 void CollectorDefaultImpl1::setAddValueHandler (CollectorAddValueHandler *addValueHandler*) `[virtual]`

Implements Collector_if.

5.9.2.6 void CollectorDefaultImpl1::setClearHandler (CollectorClearHandler *clearHandler*) `[virtual]`

Implements Collector_if.

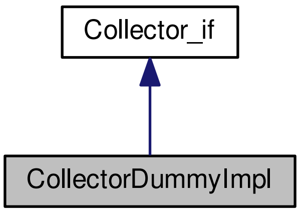
The documentation for this class was generated from the following files:

- CollectorDefaultImpl1.h
- CollectorDefaultImpl1.cpp

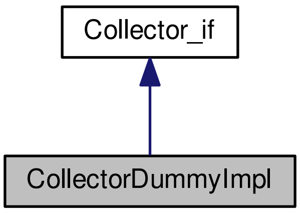
5.10 CollectorDummyImpl Class Reference

```
#include <CollectorDummyImpl.h>
```

Inheritance diagram for CollectorDummyImpl:



Collaboration diagram for CollectorDummyImpl:



Public Member Functions

- CollectorDummyImpl ()
- CollectorDummyImpl (const CollectorDummyImpl &orig)
- ∼CollectorDummyImpl ()
- void clear ()
- void addValue (double value)
- double getLastValue ()
- unsigned long numElements ()
- void setAddValueHandler (CollectorAddValueHandler addValueHandler)
- void setClearHandler (CollectorClearHandler clearHandler)

5.10.1 Constructor & Destructor Documentation

5.10.1.1 CollectorDummyImpl::CollectorDummyImpl ( )

5.10.1.2 CollectorDummyImpl::CollectorDummyImpl (const **CollectorDummyImpl** & *orig*)

5.10.1.3 CollectorDummyImpl::∼CollectorDummyImpl ( )

5.10.2 Member Function Documentation

5.10.2.1 void CollectorDummyImpl::addValue (double *value*) `[virtual]`

Implements Collector_if.

5.10.2.2 void CollectorDummyImpl::clear ( ) `[virtual]`

Implements Collector_if.

5.10.2.3 double CollectorDummyImpl::getLastValue ( ) `[virtual]`

Implements Collector_if.

5.10.2.4 unsigned long CollectorDummyImpl::numElements ( ) `[virtual]`

Implements Collector_if.

5.10.2.5 void CollectorDummyImpl::setAddValueHandler (**CollectorAddValueHandler** *addValueHandler*) `[virtual]`

Implements Collector_if.

5.10.2.6 void CollectorDummyImpl::setClearHandler (**CollectorClearHandler** *clearHandler*) `[virtual]`

Implements Collector_if.

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

- CollectorDummyImpl.h
- CollectorDummyImpl.cpp

5.11 ComponentManager Class Reference

```
#include <ComponentManager.h>
```

Public Member Functions

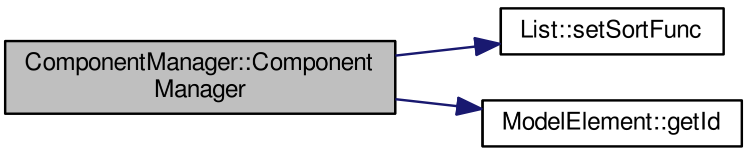
- ComponentManager (Model ∗model)
- ComponentManager (const ComponentManager &orig)
- virtual ∼ComponentManager ()
- bool insert (ModelComponent ∗comp)
- void remove (ModelComponent ∗comp)
- void clear ()
- ModelComponent ∗ getComponent (Util::identitifcation id)
- ModelComponent ∗ getComponent (std::string name)
- unsigned int getNumberOfComponents ()
- std::list< ModelComponent ∗ >::iterator begin ()
- std::list< ModelComponent ∗ >::iterator end ()

5.11.1 Constructor & Destructor Documentation

5.11.1.1 ComponentManager::ComponentManager (**Model** ∗ ***model***)

Components are sorted by ID

Here is the call graph for this function:



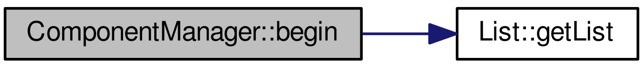
5.11.1.2 ComponentManager::ComponentManager (const **ComponentManager** & ***orig***)

5.11.1.3 ComponentManager::∼ComponentManager ( ) [virtual]

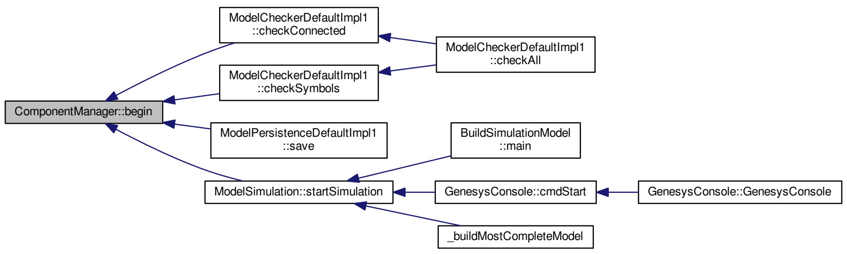
5.11.2 Member Function Documentation

5.11.2.1 std::list< **ModelComponent** ∗ >::iterator ComponentManager::begin ( )

Here is the call graph for this function:

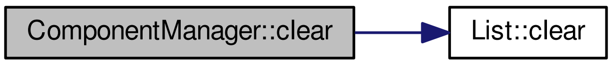


Here is the caller graph for this function:

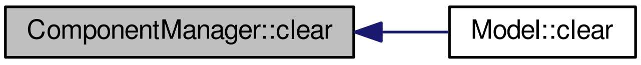

ComponentManager::begin
ModelCheckerDefaultImpl1 ::checkConnected
ModelCheckerDefaultImpl1 ::checkAll
ModelCheckerDefaultImpl1 ::checkSymbols
ModelPersistenceDefaultImpl1 ::save
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.11.2.2 void ComponentManager::clear ( )

Here is the call graph for this function:

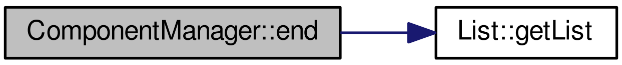


Here is the caller graph for this function:

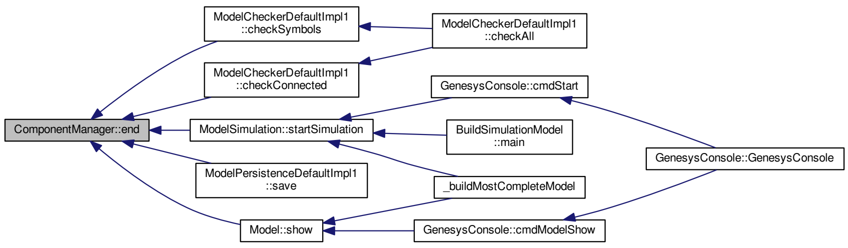


5.11.2.3 std::list< **ModelComponent** ∗ >::iterator ComponentManager::end ( )

Here is the call graph for this function:



Here is the caller graph for this function:

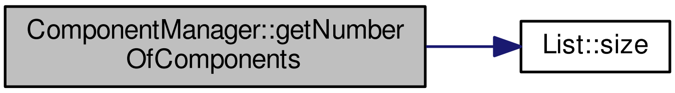

ComponentManager::end
ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkConnected
ModelSimulation::startSimulation
ModelPersistenceDefaultImpl1 ::save
Model::show
ModelCheckerDefaultImpl1 ::checkAll
GenesysConsole::cmdStart
BuildSimulationModel ::main
_buildMostCompleteModel
GenesysConsole::cmdModelShow
GenesysConsole::GenesysConsole

5.11.2.4 ModelComponent ∗ ComponentManager::getComponent (Util::identitifcation *id*)

5.11.2.5 ModelComponent ∗ ComponentManager::getComponent (std::string *name*)

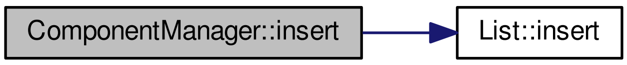
5.11.2.6 unsigned int ComponentManager::getNumberOfComponents ( )

Here is the call graph for this function:

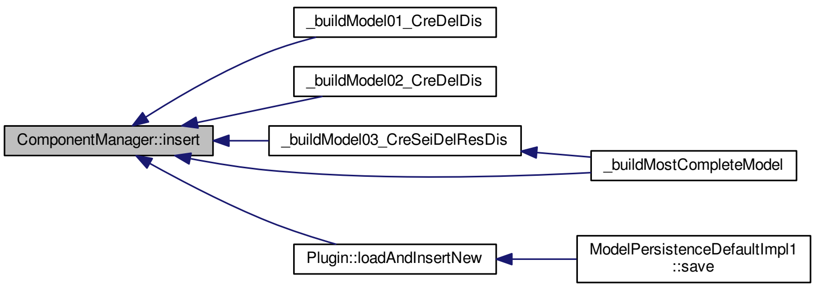


5.11.2.7 bool ComponentManager::insert (ModelComponent ∗ *comp*)

Here is the call graph for this function:

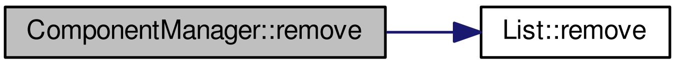


Here is the caller graph for this function:



5.11.2.8 void ComponentManager::remove (ModelComponent ∗ *comp*)

Here is the call graph for this function:



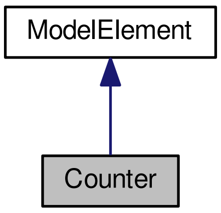
The documentation for this class was generated from the following files:

- ComponentManager.h
- ComponentManager.cpp

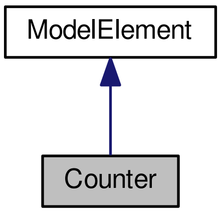
5.12 Counter Class Reference

`#include <Counter.h>`

Inheritance diagram for Counter:



Collaboration diagram for Counter:



Public Member Functions

- Counter ()
- Counter (std::string name)
- Counter (std::string name, ModelElement ∗parent)
- Counter (const Counter &orig)
- virtual ∼Counter ()
- virtual std::string show ()
- void clear ()
- void incCountValue (int value=1)
- unsigned long getCountValue ()
- ModelElement ∗ getParent () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

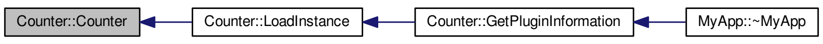
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.12.1 Constructor & Destructor Documentation

5.12.1.1 Counter::Counter ( )

Here is the caller graph for this function:



5.12.1.2 Counter::Counter (std::string *name*)

5.12.1.3 Counter::Counter (std::string *name*, ModelElement ∗ *parent*)

5.12.1.4 Counter::Counter (const Counter & *orig*)

5.12.1.5 Counter::∼Counter ( ) [virtual]

5.12.2 Member Function Documentation

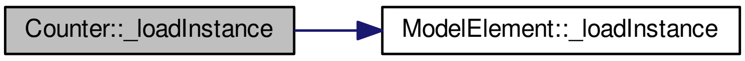
5.12.2.1 bool Counter::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

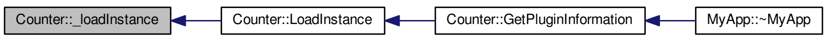
5.12.2.2 bool Counter::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



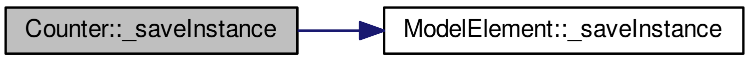
Here is the caller graph for this function:



5.12.2.3 std::map< std::string, std::string > ∗ Counter::_saveInstance ( ) [protected], [virtual]

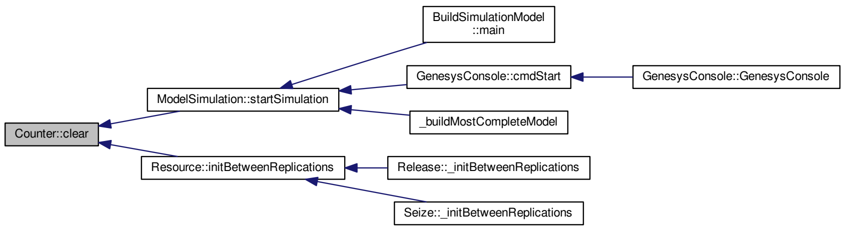
Reimplemented from ModelElement.

Here is the call graph for this function:



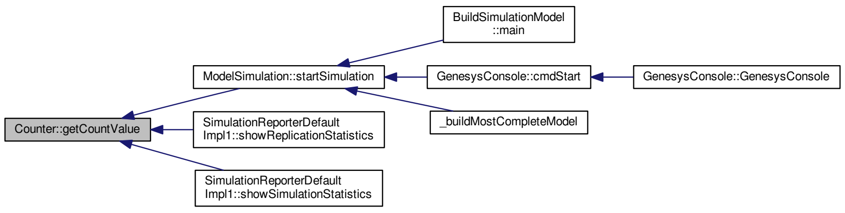
5.12.2.4 void Counter::clear ( )

Here is the caller graph for this function:


BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
ModelSimulation::startSimulation
_buildMostCompleteModel
Counter::clear
Resource::initBetweenReplications
Release::_initBetweenReplications
Seize::_initBetweenReplications

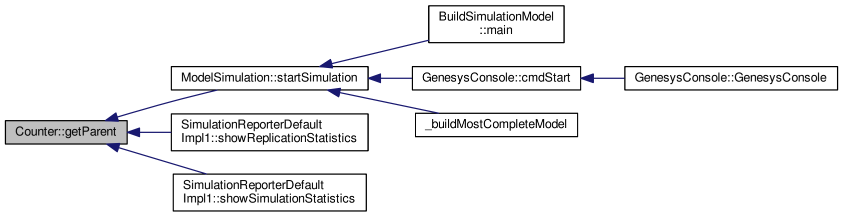
5.12.2.5 unsigned long Counter::getCountValue ( )

Here is the caller graph for this function:


BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
Counter::getCountValue
SimulationReporterDefault Impl1::showReplicationStatistics
_buildMostCompleteModel
SimulationReporterDefault Impl1::showSimulationStatistics

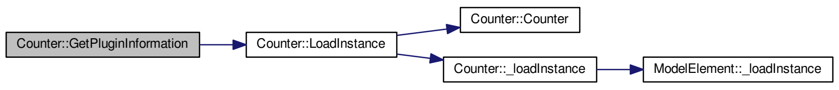
5.12.2.6 ModelElement ∗ Counter::getParent ( ) const

Here is the caller graph for this function:

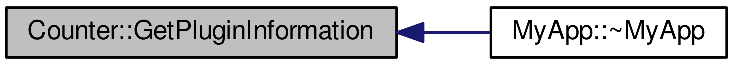

BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
Counter::getParent
SimulationReporterDefault Impl1::showReplicationStatistics
_buildMostCompleteModel
SimulationReporterDefault Impl1::showSimulationStatistics

5.12.2.7 PluginInformation ∗ Counter::GetPluginInformation ( ) [static]

Here is the call graph for this function:

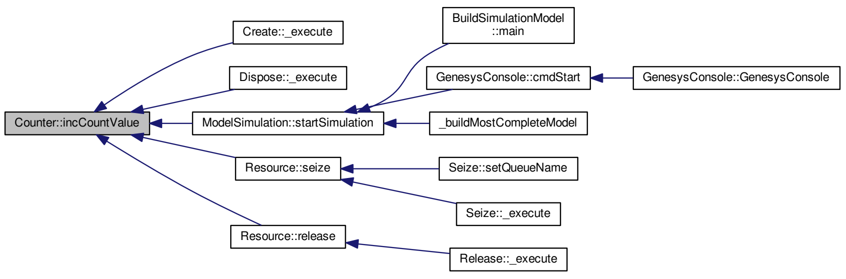


Here is the caller graph for this function:



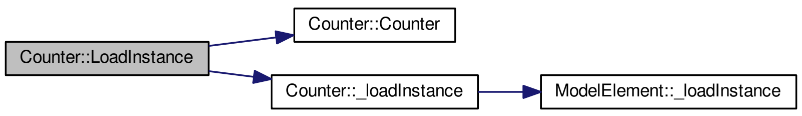
5.12.2.8 void Counter::incCountValue (int *value* = 1)

Here is the caller graph for this function:

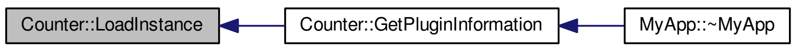


5.12.2.9 ModelElement ∗ Counter::LoadInstance (ElementManager ∗ *elems,* std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:



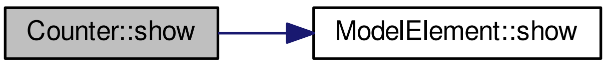
Here is the caller graph for this function:



5.12.2.10 std::string Counter::show ( ) `[virtual]`

Reimplemented from ModelElement.

Here is the call graph for this function:



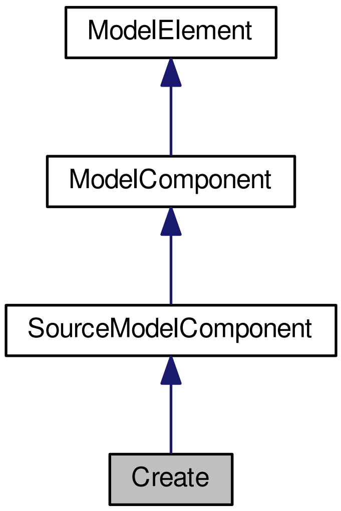
The documentation for this class was generated from the following files:

- Counter.h
- Counter.cpp

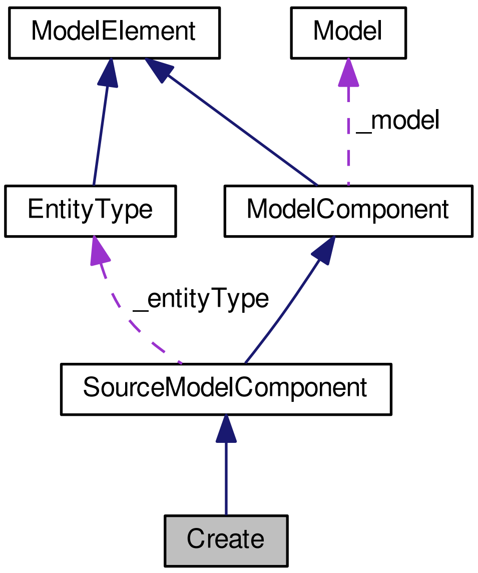
5.13 Create Class Reference

```
#include <Create.h>
```

Inheritance diagram for Create:



Collaboration diagram for Create:



Public Member Functions

- Create (Model ∗model)

- Create (const Create &orig)
- virtual ∼Create ()
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

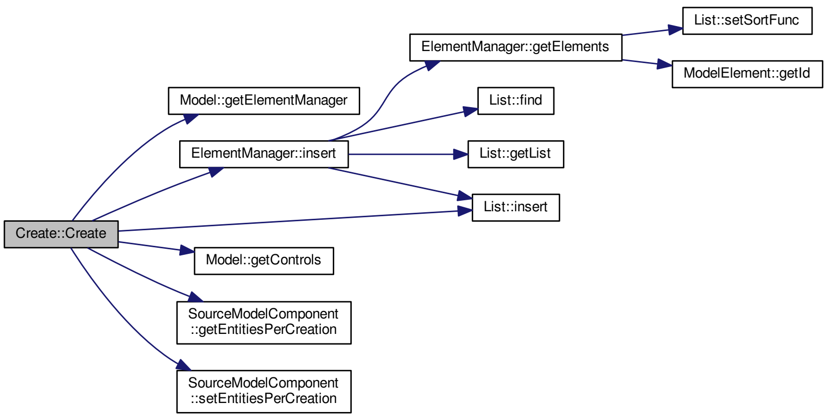
5.13.1 Detailed Description

Create is the most basic component to include the first entities into the model, and therefore is a source component (derived from SourceModelComponent)

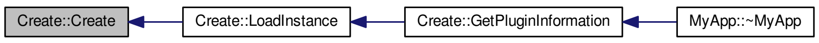
5.13.2 Constructor & Destructor Documentation

5.13.2.1 Create::Create (**Model** ∗ ***model***)

Here is the call graph for this function:



Here is the caller graph for this function:



5.13.2.2 Create::Create (const Create & *orig*)

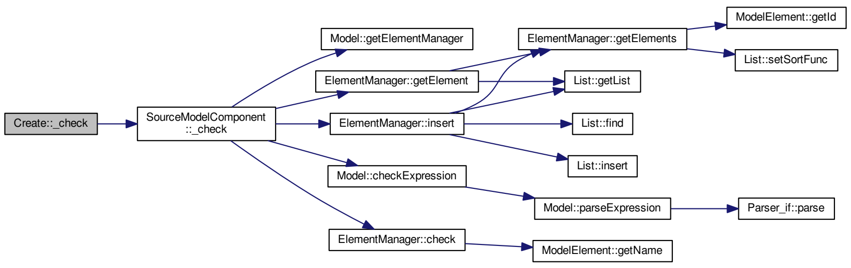
5.13.2.3 Create::∼Create ( ) [virtual]

5.13.3 Member Function Documentation

5.13.3.1 bool Create::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from SourceModelComponent.

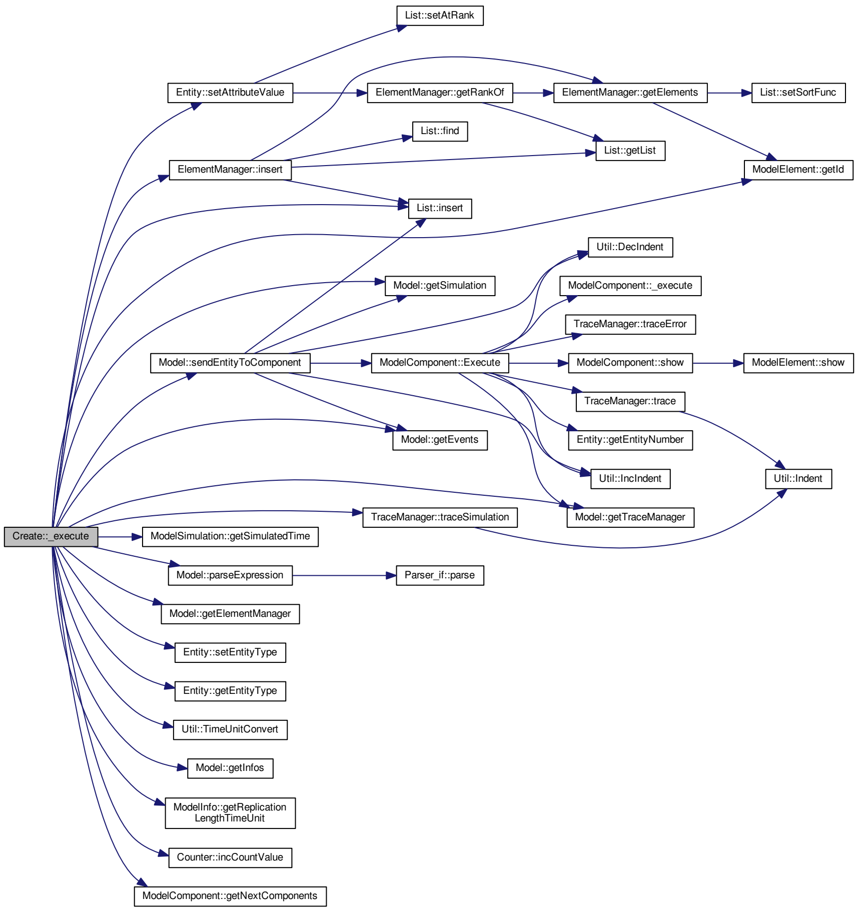
Here is the call graph for this function:



5.13.3.2 void Create::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

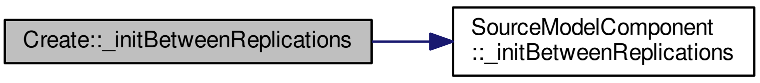
Here is the call graph for this function:



5.13.3.3 void Create::_initBetweenReplications ( ) [protected], [virtual]

Reimplemented from SourceModelComponent.

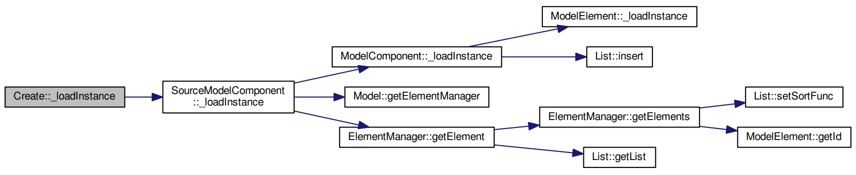
Here is the call graph for this function:


Create::_initBetweenReplications
SourceModelComponent ::_initBetweenReplications

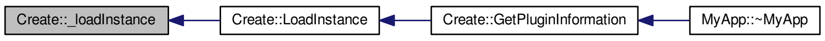
5.13.3.4 bool Create::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from SourceModelComponent.

Here is the call graph for this function:


Create::_loadInstance
SourceModelComponent ::_loadInstance
ModelComponent::_loadInstance
Model::getElementManager
ElementManager::getElement
ModelElement::_loadInstance
List::insert
ElementManager::getElements
List::getList
List::setSortFunc
ModelElement::getId

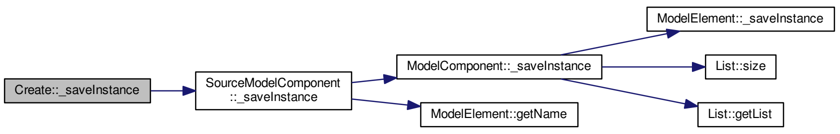
Here is the caller graph for this function:



5.13.3.5 std::map< std::string, std::string > * Create::_saveInstance ( ) [protected], [virtual]

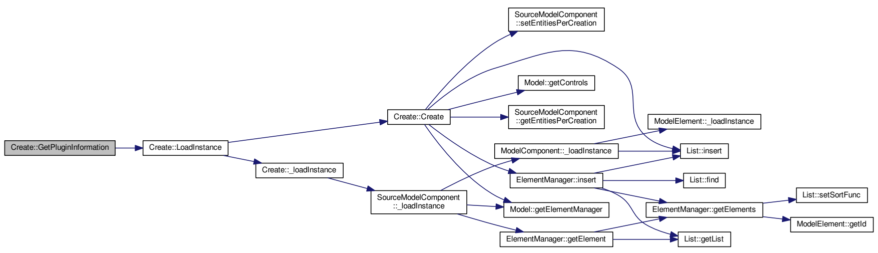
Reimplemented from SourceModelComponent.

Here is the call graph for this function:

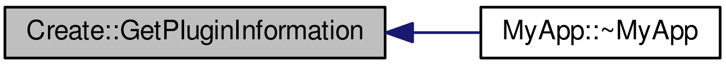

Create::_saveInstance
SourceModelComponent ::_saveInstance
ModelComponent::_saveInstance
ModelElement::getName
ModelElement::_saveInstance
List::size
List::getList

5.13.3.6 PluginInformation * Create::GetPluginInformation ( ) [static]

Here is the call graph for this function:

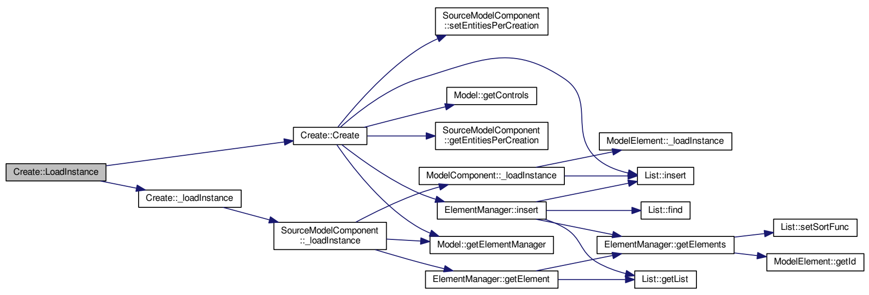

Create::GetPluginInformation
Create::LoadInstance
Create::Create
Create::_loadInstance
SourceModelComponent ::_loadInstance
SourceModelComponent ::setEntitiesPerCreation
Model::getControls
SourceModelComponent ::getEntitiesPerCreation
ModelComponent::_loadInstance
ElementManager::insert
Model::getElementManager
ElementManager::getElement
ModelElement::_loadInstance
List::insert
List::find
ElementManager::getElements
List::getList
List::setSortFunc
ModelElement::getId

Here is the caller graph for this function:

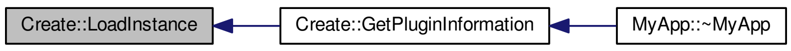


5.13.3.7 ModelComponent ∗ Create::LoadInstance (Model ∗ *model*, std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:


Create::LoadInstance
Create::Create
Create::_loadInstance
SourceModelComponent ::setEntitiesPerCreation
Model::getControls
SourceModelComponent ::getEntitiesPerCreation
ModelElement::_loadInstance
ModelComponent::_loadInstance
List::insert
ElementManager::insert
List::find
SourceModelComponent ::_loadInstance
Model::getElementManager
ElementManager::getElements
List::setSortFunc
ModelElement::getId
ElementManager::getElement
List::getList

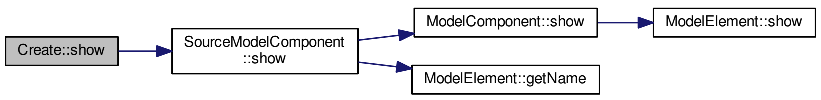
Here is the caller graph for this function:



5.13.3.8 std::string Create::show ( ) [virtual]

Reimplemented from SourceModelComponent.

Here is the call graph for this function:



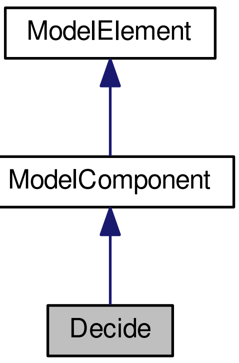
The documentation for this class was generated from the following files:

- Create.h
- Create.cpp

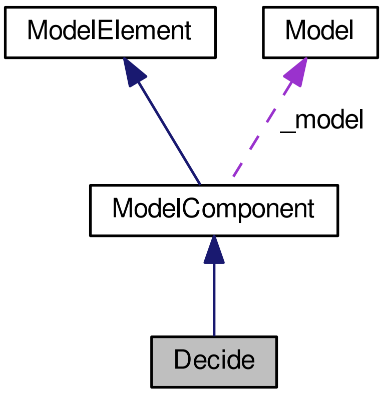
5.14 Decide Class Reference

```
#include <Decide.h>
```

Inheritance diagram for Decide:



Collaboration diagram for Decide:



Public Member Functions

- Decide (Model ∗model)
- Decide (const Decide &orig)
- virtual ∼Decide ()
- List< std::string > ∗ getConditions () const
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

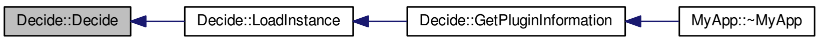
- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.14.1 Constructor & Destructor Documentation

5.14.1.1 Decide::Decide (**Model** ∗ *model*)

Here is the caller graph for this function:



5.14.1.2 Decide::Decide (const **Decide** & *orig*)

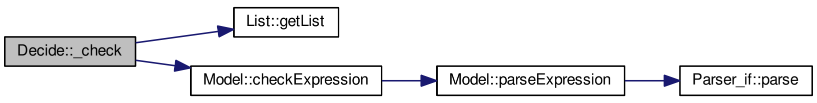
5.14.1.3 Decide::∼Decide ( ) [virtual]

5.14.2 Member Function Documentation

5.14.2.1 bool Decide::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

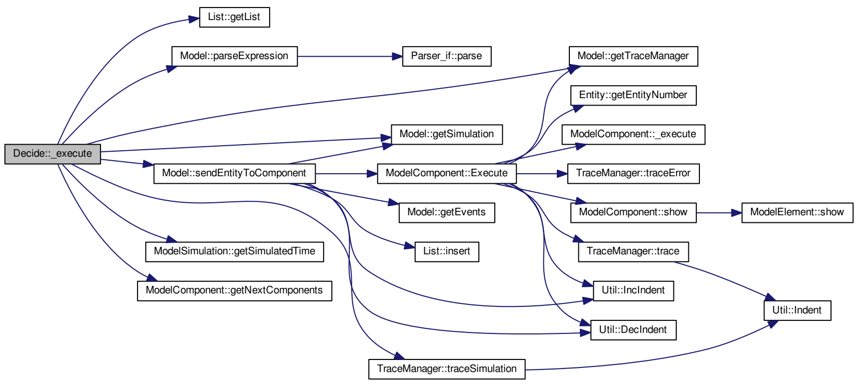
Here is the call graph for this function:



5.14.2.2 void Decide::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

Here is the call graph for this function:



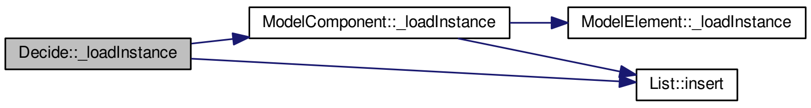
5.14.2.3 void Decide::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

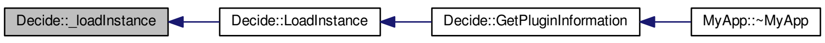
5.14.2.4 bool Decide::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



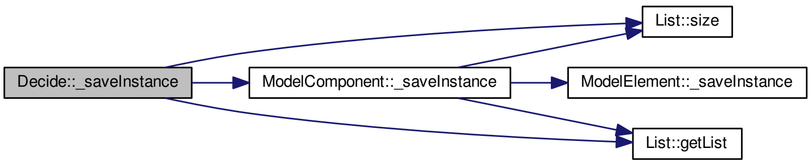
Here is the caller graph for this function:



5.14.2.5 std::map< std::string, std::string > ∗ Decide::_saveInstance ( ) [protected], [virtual]

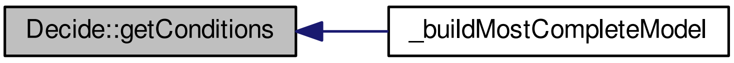
Reimplemented from ModelComponent.

Here is the call graph for this function:



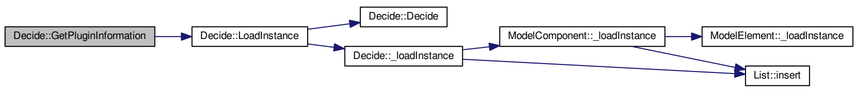
5.14.2.6 List< std::string > ∗ Decide::getConditions ( ) const

Here is the caller graph for this function:

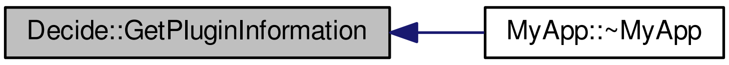


5.14.2.7 PluginInformation ∗ Decide::GetPluginInformation ( ) [static]

Here is the call graph for this function:

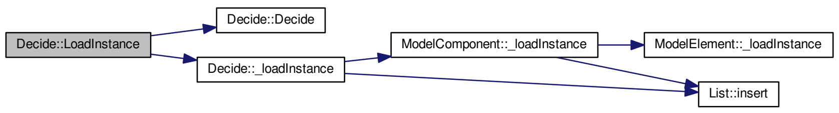

Decide::GetPluginInformation
Decide::LoadInstance
Decide::Decide
Decide::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
List::insert

Here is the caller graph for this function:

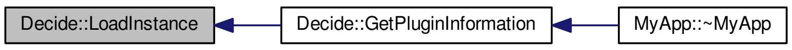


5.14.2.8 **ModelComponent** ∗ Decide::LoadInstance (**Model** ∗ *model*, std::map< std::string, std::string > ∗ *fields*) `[static]`

Here is the call graph for this function:



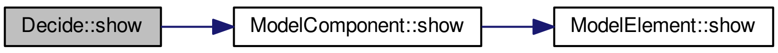
Here is the caller graph for this function:



5.14.2.9 std::string Decide::show ( ) `[virtual]`

Reimplemented from ModelComponent.

Here is the call graph for this function:



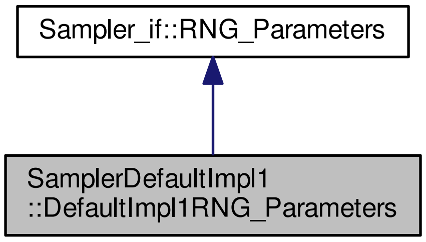
The documentation for this class was generated from the following files:

- Decide.h
- Decide.cpp

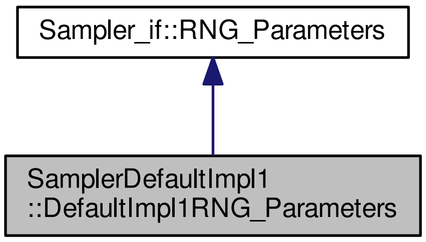
5.15 SamplerDefaultImpl1::DefaultImpl1RNG_Parameters Struct Reference

```
#include <SamplerDefaultImpl1.h>
```

Inheritance diagram for SamplerDefaultImpl1::DefaultImpl1RNG_Parameters:


Sampler_if::RNG_Parameters
SamplerDefaultImpl1 ::DefaultImpl1RNG_Parameters

Collaboration diagram for SamplerDefaultImpl1::DefaultImpl1RNG_Parameters:


Sampler_if::RNG_Parameters
SamplerDefaultImpl1 ::DefaultImpl1RNG_Parameters

Public Member Functions

- ~DefaultImpl1RNG_Parameters ()=default

Public Attributes

- unsigned int seed = 666
- unsigned int module = 2147483647
- unsigned int multiplier = 950706376

5.15.1 Constructor & Destructor Documentation

5.15.1.1 SamplerDefaultImpl1::DefaultImpl1RNG_Parameters::~DefaultImpl1RNG_Parameters ( ) [default]

5.15.2 Member Data Documentation

5.15.2.1 unsigned int SamplerDefaultImpl1::DefaultImpl1RNG_Parameters::module = 2147483647

5.15.2.2 unsigned int SamplerDefaultImpl1::DefaultImpl1RNG_Parameters::multiplier = 950706376

5.15.2.3 unsigned int SamplerDefaultImpl1::DefaultImpl1RNG_Parameters::seed = 666

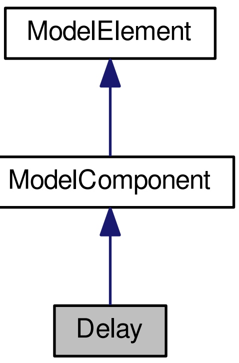
The documentation for this struct was generated from the following file:

- SamplerDefaultImpl1.h

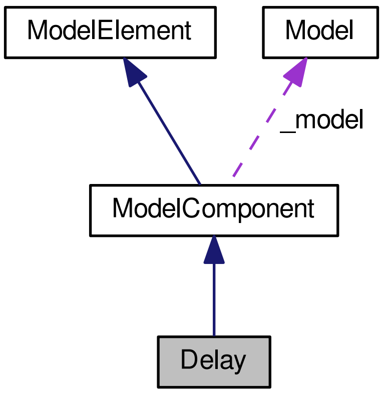
5.16 Delay Class Reference

```
#include <Delay.h>
```

Inheritance diagram for Delay:



Collaboration diagram for Delay:



Public Member Functions

- Delay (Model ∗model)
- Delay (const Delay &orig)
- virtual ∼Delay ()
- void setDelayExpression (std::string _delayExpression)
- std::string getDelayExpression () const
- void setDelayTimeUnit (Util::TimeUnit _delayTimeUnit)
- Util::TimeUnit getDelayTimeUnit () const
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

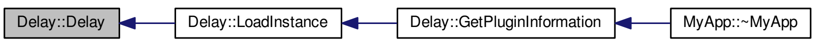
- virtual void _execute (Entity ∗entity)
- virtual void _initBetweenReplications ()
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.16.1 Constructor & Destructor Documentation

5.16.1.1 Delay::Delay (**Model** ∗ ***model***)

Here is the caller graph for this function:



5.16.1.2 Delay::Delay (const Delay & *orig*)

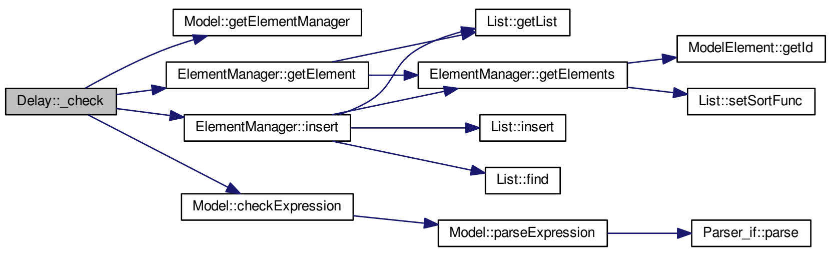
5.16.1.3 Delay::∼Delay ( ) [virtual]

5.16.2 Member Function Documentation

5.16.2.1 bool Delay::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

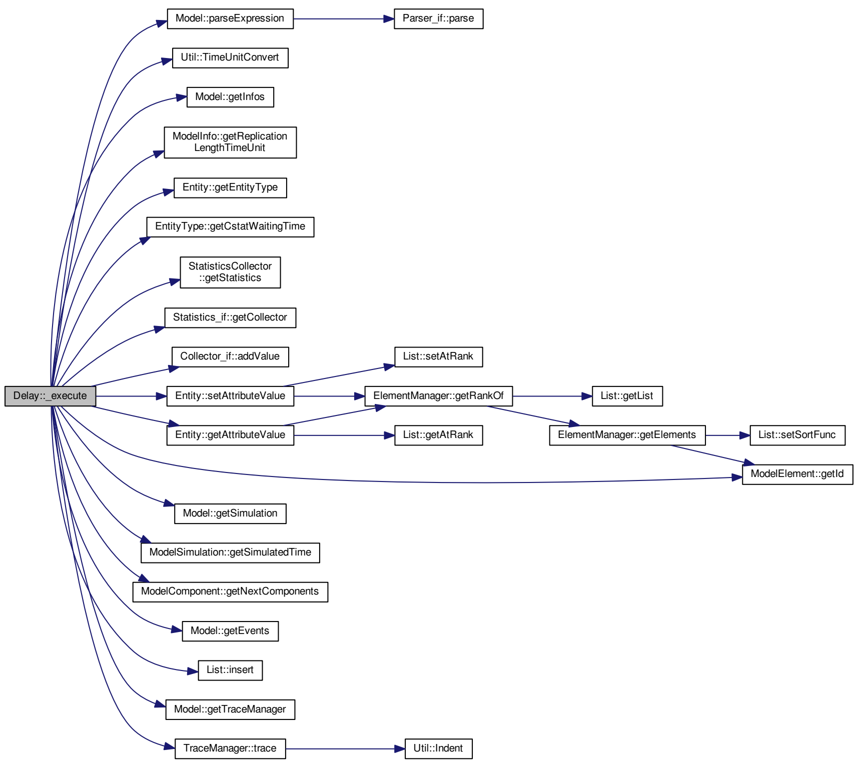
Here is the call graph for this function:



5.16.2.2 void Delay::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

Here is the call graph for this function:



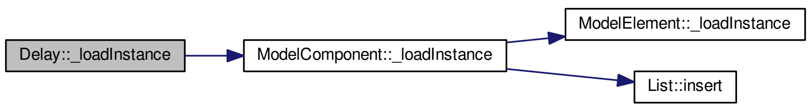
5.16.2.3 void Delay::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

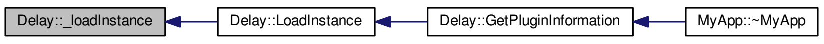
5.16.2.4 bool Delay::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



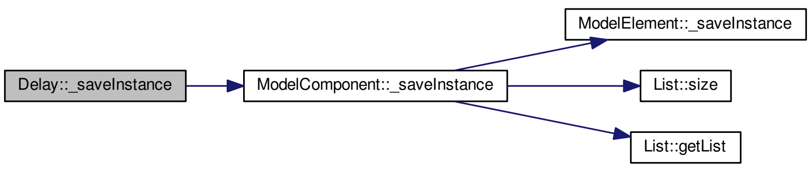
Here is the caller graph for this function:



5.16.2.5 std::map< std::string, std::string > * Delay::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:

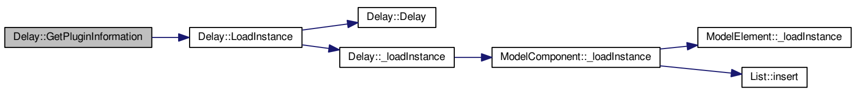


5.16.2.6 std::string Delay::getDelayExpression ( ) const

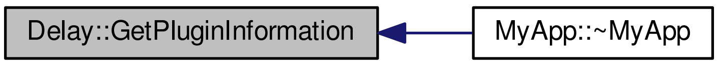
5.16.2.7 Util::TimeUnit Delay::getDelayTimeUnit ( ) const

5.16.2.8 PluginInformation ∗ Delay::GetPluginInformation ( ) [static]

Here is the call graph for this function:

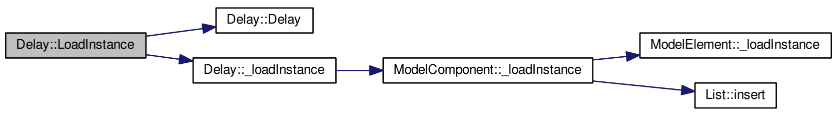

Delay::GetPluginInformation
Delay::LoadInstance
Delay::Delay
Delay::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
List::insert

Here is the caller graph for this function:

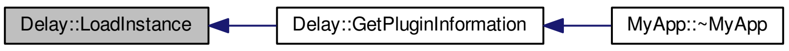


5.16.2.9 ModelComponent ∗ Delay::LoadInstance (Model ∗ *model,* std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:

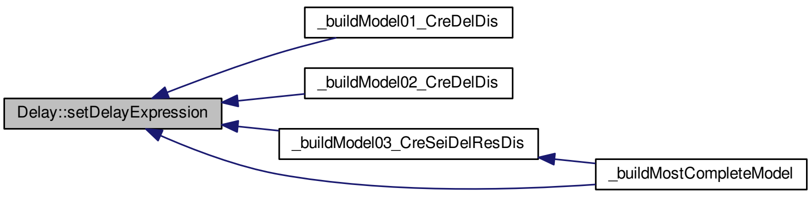


Here is the caller graph for this function:



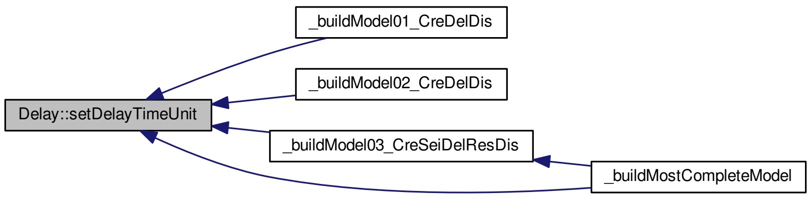
5.16.2.10 void Delay::setDelayExpression (std::string *_delayExpression*)

Here is the caller graph for this function:



5.16.2.11 void Delay::setDelayTimeUnit (Util::TimeUnit *_delayTimeUnit*)

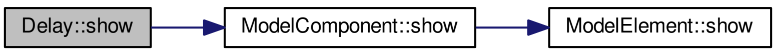
Here is the caller graph for this function:



5.16.2.12 std::string Delay::show ( ) [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



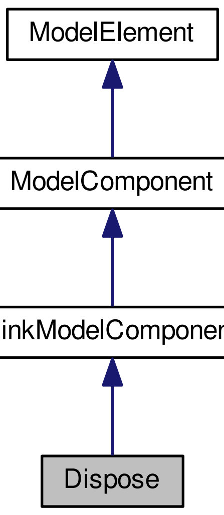
The documentation for this class was generated from the following files:

- Delay.h
- Delay.cpp

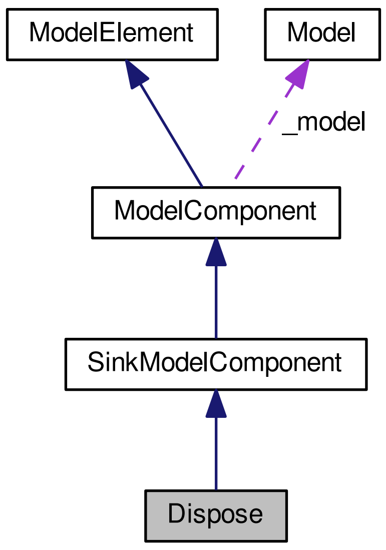
5.17 Dispose Class Reference

```
#include <Dispose.h>
```

Inheritance diagram for Dispose:



Collaboration diagram for Dispose:



Public Member Functions

- Dispose (Model ∗model)
- Dispose (const Dispose &orig)
- virtual ∼Dispose ()
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

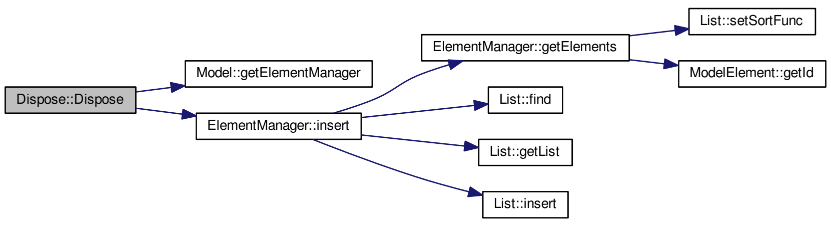
- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

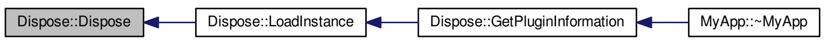
5.17.1 Constructor & Destructor Documentation

5.17.1.1 Dispose::Dispose (**Model** ∗ *model*)

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.1.2 Dispose::Dispose (const Dispose & *orig*)

5.17.1.3 Dispose::∼Dispose ( ) `[virtual]`

5.17.2 Member Function Documentation

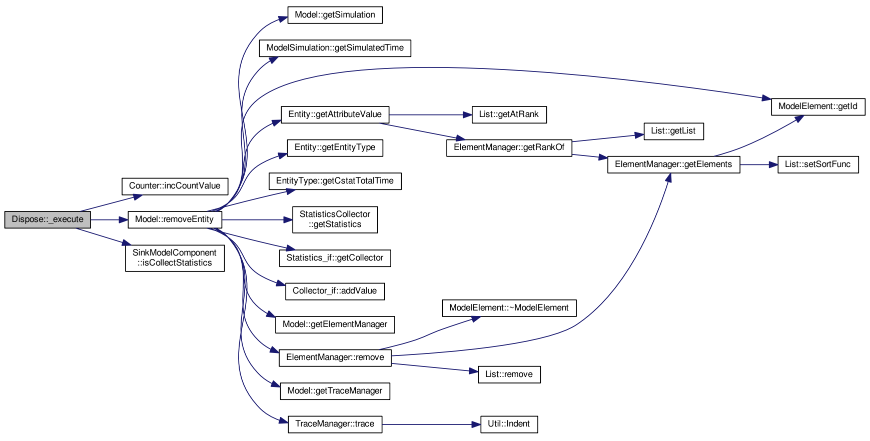
5.17.2.1 bool Dispose::_check (std::string ∗ *errorMessage*) `[protected], [virtual]`

Reimplemented from SinkModelComponent.

5.17.2.2 void Dispose::_execute (Entity ∗ *entity*) `[protected], [virtual]`

Implements ModelComponent.

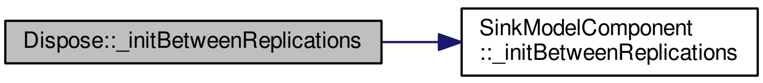
Here is the call graph for this function:



5.17.2.3 void Dispose::_initBetweenReplications ( ) `[protected], [virtual]`

Reimplemented from SinkModelComponent.

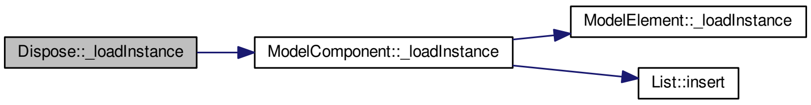
Here is the call graph for this function:


Dispose::_initBetweenReplications
SinkModelComponent ::_initBetweenReplications

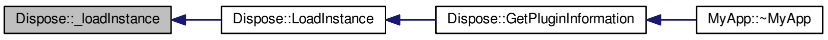
5.17.2.4 bool Dispose::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from SinkModelComponent.

Here is the call graph for this function:



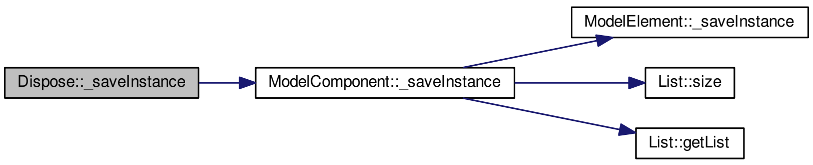
Here is the caller graph for this function:



5.17.2.5 std::map< std::string, std::string > ∗ Dispose::_saveInstance ( ) [protected], [virtual]

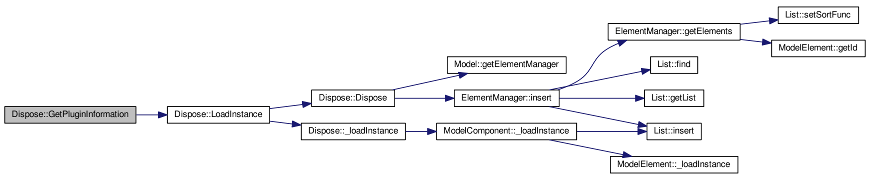
Reimplemented from SinkModelComponent.

Here is the call graph for this function:

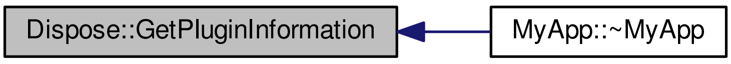


5.17.2.6 PluginInformation ∗ Dispose::GetPluginInformation ( ) [static]

Here is the call graph for this function:

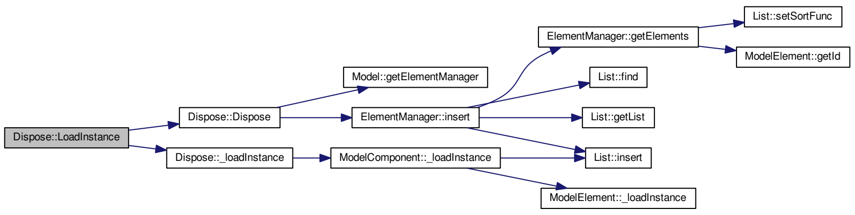


Here is the caller graph for this function:

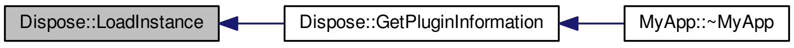


5.17.2.7 ModelComponent $*$ Dispose::LoadInstance (Model $*$ *model*, std::map$<$ std::string, std::string $>$ $*$ *fields*) [static]

Here is the call graph for this function:



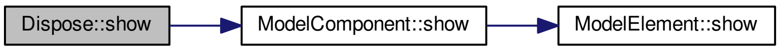
Here is the caller graph for this function:



5.17.2.8 std::string Dispose::show ( ) [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



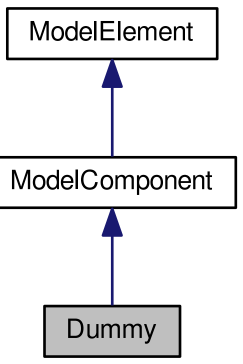
The documentation for this class was generated from the following files:

- Dispose.h
- Dispose.cpp

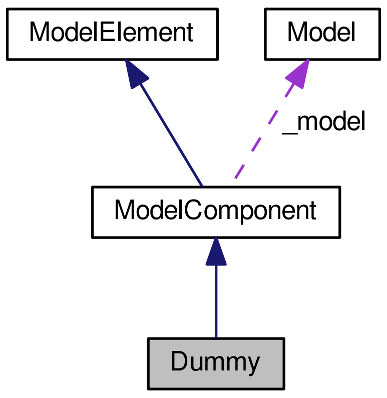
5.18 Dummy Class Reference

```
#include <Dummy.h>
```

Inheritance diagram for Dummy:



Collaboration diagram for Dummy:



Public Member Functions

- Dummy (Model ∗model)
- Dummy (const Dummy &orig)
- virtual ∼Dummy ()
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

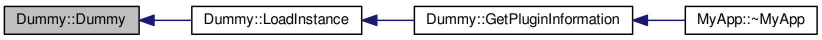
- virtual void _execute (Entity ∗entity)
- virtual void _initBetweenReplications ()
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.18.1 Constructor & Destructor Documentation

5.18.1.1 Dummy::Dummy (Model ∗ *model*)

Here is the caller graph for this function:



5.18.1.2 Dummy::Dummy (const Dummy & *orig*)

5.18.1.3 Dummy::∼Dummy ( ) [virtual]

5.18.2 Member Function Documentation

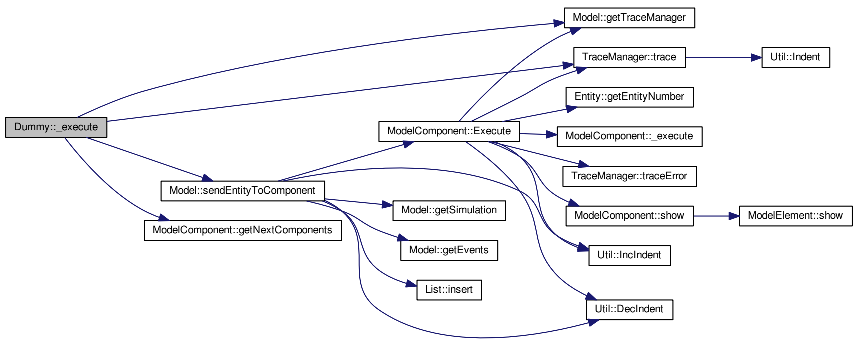
5.18.2.1 bool Dummy::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

5.18.2.2 void Dummy::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

Here is the call graph for this function:



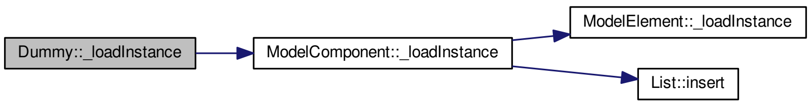
5.18.2.3 void Dummy::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

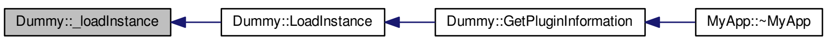
5.18.2.4 bool Dummy::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



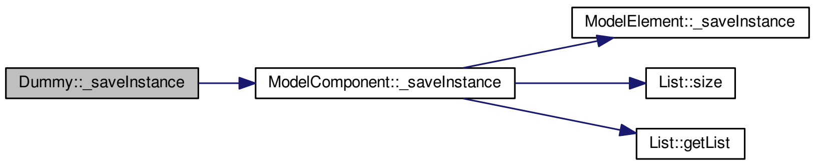
Here is the caller graph for this function:



5.18.2.5 std::map< std::string, std::string > ∗ Dummy::_saveInstance ( ) [protected], [virtual]

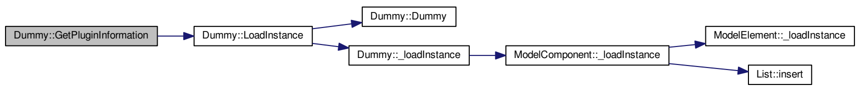
Reimplemented from ModelComponent.

Here is the call graph for this function:

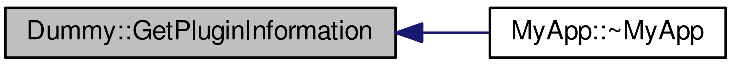


5.18.2.6 PluginInformation ∗ Dummy::GetPluginInformation ( ) [static]

Here is the call graph for this function:

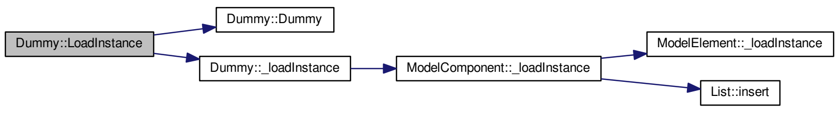

Dummy::GetPluginInformation
Dummy::LoadInstance
Dummy::Dummy
Dummy::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
List::insert

Here is the caller graph for this function:

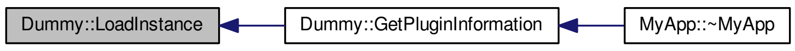


5.18.2.7 ModelComponent ∗ Dummy::LoadInstance (Model ∗ *model,* std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:



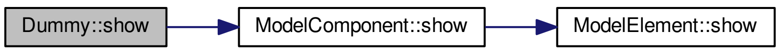
Here is the caller graph for this function:



5.18.2.8 std::string Dummy::show ( ) `[virtual]`

Reimplemented from ModelComponent.

Here is the call graph for this function:



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

- Dummy.h
- Dummy.cpp

5.19 ElementManager Class Reference

`#include <ElementManager.h>`

Public Member Functions

- ElementManager (Model ∗model)
- ElementManager (const ElementManager &orig)
- virtual ∼ElementManager ()
- bool insert (std::string infraTypename, ModelElement ∗infra)
- void remove (std::string infraTypename, ModelElement ∗infra)
- bool check (std::string infraTypename, ModelElement ∗infra, std::string expressionName, std::string ∗error←Message)
- bool check (std::string infraTypename, std::string infraName, std::string expressionName, bool mandatory, std::string ∗errorMessage)
- void clear ()
- ModelElement ∗ getElement (std::string infraTypename, Util::identitifcation id)
- ModelElement ∗ getElement (std::string infraTypename, std::string name)
- unsigned int getNumberOfElements (std::string infraTypename)
- int getRankOf (std::string infraTypename, std::string name)

    returns the position (1st position=0) of the element if found, or negative value if not found
- std::list< std::string > ∗ getElementTypenames () const
- List< ModelElement ∗ > ∗ getElements (std::string infraTypename) const
- void show ()
- Model ∗ getModel () const

5.19.1 Detailed Description

The ElementManager is responsible for inserting and removing elements (ModelElement) used by components, in a consistent way. TO FIX: No direct access for insertion or deletion should be allow

5.19.2 Constructor & Destructor Documentation

5.19.2.1 ElementManager::ElementManager (**Model** ∗ *model*)

Elements are organized as a map from a string (key), the type of an element, and a list of elements of that type

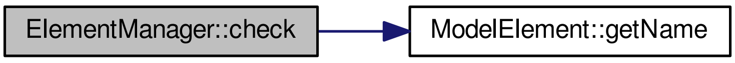
5.19.2.2 ElementManager::ElementManager (const **ElementManager** & *orig*)

5.19.2.3 ElementManager::∼ElementManager ( ) `[virtual]`

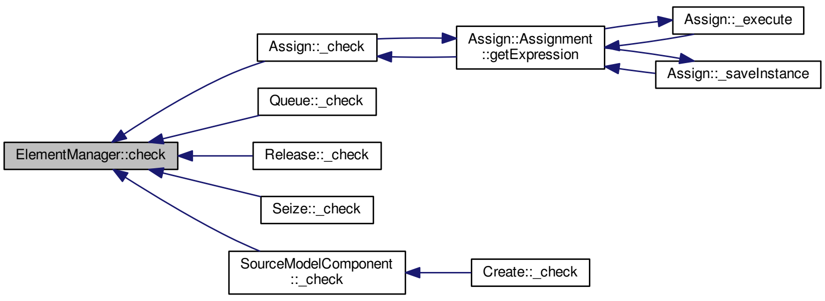
5.19.3 Member Function Documentation

5.19.3.1 bool ElementManager::check (std::string *infraTypename,* **ModelElement** ∗ *infra,* std::string *expressionName,* std::string ∗ *errorMessage*)

Here is the call graph for this function:

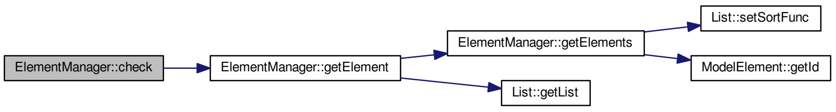


Here is the caller graph for this function:



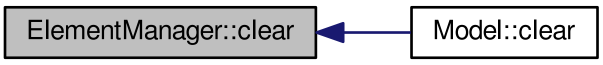
5.19.3.2 bool ElementManager::check (std::string *infraTypename,* std::string *infraName,* std::string *expressionName,* bool *mandatory,* std::string ∗ *errorMessage*)

Here is the call graph for this function:



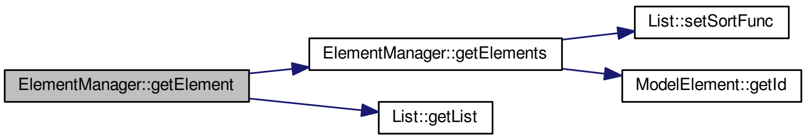
5.19.3.3 void ElementManager::clear ( )

Here is the caller graph for this function:

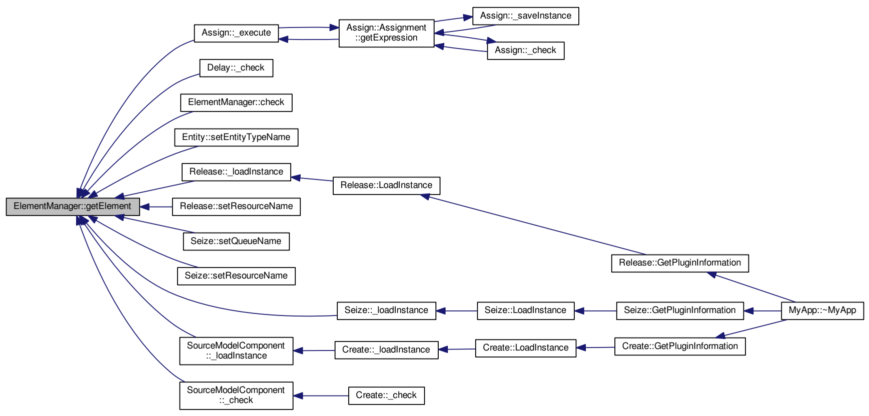


5.19.3.4 ModelElement ∗ ElementManager::getElement (std::string *infraTypename,* Util::identitifcation *id*)

Here is the call graph for this function:

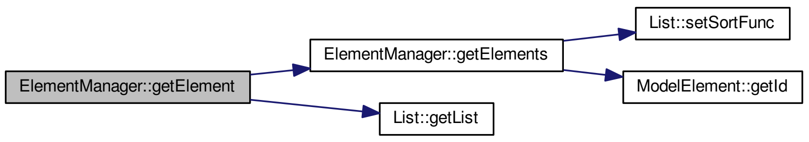


Here is the caller graph for this function:



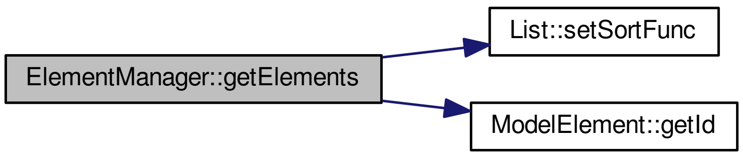
5.19.3.5 **ModelElement** $*$ **ElementManager::getElement (std::string *infraTypename,* std::string *name*)**

Here is the call graph for this function:

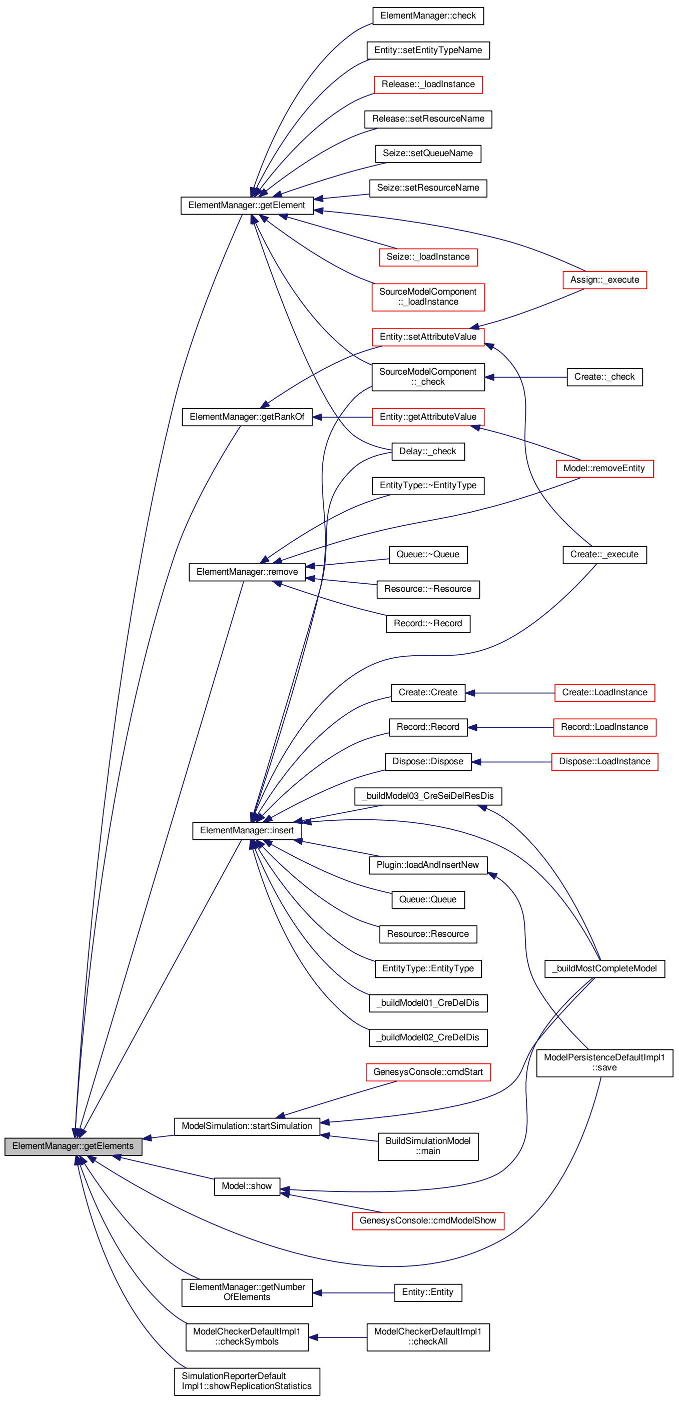


5.19.3.6 **List**$<$ **ModelElement** $* > *$ **ElementManager::getElements (std::string *infraTypename*) const**

Here is the call graph for this function:

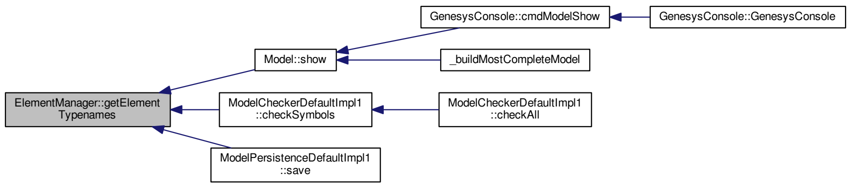


Here is the caller graph for this function:



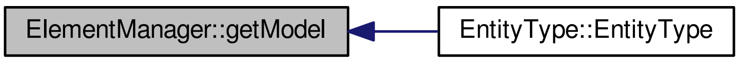
5.19.3.7 std::list< std::string > ∗ ElementManager::getElementTypenames ( ) const

Here is the caller graph for this function:


ElementManager::getElement Typenames
Model::show
GenesysConsole::cmdModelShow
GenesysConsole::GenesysConsole
_buildMostCompleteModel
ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkAll
ModelPersistenceDefaultImpl1 ::save

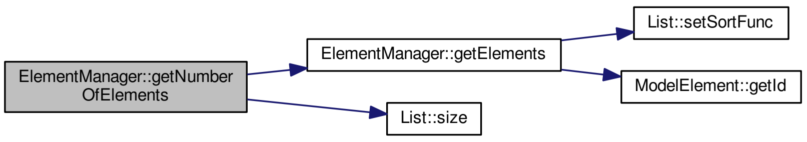
5.19.3.8 Model ∗ ElementManager::getModel ( ) const

Here is the caller graph for this function:

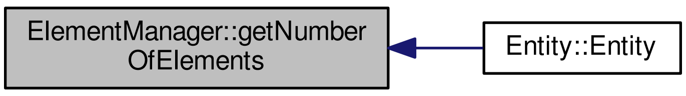


5.19.3.9 unsigned int ElementManager::getNumberOfElements (std::string *infraTypename*)

Here is the call graph for this function:


ElementManager::getNumber OfElements
ElementManager::getElements
List::setSortFunc
ModelElement::getId
List::size

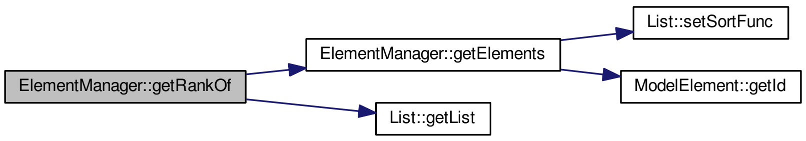
Here is the caller graph for this function:



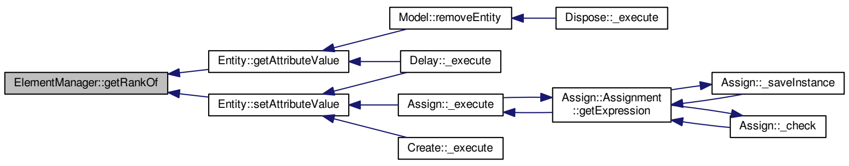
5.19.3.10 int ElementManager::getRankOf (std::string *infraTypename,* std::string *name*)

returns the position (1st position=0) of the element if found, or negative value if not found

Here is the call graph for this function:

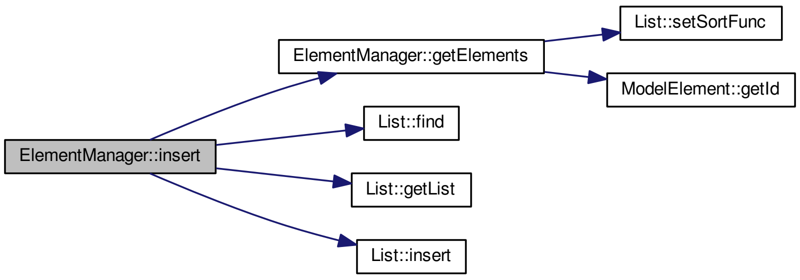


Here is the caller graph for this function:

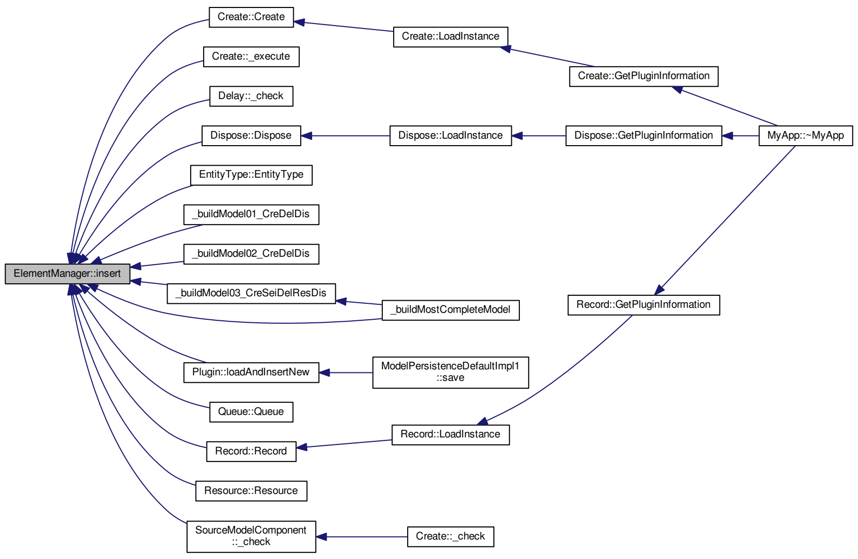


5.19.3.11 bool ElementManager::insert (std::string *infraTypename,* ModelElement ∗ *infra*)

Here is the call graph for this function:

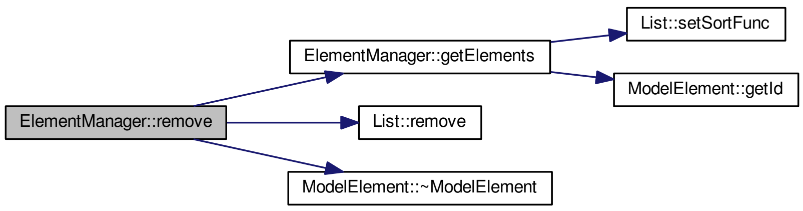


Here is the caller graph for this function:

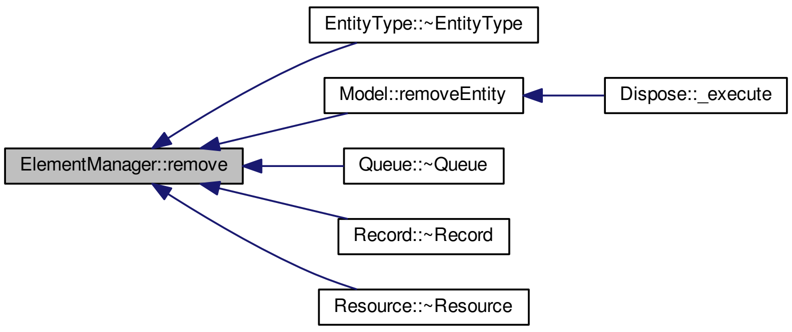


5.19.3.12 void ElementManager::remove (std::string *infraTypename,* **ModelElement** ∗ *infra*)

Here is the call graph for this function:

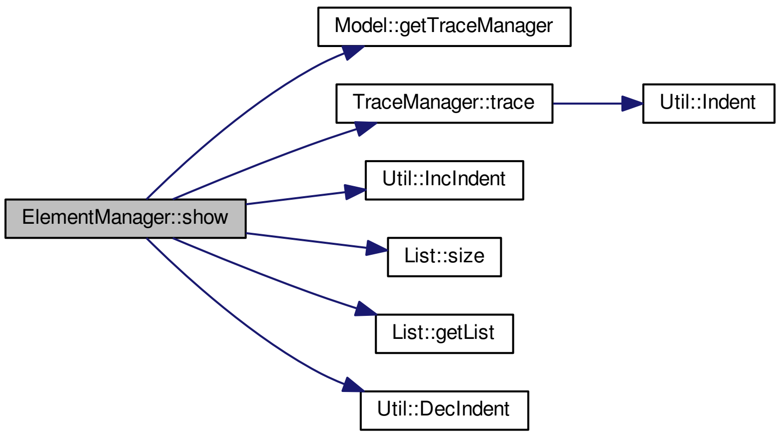


Here is the caller graph for this function:



5.19.3.13 void ElementManager::show ( )

Here is the call graph for this function:



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

- ElementManager.h
- ElementManager.cpp

5.20 ElementManager_if Class Reference

```
#include <ElementManager_if.h>
```

Public Member Functions

- ElementManager_if ()
- ElementManager_if (const ElementManager_if &orig)
- virtual ∼ElementManager_if ()

5.20.1 Constructor & Destructor Documentation

5.20.1.1 ElementManager_if::ElementManager_if ( )

5.20.1.2 ElementManager_if::ElementManager_if (const ElementManager_if & *orig*)

5.20.1.3 virtual ElementManager_if::∼ElementManager_if ( ) `[virtual]`

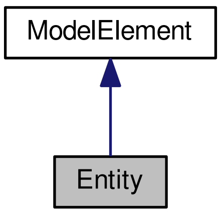
The documentation for this class was generated from the following file:

- ElementManager_if.h

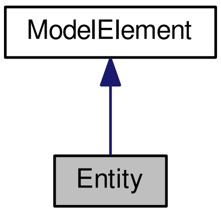
5.21 Entity Class Reference

```
#include <Entity.h>
```

Inheritance diagram for Entity:



Collaboration diagram for Entity:



Public Member Functions

- Entity (ElementManager ∗elements)
- Entity (const Entity &orig)
- virtual ∼Entity ()
- virtual std::string show ()
- void setEntityTypeName (std::string entityTypeName) throw ()
- std::string getEntityTypeName () const
- void setEntityType (EntityType ∗entityType)
- EntityType ∗ getEntityType () const
- double getAttributeValue (std::string attributeName)
- void setAttributeValue (std::string attributeName, double value)
- Util::identitifcation getEntityNumber () const

Protected Member Functions

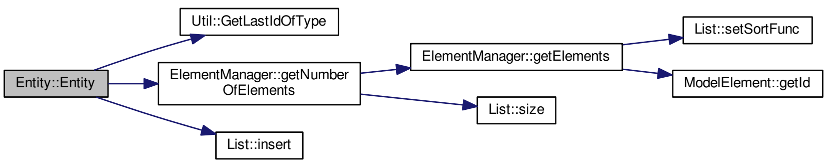
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.21.1 Constructor & Destructor Documentation

5.21.1.1 Entity::Entity (ElementManager ∗ *elements*)

Here is the call graph for this function:



5.21.1.2 Entity::Entity (const Entity & *orig*)

5.21.1.3 Entity::∼Entity ( ) [virtual]

5.21.2 Member Function Documentation

5.21.2.1 bool Entity::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

5.21.2.2 bool Entity::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

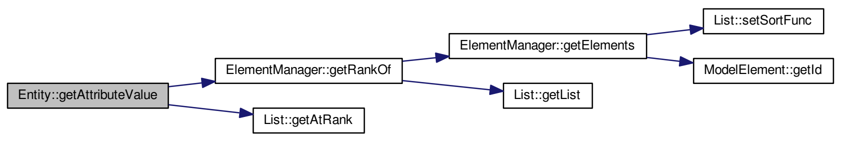
Reimplemented from ModelElement.

5.21.2.3 std::map< std::string, std::string > ∗ Entity::_saveInstance ( ) [protected], [virtual]

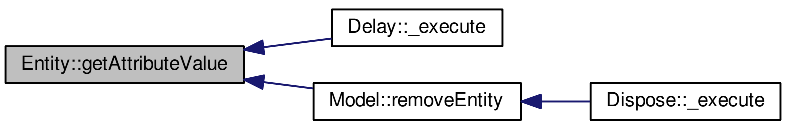
Reimplemented from ModelElement.

5.21.2.4 double Entity::getAttributeValue (std::string *attributeName*)

Here is the call graph for this function:

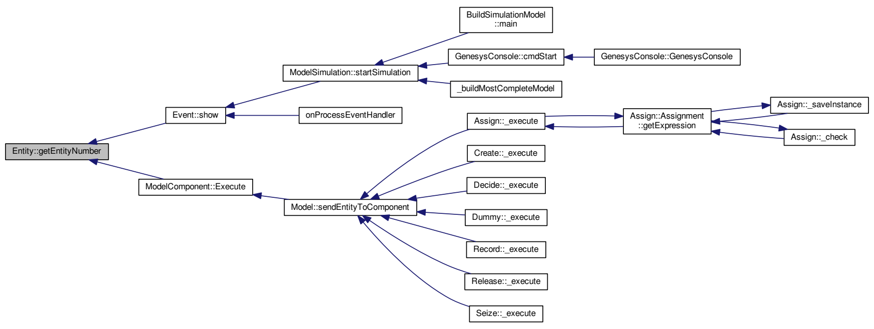

Entity::getAttributeValue
ElementManager::getRankOf
List::getAtRank
ElementManager::getElements
List::getList
List::setSortFunc
ModelElement::getId

Here is the caller graph for this function:



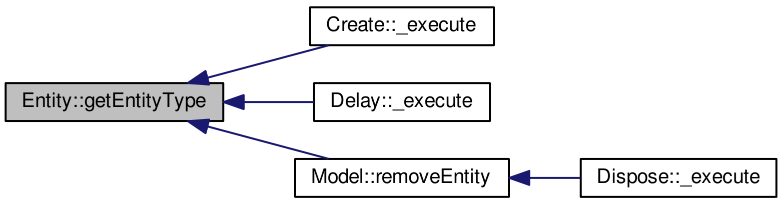
5.21.2.5 Util::identitifcation Entity::getEntityNumber ( ) const

Here is the caller graph for this function:



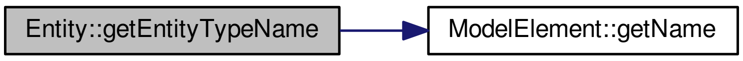
5.21.2.6 EntityType ∗ Entity::getEntityType ( ) const

Here is the caller graph for this function:



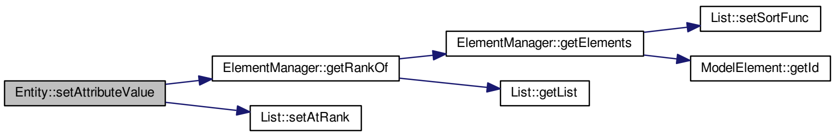
5.21.2.7 std::string Entity::getEntityTypeName ( ) const

Here is the call graph for this function:

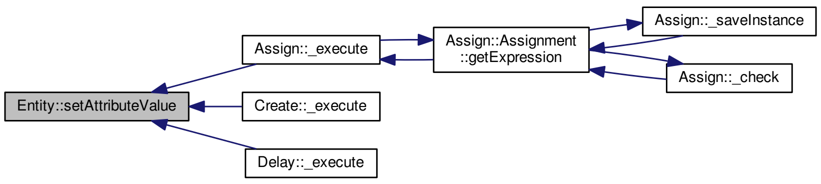


5.21.2.8 void Entity::setAttributeValue (std::string *attributeName,* double *value*)

Here is the call graph for this function:

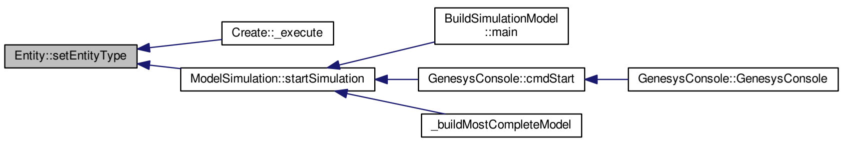

List::setSortFunc
ElementManager::getElements
ElementManager::getRankOf
ModelElement::getId
Entity::setAttributeValue
List::getList
List::setAtRank

Here is the caller graph for this function:



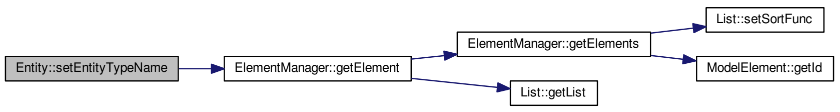
5.21.2.9 void Entity::setEntityType (EntityType ∗ *entityType*)

Here is the caller graph for this function:


Entity::setEntityType
Create::_execute
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.21.2.10 void Entity::setEntityTypeName (std::string *entityTypeName*) throw)

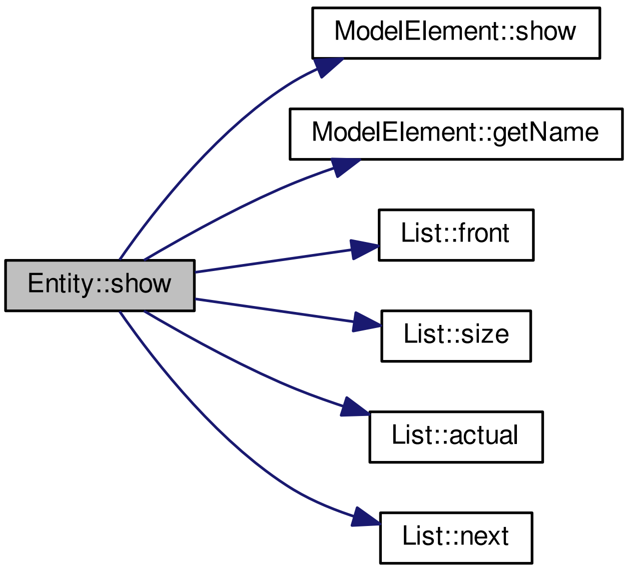
Here is the call graph for this function:


Entity::setEntityTypeName
ElementManager::getElement
ElementManager::getElements
List::getList
List::setSortFunc
ModelElement::getId

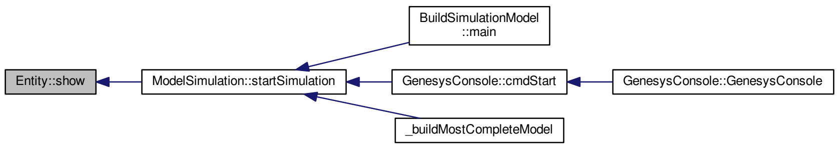
5.21.2.11 std::string Entity::show ( ) `[virtual]`

Reimplemented from ModelElement.

Here is the call graph for this function:



Here is the caller graph for this function:


BuildSimulationModel ::main
Entity::show
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

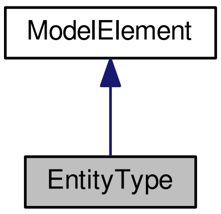
The documentation for this class was generated from the following files:

- Entity.h
- Entity.cpp

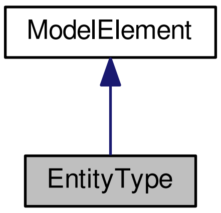
5.22 EntityType Class Reference

```
#include <EntityType.h>
```

Inheritance diagram for EntityType:



Collaboration diagram for EntityType:



Public Member Functions

- EntityType (ElementManager ∗elemManager)
- EntityType (ElementManager ∗elemManager, std::string name)
- EntityType (const EntityType &orig)
- virtual ∼EntityType ()
- virtual std::string show ()
- void setInitialWaitingCost (double _initialWaitingCost)
- double getInitialWaitingCost () const
- void setInitialOtherCost (double _initialOtherCost)
- double getInitialOtherCost () const
- void setInitialNVACost (double _initialNVACost)
- double getInitialNVACost () const
- void setInitialVACost (double _initialVACost)
- double getInitialVACost () const
- void setInitialPicture (std::string _initialPicture)
- std::string getInitialPicture () const
- StatisticsCollector ∗ getCstatTotalTime () const
- StatisticsCollector ∗ getCstatNVATime () const
- StatisticsCollector ∗ getCstatVATime () const
- StatisticsCollector ∗ getCstatOtherTime () const
- StatisticsCollector ∗ getCstatTransferTime () const
- StatisticsCollector ∗ getCstatWaitingTime () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

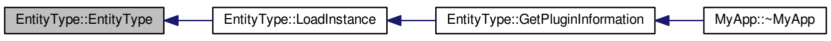
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.22.1 Constructor & Destructor Documentation

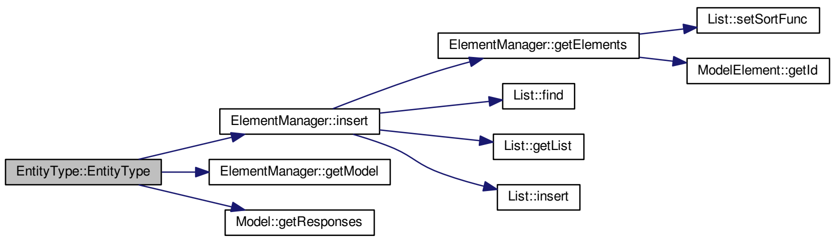
5.22.1.1 EntityType::EntityType (**ElementManager** ∗ ***elemManager***)

Here is the caller graph for this function:



5.22.1.2 EntityType::EntityType (ElementManager ∗ *elemManager,* std::string *name*)

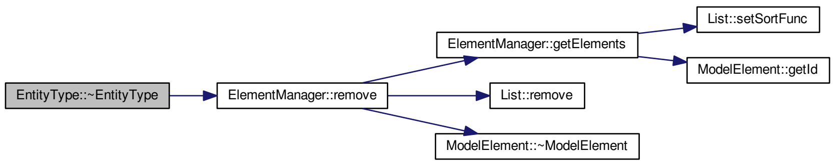
Here is the call graph for this function:



5.22.1.3 EntityType::EntityType (const EntityType & *orig*)

5.22.1.4 EntityType::∼EntityType ( ) [virtual]

Here is the call graph for this function:



5.22.2 Member Function Documentation

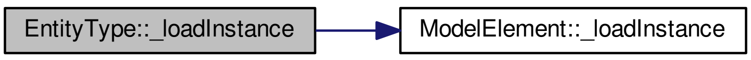
5.22.2.1 bool EntityType::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

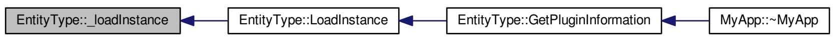
5.22.2.2 bool EntityType::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



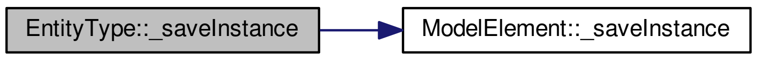
Here is the caller graph for this function:



5.22.2.3 std::map< std::string, std::string > * EntityType::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:

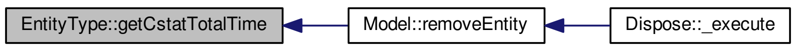


5.22.2.4 StatisticsCollector * EntityType::getCstatNVATime ( ) const

5.22.2.5 StatisticsCollector * EntityType::getCstatOtherTime ( ) const

5.22.2.6 StatisticsCollector * EntityType::getCstatTotalTime ( ) const

Here is the caller graph for this function:

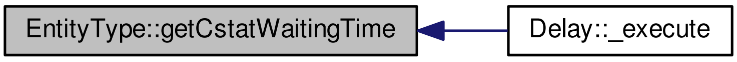


5.22.2.7 StatisticsCollector * EntityType::getCstatTransferTime ( ) const

5.22.2.8 StatisticsCollector * EntityType::getCstatVATime ( ) const

5.22.2.9 StatisticsCollector ∗ EntityType::getCstatWaitingTime ( ) const

Here is the caller graph for this function:



5.22.2.10 double EntityType::getInitialNVACost ( ) const

5.22.2.11 double EntityType::getInitialOtherCost ( ) const

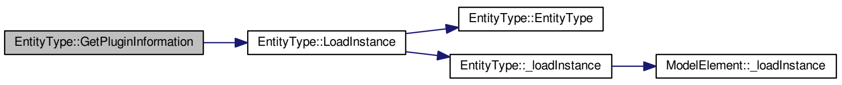
5.22.2.12 std::string EntityType::getInitialPicture ( ) const

5.22.2.13 double EntityType::getInitialVACost ( ) const

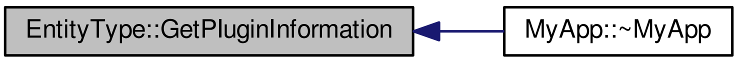
5.22.2.14 double EntityType::getInitialWaitingCost ( ) const

5.22.2.15 PluginInformation ∗ EntityType::GetPluginInformation ( ) [static]

Here is the call graph for this function:

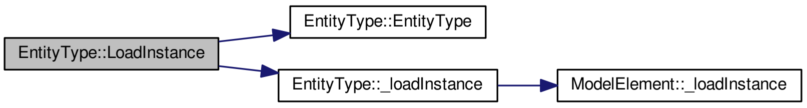

EntityType::GetPluginInformation
EntityType::LoadInstance
EntityType::EntityType
EntityType::_loadInstance
ModelElement::_loadInstance

Here is the caller graph for this function:

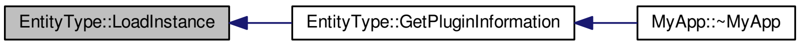


5.22.2.16 **ModelElement** ∗ EntityType::LoadInstance (**ElementManager** ∗ *elems*, std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:



Here is the caller graph for this function:



5.22.2.17 void EntityType::setInitialNVACost (double *_initialNVACost*)

5.22.2.18 void EntityType::setInitialOtherCost (double *_initialOtherCost*)

5.22.2.19 void EntityType::setInitialPicture (std::string *_initialPicture*)

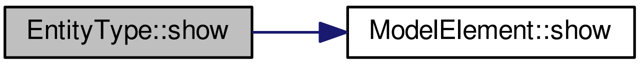
5.22.2.20 void EntityType::setInitialVACost (double *_initialVACost*)

5.22.2.21 void EntityType::setInitialWaitingCost (double *_initialWaitingCost*)

5.22.2.22 std::string EntityType::show ( ) [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



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

- EntityType.h
- EntityType.cpp

5.23 Event Class Reference

```
#include <Event.h>
```

Public Member Functions

- Event (double time, Entity ∗entity, ModelComponent ∗component)
- Event (const Event &orig)
- virtual ∼Event ()
- double getTime () const
- ModelComponent ∗ getComponent () const
- Entity ∗ getEntity () const
- std::string show ()

5.23.1 Constructor & Destructor Documentation

5.23.1.1 Event::Event (double *time*, Entity ∗ *entity*, ModelComponent ∗ *component*)

5.23.1.2 Event::Event (const Event & *orig*)

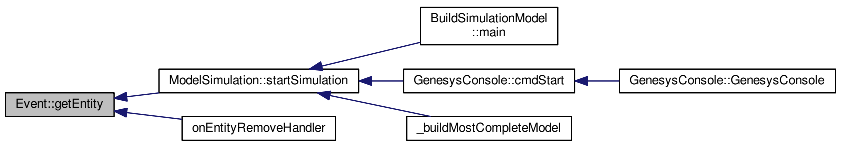
5.23.1.3 Event::∼Event ( ) [virtual]

5.23.2 Member Function Documentation

5.23.2.1 ModelComponent ∗ Event::getComponent ( ) const

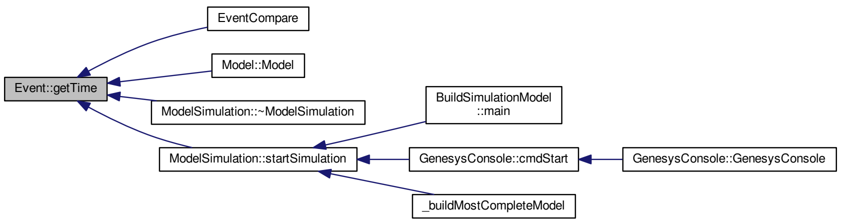
5.23.2.2 Entity ∗ Event::getEntity ( ) const

Here is the caller graph for this function:


BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
Event::getEntity
onEntityRemoveHandler
_buildMostCompleteModel

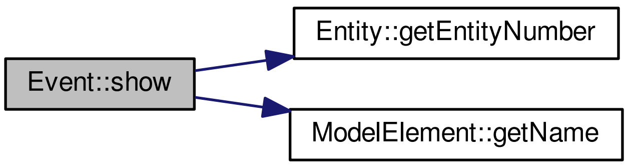
5.23.2.3 double Event::getTime ( ) const

Here is the caller graph for this function:

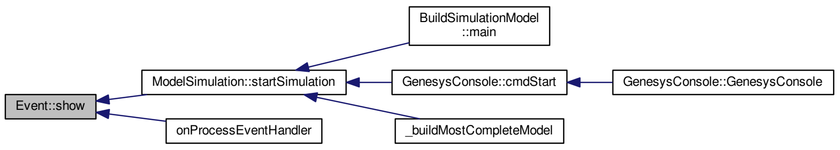

EventCompare
Model::Model
Event::getTime
ModelSimulation::~ModelSimulation
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.23.2.4 std::string Event::show ( )

Here is the call graph for this function:



Here is the caller graph for this function:


Event::show
ModelSimulation::startSimulation
onProcessEventHandler
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

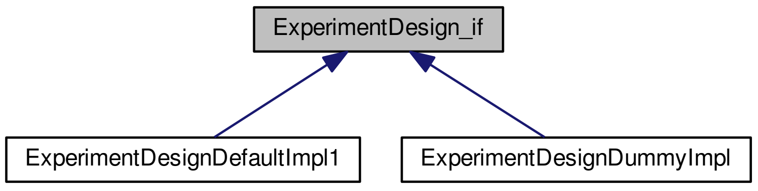
The documentation for this class was generated from the following files:

- Event.h
- Event.cpp

5.24 ExperimentDesign_if Class Reference

```
#include <ExperimentDesign_if.h>
```

Inheritance diagram for ExperimentDesign_if:



Public Member Functions

- virtual ProcessAnalyser_if * getProcessAnalyser () const =0
- virtual bool generate2krScenarioExperiments ()=0
- virtual bool calculateContributionAndCoefficients ()=0
- virtual std::list< FactorOrInteractionContribution * > * getContributions () const =0

5.24.1 Detailed Description

It designs a set of experiments (SimulationScenario) where que level of factors (SimulationControl) are set automatically to create a 2^k.r experiment design, and where the contributions of the factors and their interactions (just a set of SimulationControl) can be obtained.

5.24.2 Member Function Documentation

5.24.2.1 virtual bool ExperimentDesign_if::calculateContributionAndCoefficients ( ) `[pure virtual]`

Implemented in ExperimentDesignDummyImpl, and ExperimentDesignDefaultImpl1.

5.24.2.2 virtual bool ExperimentDesign_if::generate2krScenarioExperiments ( ) `[pure virtual]`

Implemented in ExperimentDesignDummyImpl, and ExperimentDesignDefaultImpl1.

5.24.2.3 virtual std::list<FactorOrInteractionContribution*>* ExperimentDesign_if::getContributions ( ) const `[pure virtual]`

Implemented in ExperimentDesignDummyImpl, and ExperimentDesignDefaultImpl1.

5.24.2.4 virtual ProcessAnalyser_if* ExperimentDesign_if::getProcessAnalyser ( ) const `[pure virtual]`

Implemented in ExperimentDesignDummyImpl, and ExperimentDesignDefaultImpl1.

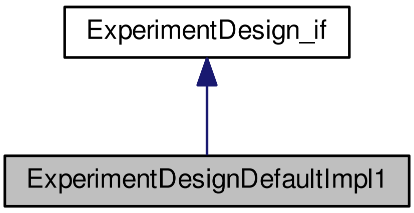
The documentation for this class was generated from the following file:

- ExperimentDesign_if.h

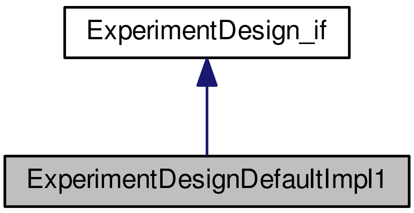
5.25 ExperimentDesignDefaultImpl1 Class Reference

```
#include <ExperimentDesignDefaultImpl1.h>
```

Inheritance diagram for ExperimentDesignDefaultImpl1:



Collaboration diagram for ExperimentDesignDefaultImpl1:



Public Member Functions

- ExperimentDesignDefaultImpl1 ()
- ExperimentDesignDefaultImpl1 (const ExperimentDesignDefaultImpl1 &orig)
- virtual ∼ExperimentDesignDefaultImpl1 ()
- virtual ProcessAnalyser_if ∗ getProcessAnalyser () const
- virtual bool generate2krScenarioExperiments ()
- virtual bool calculateContributionAndCoefficients ()
- virtual std::list< FactorOrInteractionContribution ∗ > ∗ getContributions () const

5.25.1 Constructor & Destructor Documentation

5.25.1.1 ExperimentDesignDefaultImpl1::ExperimentDesignDefaultImpl1 ( )

5.25.1.2 ExperimentDesignDefaultImpl1::ExperimentDesignDefaultImpl1 (const ExperimentDesignDefaultImpl1 & *orig*)

5.25.1.3 ExperimentDesignDefaultImpl1::∼ExperimentDesignDefaultImpl1 ( ) [virtual]

5.25.2 Member Function Documentation

5.25.2.1 bool ExperimentDesignDefaultImpl1::calculateContributionAndCoefficients ( ) [virtual]

Implements ExperimentDesign_if.

5.25.2.2 bool ExperimentDesignDefaultImpl1::generate2krScenarioExperiments ( ) [virtual]

Implements ExperimentDesign_if.

5.25.2.3 std::list< FactorOrInteractionContribution ∗ > ∗ ExperimentDesignDefaultImpl1::getContributions ( ) const [virtual]

Implements ExperimentDesign_if.

5.25.2.4 ProcessAnalyser_if ∗ ExperimentDesignDefaultImpl1::getProcessAnalyser ( ) const [virtual]

Implements ExperimentDesign_if.

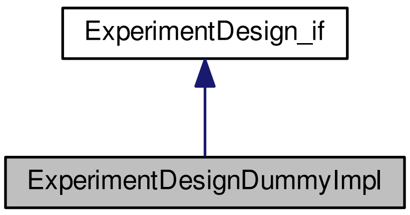
The documentation for this class was generated from the following files:

- ExperimentDesignDefaultImpl1.h
- ExperimentDesignDefaultImpl1.cpp

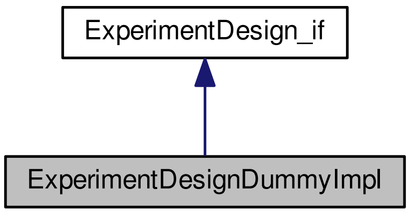
5.26 ExperimentDesignDummyImpl Class Reference

```
#include <ExperimentDesignDummyImpl.h>
```

Inheritance diagram for ExperimentDesignDummyImpl:



Collaboration diagram for ExperimentDesignDummyImpl:



Public Member Functions

- ExperimentDesignDummyImpl ()
- ExperimentDesignDummyImpl (const ExperimentDesignDummyImpl &orig)
- virtual ∼ExperimentDesignDummyImpl ()
- virtual ProcessAnalyser_if $*$ getProcessAnalyser () const
- virtual bool generate2krScenarioExperiments ()
- virtual bool calculateContributionAndCoefficients ()
- virtual std::list$<$ FactorOrInteractionContribution $*$ $>$ $*$ getContributions () const

5.26.1 Constructor & Destructor Documentation

5.26.1.1 ExperimentDesignDummyImpl::ExperimentDesignDummyImpl ( )

5.26.1.2 ExperimentDesignDummyImpl::ExperimentDesignDummyImpl (const ExperimentDesignDummyImpl & *orig*)

5.26.1.3 ExperimentDesignDummyImpl::∼ExperimentDesignDummyImpl ( ) `[virtual]`

5.26.2 Member Function Documentation

5.26.2.1 bool ExperimentDesignDummyImpl::calculateContributionAndCoefficients ( ) `[virtual]`

Implements ExperimentDesign_if.

5.26.2.2 bool ExperimentDesignDummyImpl::generate2krScenarioExperiments ( ) `[virtual]`

Implements ExperimentDesign_if.

5.26.2.3 std::list< FactorOrInteractionContribution ∗ > ∗ ExperimentDesignDummyImpl::getContributions ( ) const `[virtual]`

Implements ExperimentDesign_if.

5.26.2.4 ProcessAnalyser_if ∗ ExperimentDesignDummyImpl::getProcessAnalyser ( ) const `[virtual]`

Implements ExperimentDesign_if.

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

- ExperimentDesignDummyImpl.h
- ExperimentDesignDummyImpl.cpp

5.27 FactorOrInteractionContribution Class Reference

```
#include <FactorOrInteractionContribution.h>
```

Public Member Functions

- FactorOrInteractionContribution (double contribution, double modelCoefficient, std::list< SimulationControl ∗ > ∗controls)
- FactorOrInteractionContribution (const FactorOrInteractionContribution &orig)
- ∼FactorOrInteractionContribution ()
- double getModelCoefficient () const
- std::list< SimulationControl ∗ > ∗ getControls () const
- double getContribution () const

5.27.1 Detailed Description

This simple class corresponds to a factor when it refers to just one SimulationControl, or to the interaction between two or more factors when it refers to more SimulationControl. It also encapsulates the contribution of the factor or interaction and its coefficient in the full model that estimates one specific SimulationResponse.

5.27.2 Constructor & Destructor Documentation

5.27.2.1 FactorOrInteractionContribution::FactorOrInteractionContribution (double *contribution,* double *modelCoefficient,* std::list< **SimulationControl** $*$ > $*$ *controls*)

5.27.2.2 FactorOrInteractionContribution::FactorOrInteractionContribution (const **FactorOrInteractionContribution** & *orig*)

5.27.2.3 FactorOrInteractionContribution::∼FactorOrInteractionContribution ( )

5.27.3 Member Function Documentation

5.27.3.1 double FactorOrInteractionContribution::getContribution ( ) const

5.27.3.2 std::list< **SimulationControl** $*$ > $*$ FactorOrInteractionContribution::getControls ( ) const

5.27.3.3 double FactorOrInteractionContribution::getModelCoefficient ( ) const

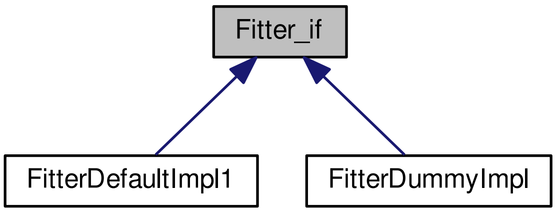
The documentation for this class was generated from the following files:

- FactorOrInteractionContribution.h
- FactorOrInteractionContribution.cpp

5.28 Fitter_if Class Reference

```
#include <Fitter_if.h>
```

Inheritance diagram for Fitter_if:



Public Member Functions

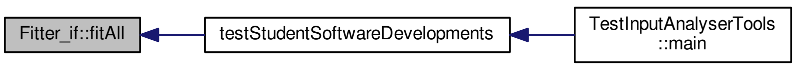
- virtual bool isNormalDistributed (double confidencelevel)=0
- virtual void fitUniform (double ∗sqrerror, double ∗min, double ∗max)=0
- virtual void fitTriangular (double ∗sqrerror, double ∗min, double ∗mo, double ∗max)=0
- virtual void fitNormal (double ∗sqrerror, double ∗avg, double ∗stddev)=0
- virtual void fitExpo (double ∗sqrerror, double ∗avg1)=0
- virtual void fitErlang (double ∗sqrerror, double ∗avg, double ∗m)=0
- virtual void fitBeta (double ∗sqrerror, double ∗alpha, double ∗beta, double ∗infLimit, double ∗supLimit)=0
- virtual void fitWeibull (double ∗sqrerror, double ∗alpha, double ∗scale)=0
- virtual void fitAll (double ∗sqrerror, std::string ∗name)=0
- virtual void setDataFilename (std::string dataFilename)=0
- virtual std::string getDataFilename ()=0

5.28.1 Member Function Documentation

5.28.1.1 virtual void Fitter_if::fitAll (double ∗ *sqrerror,* std::string ∗ *name*) `[pure virtual]`

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

Here is the caller graph for this function:


Fitter_if::fitAll
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.28.1.2 virtual void Fitter_if::fitBeta (double ∗ *sqrerror,* double ∗ *alpha,* double ∗ *beta,* double ∗ *infLimit,* double ∗ *supLimit*) `[pure virtual]`

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

5.28.1.3 virtual void Fitter_if::fitErlang (double ∗ *sqrerror,* double ∗ *avg,* double ∗ *m*) `[pure virtual]`

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

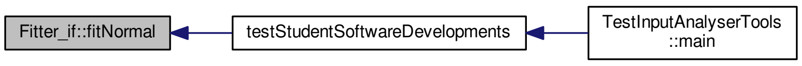
5.28.1.4 virtual void Fitter_if::fitExpo (double ∗ *sqrerror,* double ∗ *avg1*) `[pure virtual]`

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

5.28.1.5 virtual void Fitter_if::fitNormal (double ∗ *sqrerror,* double ∗ *avg,* double ∗ *stddev*) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

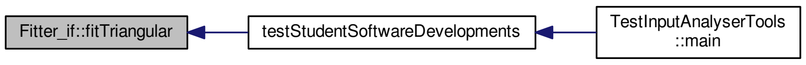
Here is the caller graph for this function:


Fitter_if::fitNormal
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.28.1.6 virtual void Fitter_if::fitTriangular (double ∗ *sqrerror,* double ∗ *min,* double ∗ *mo,* double ∗ *max*) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

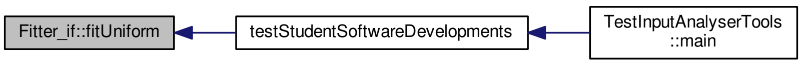
Here is the caller graph for this function:


Fitter_if::fitTriangular
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.28.1.7 virtual void Fitter_if::fitUniform (double ∗ *sqrerror,* double ∗ *min,* double ∗ *max*) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

Here is the caller graph for this function:


Fitter_if::fitUniform
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.28.1.8 virtual void Fitter_if::fitWeibull (double ∗ *sqrerror,* double ∗ *alpha,* double ∗ *scale*) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

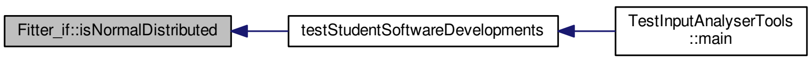
5.28.1.9 virtual std::string Fitter_if::getDataFilename ( ) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

5.28.1.10 virtual bool Fitter_if::isNormalDistributed (double *confidencelevel*) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

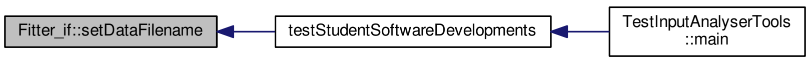
Here is the caller graph for this function:


Fitter_if::isNormalDistributed
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.28.1.11 virtual void Fitter_if::setDataFilename (std::string *dataFilename*) [pure virtual]

Implemented in FitterDefaultImpl1, and FitterDummyImpl.

Here is the caller graph for this function:


Fitter_if::setDataFilename
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

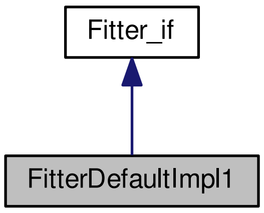
The documentation for this class was generated from the following file:

- Fitter_if.h

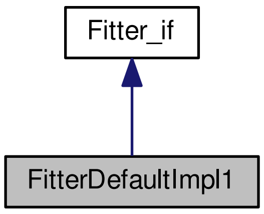
5.29 FitterDefaultImpl1 Class Reference

```
#include <FitterDefaultImpl1.h>
```

Inheritance diagram for FitterDefaultImpl1:



Collaboration diagram for FitterDefaultImpl1:



Public Member Functions

- FitterDefaultImpl1 ()
- FitterDefaultImpl1 (const FitterDefaultImpl1 &orig)
- virtual ∼FitterDefaultImpl1 ()
- bool isNormalDistributed (double confidencelevel)
- void fitUniform (double ∗sqrerror, double ∗min, double ∗max)
- void fitTriangular (double ∗sqrerror, double ∗min, double ∗mo, double ∗max)
- void fitNormal (double ∗sqrerror, double ∗avg, double ∗stddev)
- void fitExpo (double ∗sqrerror, double ∗avg1)
- void fitErlang (double ∗sqrerror, double ∗avg, double ∗m)
- void fitBeta (double ∗sqrerror, double ∗alpha, double ∗beta, double ∗infLimit, double ∗supLimit)
- void fitWeibull (double ∗sqrerror, double ∗alpha, double ∗scale)
- void fitAll (double ∗sqrerror, std::string ∗name)
- void setDataFilename (std::string dataFilename)
- std::string getDataFilename ()

5.29.1 Constructor & Destructor Documentation

5.29.1.1 FitterDefaultImpl1::FitterDefaultImpl1 ( )

5.29.1.2 FitterDefaultImpl1::FitterDefaultImpl1 (const FitterDefaultImpl1 & *orig*)

5.29.1.3 FitterDefaultImpl1::∼FitterDefaultImpl1 ( ) `[virtual]`

5.29.2 Member Function Documentation

5.29.2.1 void FitterDefaultImpl1::fitAll (double ∗ *sqrerror,* std::string ∗ *name*) `[virtual]`

Implements Fitter_if.

5.29.2.2 void FitterDefaultImpl1::fitBeta (double ∗ *sqrerror,* double ∗ *alpha,* double ∗ *beta,* double ∗ *infLimit,* double ∗ *supLimit*) [virtual]

Implements Fitter_if.

5.29.2.3 void FitterDefaultImpl1::fitErlang (double ∗ *sqrerror,* double ∗ *avg,* double ∗ *m*) [virtual]

Implements Fitter_if.

5.29.2.4 void FitterDefaultImpl1::fitExpo (double ∗ *sqrerror,* double ∗ *avg1*) [virtual]

Implements Fitter_if.

5.29.2.5 void FitterDefaultImpl1::fitNormal (double ∗ *sqrerror,* double ∗ *avg,* double ∗ *stddev*) [virtual]

Implements Fitter_if.

5.29.2.6 void FitterDefaultImpl1::fitTriangular (double ∗ *sqrerror,* double ∗ *min,* double ∗ *mo,* double ∗ *max*) [virtual]

Implements Fitter_if.

5.29.2.7 void FitterDefaultImpl1::fitUniform (double ∗ *sqrerror,* double ∗ *min,* double ∗ *max*) [virtual]

Implements Fitter_if.

5.29.2.8 void FitterDefaultImpl1::fitWeibull (double ∗ *sqrerror,* double ∗ *alpha,* double ∗ *scale*) [virtual]

Implements Fitter_if.

5.29.2.9 std::string FitterDefaultImpl1::getDataFilename ( ) [virtual]

Implements Fitter_if.

5.29.2.10 bool FitterDefaultImpl1::isNormalDistributed (double *confidencelevel*) [virtual]

Implements Fitter_if.

5.29.2.11 void FitterDefaultImpl1::setDataFilename (std::string *dataFilename*) [virtual]

Implements Fitter_if.

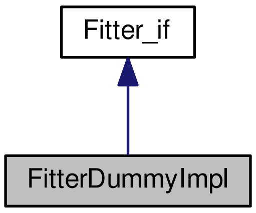
The documentation for this class was generated from the following files:

- FitterDefaultImpl1.h
- FitterDefaultImpl1.cpp

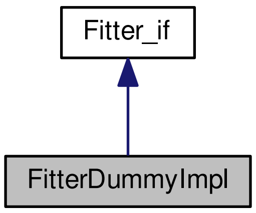
5.30 FitterDummyImpl Class Reference

```
#include <FitterDummyImpl.h>
```

Inheritance diagram for FitterDummyImpl:



Collaboration diagram for FitterDummyImpl:



Public Member Functions

- FitterDummyImpl ()
- FitterDummyImpl (const FitterDummyImpl &orig)
- ∼FitterDummyImpl ()
- bool isNormalDistributed (double confidencelevel)
- void fitUniform (double ∗sqrerror, double ∗min, double ∗max)
- void fitTriangular (double ∗sqrerror, double ∗min, double ∗mo, double ∗max)
- void fitNormal (double ∗sqrerror, double ∗avg, double ∗stddev)
- void fitExpo (double ∗sqrerror, double ∗avg1)
- void fitErlang (double ∗sqrerror, double ∗avg, double ∗m)
- void fitBeta (double ∗sqrerror, double ∗alpha, double ∗beta, double ∗infLimit, double ∗supLimit)
- void fitWeibull (double ∗sqrerror, double ∗alpha, double ∗scale)
- void fitAll (double ∗sqrerror, std::string ∗name)
- void setDataFilename (std::string dataFilename)
- std::string getDataFilename ()

5.30.1 Constructor & Destructor Documentation

5.30.1.1 FitterDummyImpl::FitterDummyImpl ( )

5.30.1.2 FitterDummyImpl::FitterDummyImpl (const **FitterDummyImpl** & *orig*)

5.30.1.3 FitterDummyImpl::∼FitterDummyImpl ( )

5.30.2 Member Function Documentation

5.30.2.1 void FitterDummyImpl::fitAll (double ∗ *sqrerror,* std::string ∗ *name*) `[virtual]`

Implements Fitter_if.

5.30.2.2 void FitterDummyImpl::fitBeta (double ∗ *sqrerror,* double ∗ *alpha,* double ∗ *beta,* double ∗ *infLimit,* double ∗ *supLimit*) `[virtual]`

Implements Fitter_if.

5.30.2.3 void FitterDummyImpl::fitErlang (double ∗ *sqrerror,* double ∗ *avg,* double ∗ *m*) `[virtual]`

Implements Fitter_if.

5.30.2.4 void FitterDummyImpl::fitExpo (double ∗ *sqrerror,* double ∗ *avg1*) `[virtual]`

Implements Fitter_if.

5.30.2.5 void FitterDummyImpl::fitNormal (double ∗ *sqrerror,* double ∗ *avg,* double ∗ *stddev*) `[virtual]`

Implements Fitter_if.

5.30.2.6 void FitterDummyImpl::fitTriangular (double ∗ *sqrerror,* double ∗ *min,* double ∗ *mo,* double ∗ *max*) `[virtual]`

Implements Fitter_if.

5.30.2.7 void FitterDummyImpl::fitUniform (double ∗ *sqrerror,* double ∗ *min,* double ∗ *max*) `[virtual]`

Implements Fitter_if.

5.30.2.8 void FitterDummyImpl::fitWeibull (double ∗ *sqrerror,* double ∗ *alpha,* double ∗ *scale*) `[virtual]`

Implements Fitter_if.

5.30.2.9 std::string FitterDummyImpl::getDataFilename ( ) [virtual]

Implements Fitter_if.

5.30.2.10 bool FitterDummyImpl::isNormalDistributed (double *confidencelevel*) [virtual]

Implements Fitter_if.

5.30.2.11 void FitterDummyImpl::setDataFilename (std::string *dataFilename*) [virtual]

Implements Fitter_if.

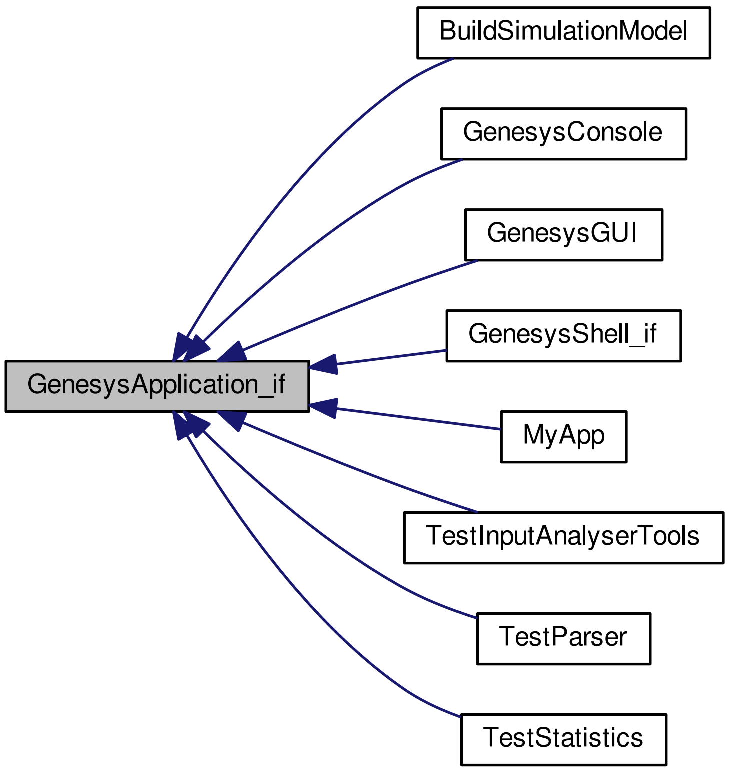
The documentation for this class was generated from the following files:

- FitterDummyImpl.h
- FitterDummyImpl.cpp

5.31 GenesysApplication_if Class Reference

```
#include <GenesysApplication_if.h>
```

Inheritance diagram for GenesysApplication_if:



Public Member Functions

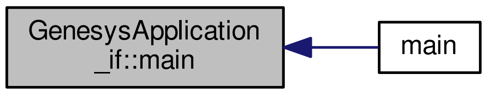
- virtual int main (int argc, char ∗∗argv)=0

5.31.1 Member Function Documentation

5.31.1.1 virtual int GenesysApplication_if::main (int *argc,* char ∗∗ *argv*) `[pure virtual]`

Implemented in GenesysConsole, MyApp, GenesysGUI, TestInputAnalyserTools, TestParser, BuildSimulation←Model, and TestStatistics.

Here is the caller graph for this function:



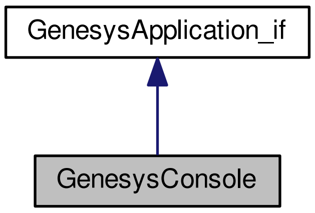
The documentation for this class was generated from the following file:

- GenesysApplication_if.h

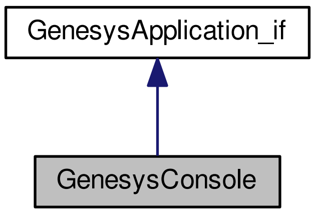
5.32 GenesysConsole Class Reference

```
#include <GenesysConsole.h>
```

Inheritance diagram for GenesysConsole:



Collaboration diagram for GenesysConsole:



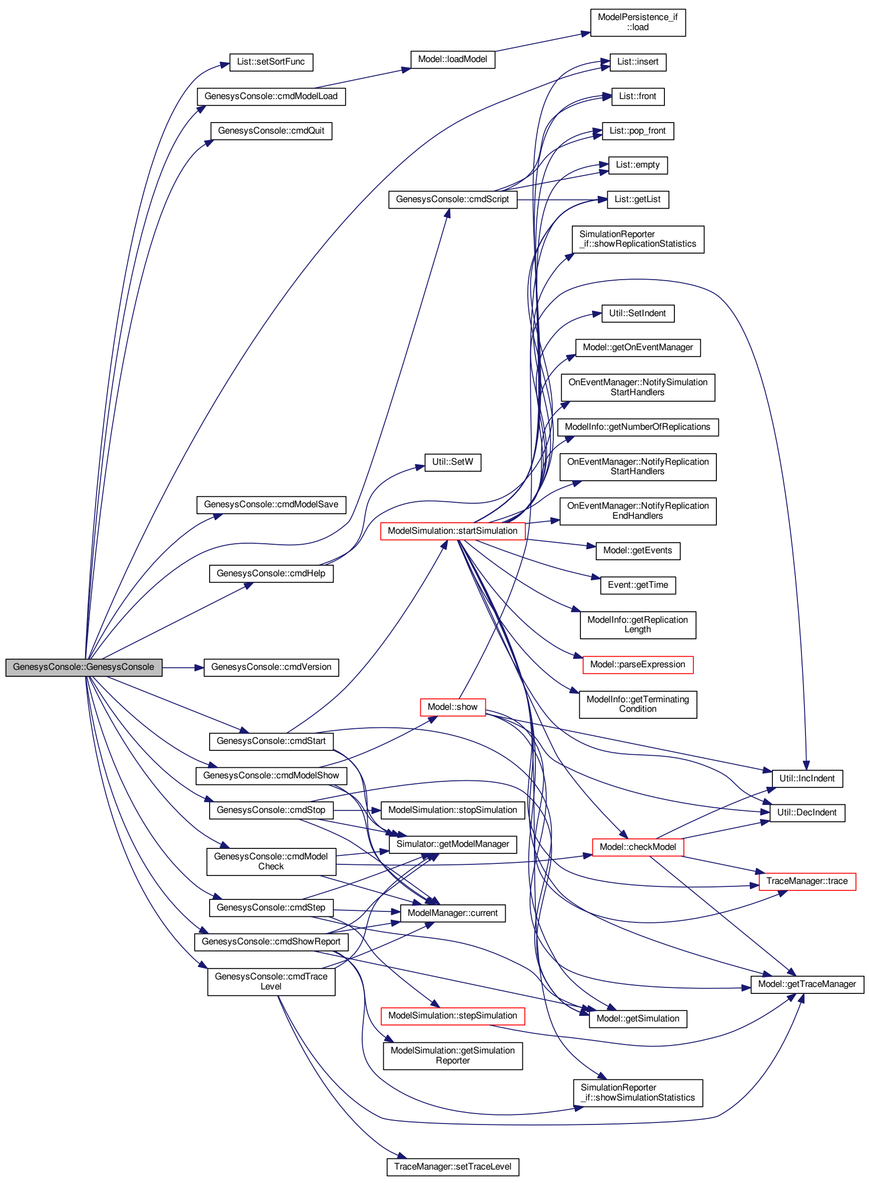
Public Member Functions

- GenesysConsole ()
- GenesysConsole (const GenesysConsole &orig)
- virtual ∼GenesysConsole ()
- virtual int main (int argc, char ∗∗argv)
- void cmdScript ()
- void cmdHelp ()
- void cmdQuit ()
- void cmdModelLoad ()
- void cmdModelCheck ()
- void cmdStart ()
- void cmdStep ()
- void cmdStop ()
- void cmdShowReport ()
- void cmdModelSave ()
- void cmdModelShow ()
- void cmdVersion ()
- void cmdTraceLevel ()

5.32.1 Constructor & Destructor Documentation

5.32.1.1 GenesysConsole::GenesysConsole ( )

Here is the call graph for this function:


GenesysConsole::GenesysConsole
List::setSortFunc
GenesysConsole::cmdModelLoad
GenesysConsole::cmdQuit
GenesysConsole::cmdModelSave
GenesysConsole::cmdHelp
GenesysConsole::cmdVersion
GenesysConsole::cmdStart
GenesysConsole::cmdModelShow
GenesysConsole::cmdStop
GenesysConsole::cmdModelCheck
GenesysConsole::cmdStep
GenesysConsole::cmdShowReport
GenesysConsole::cmdTraceLevel
Model::loadModel
ModelPersistence_if::load
GenesysConsole::cmdScript
Util::SetW
ModelSimulation::startSimulation
Model::show
ModelSimulation::stopSimulation
Simulator::getModelManager
ModelManager::current
ModelSimulation::stepSimulation
ModelSimulation::getSimulationReporter
TraceManager::setTraceLevel
List::insert
List::front
List::pop_front
List::empty
List::getList
SimulationReporter_if::showReplicationStatistics
Util::SetIndent
Model::getOnEventManager
OnEventManager::NotifySimulationStartHandlers
ModelInfo::getNumberOfReplications
OnEventManager::NotifyReplicationStartHandlers
OnEventManager::NotifyReplicationEndHandlers
Model::getEvents
Event::getTime
ModelInfo::getReplicationLength
Model::parseExpression
ModelInfo::getTerminatingCondition
Util::IncIndent
Util::DecIndent
Model::checkModel
TraceManager::trace
Model::getTraceManager
Model::getSimulation
SimulationReporter_if::showSimulationStatistics

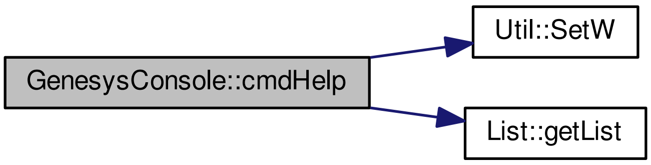
5.32.1.2 GenesysConsole::GenesysConsole (const GenesysConsole & *orig*)

5.32.1.3 GenesysConsole::∼GenesysConsole ( ) [virtual]

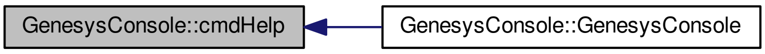
5.32.2 Member Function Documentation

5.32.2.1 void GenesysConsole::cmdHelp ( )

Here is the call graph for this function:

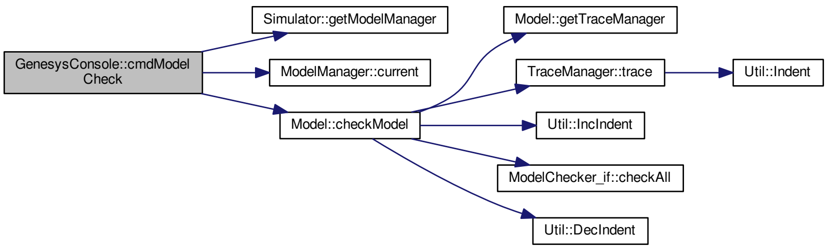


Here is the caller graph for this function:

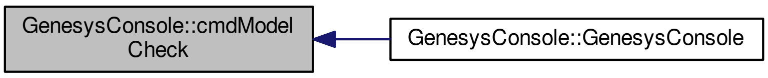


5.32.2.2 void GenesysConsole::cmdModelCheck ( )

Here is the call graph for this function:

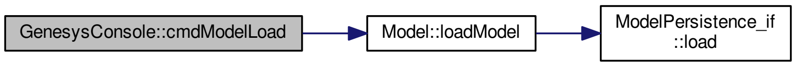


Here is the caller graph for this function:

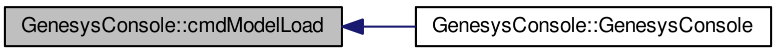

GenesysConsole::cmdModel Check
GenesysConsole::GenesysConsole

5.32.2.3 void GenesysConsole::cmdModelLoad ( )

Here is the call graph for this function:

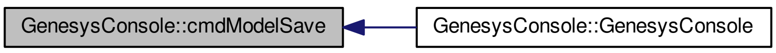


Here is the caller graph for this function:



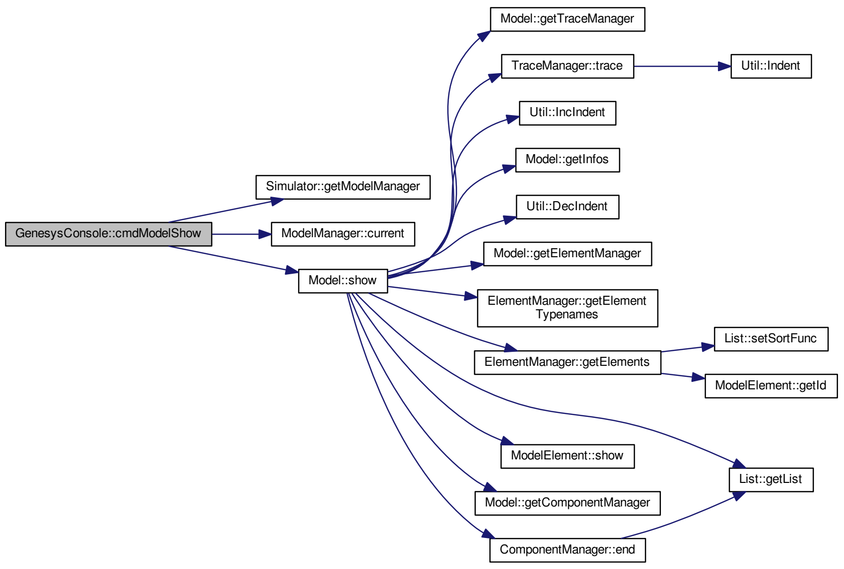
5.32.2.4 void GenesysConsole::cmdModelSave ( )

Here is the caller graph for this function:

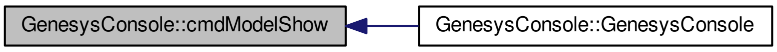


5.32.2.5 void GenesysConsole::cmdModelShow ( )

Here is the call graph for this function:

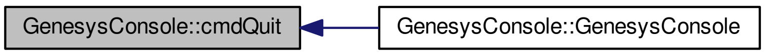


Here is the caller graph for this function:



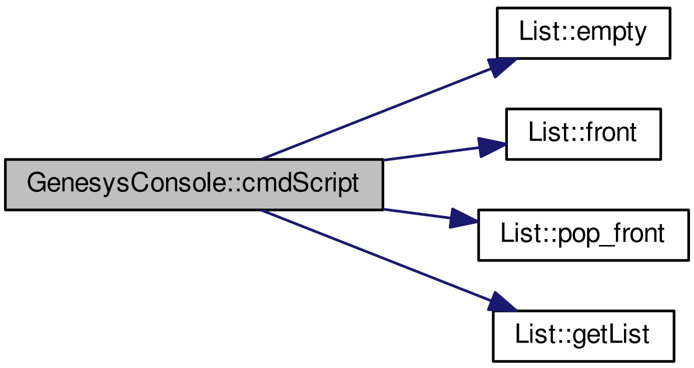
5.32.2.6 void GenesysConsole::cmdQuit ( )

Here is the caller graph for this function:

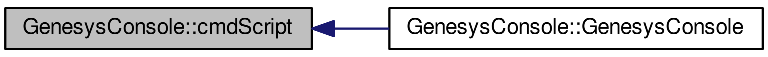


5.32.2.7 void GenesysConsole::cmdScript ( )

Here is the call graph for this function:

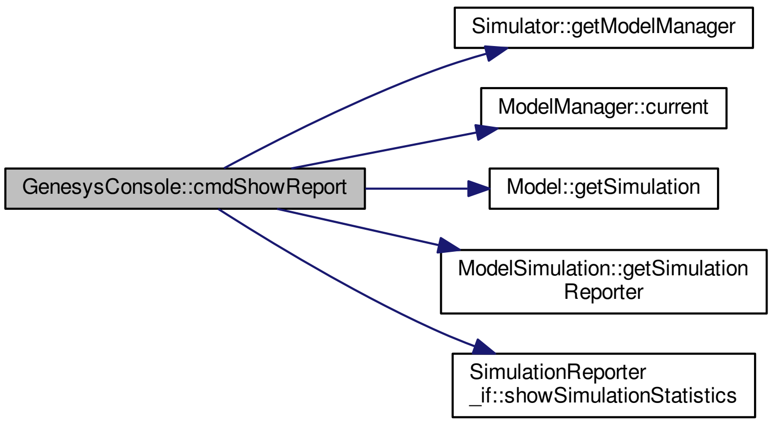


Here is the caller graph for this function:

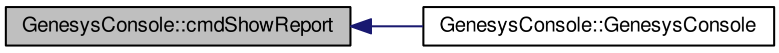


5.32.2.8 void GenesysConsole::cmdShowReport ( )

Here is the call graph for this function:

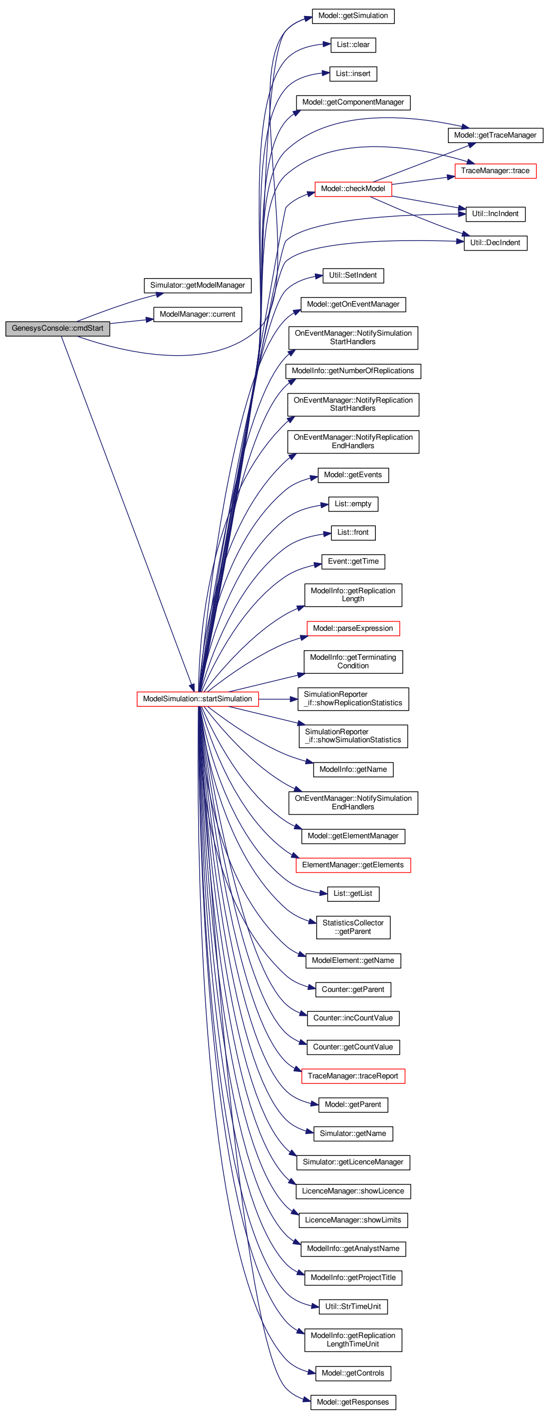

Simulator::getModelManager
ModelManager::current
GenesysConsole::cmdShowReport
Model::getSimulation
ModelSimulation::getSimulation Reporter
SimulationReporter _if::showSimulationStatistics

Here is the caller graph for this function:

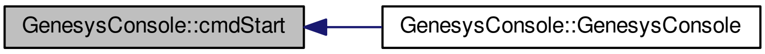


5.32.2.9 void GenesysConsole::cmdStart ( )

Here is the call graph for this function:

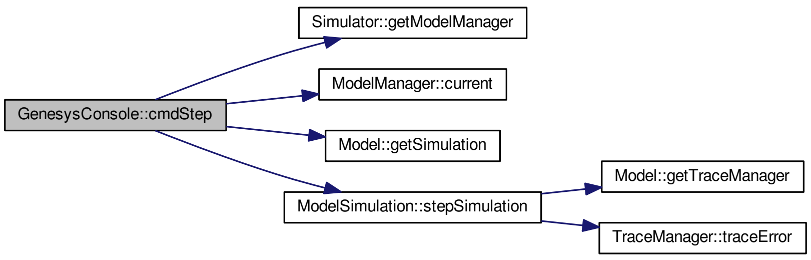

GenesysConsole::cmdStart
Simulator::getModelManager
ModelManager::current
ModelSimulation::startSimulation
Model::getSimulation
List::clear
List::insert
Model::getComponentManager
Model::getTraceManager
TraceManager::trace
Model::checkModel
Util::IncIndent
Util::DecIndent
Util::SetIndent
Model::getOnEventManager
OnEventManager::NotifySimulationStartHandlers
ModelInfo::getNumberOfReplications
OnEventManager::NotifyReplicationStartHandlers
OnEventManager::NotifyReplicationEndHandlers
Model::getEvents
List::empty
List::front
Event::getTime
ModelInfo::getReplicationLength
Model::parseExpression
ModelInfo::getTerminatingCondition
SimulationReporter_if::showReplicationStatistics
SimulationReporter_if::showSimulationStatistics
ModelInfo::getName
OnEventManager::NotifySimulationEndHandlers
Model::getElementManager
ElementManager::getElements
List::getList
StatisticsCollector::getParent
ModelElement::getName
Counter::getParent
Counter::incCountValue
Counter::getCountValue
TraceManager::traceReport
Model::getParent
Simulator::getName
Simulator::getLicenceManager
LicenceManager::showLicence
LicenceManager::showLimits
ModelInfo::getAnalystName
ModelInfo::getProjectTitle
Util::StrTimeUnit
ModelInfo::getReplicationLengthTimeUnit
Model::getControls
Model::getResponses

Here is the caller graph for this function:

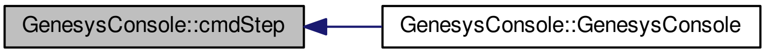


5.32.2.10 void GenesysConsole::cmdStep ( )

Here is the call graph for this function:

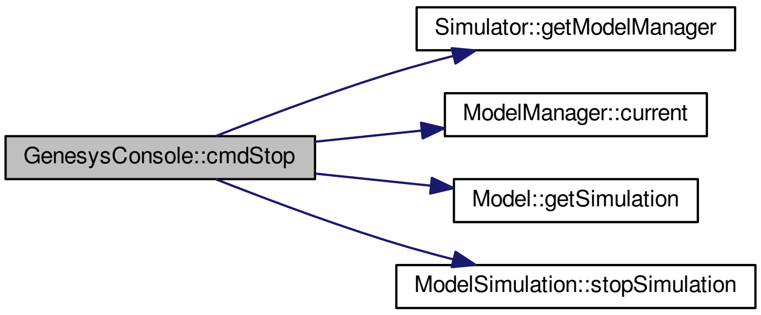


Here is the caller graph for this function:

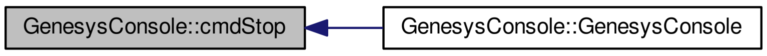


5.32.2.11 void GenesysConsole::cmdStop ( )

Here is the call graph for this function:

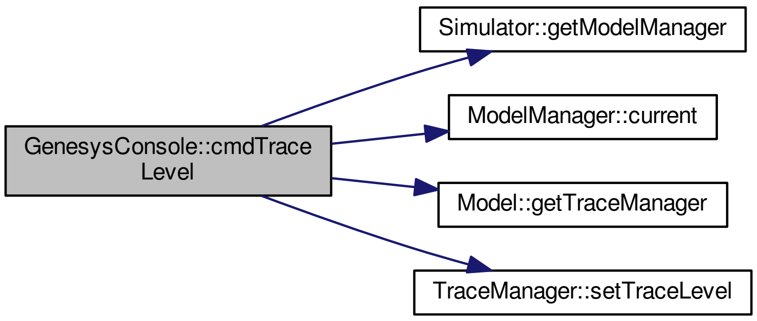


Here is the caller graph for this function:

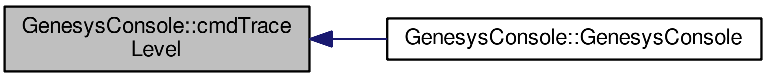


5.32.2.12 void GenesysConsole::cmdTraceLevel ( )

Here is the call graph for this function:

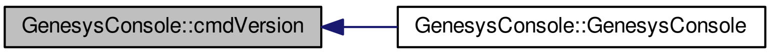

Simulator::getModelManager
ModelManager::current
GenesysConsole::cmdTrace Level
Model::getTraceManager
TraceManager::setTraceLevel

Here is the caller graph for this function:


GenesysConsole::cmdTrace Level
GenesysConsole::GenesysConsole

5.32.2.13 void GenesysConsole::cmdVersion ( )

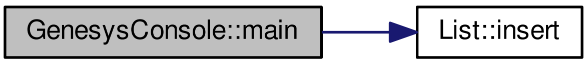
Here is the caller graph for this function:



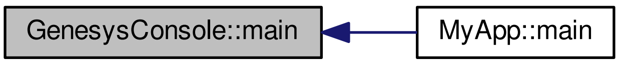
5.32.2.14 int GenesysConsole::main (int *argc*, char ∗∗ *argv*) [virtual]

Implements GenesysApplication_if.

Here is the call graph for this function:



Here is the caller graph for this function:



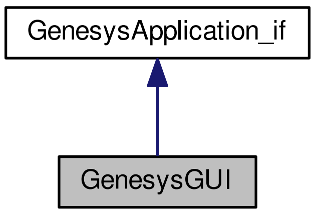
The documentation for this class was generated from the following files:

- GenesysConsole.h
- GenesysConsole.cpp

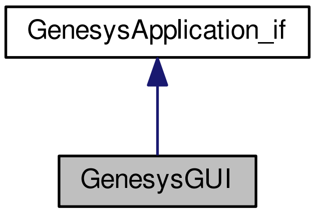
5.33 GenesysGUI Class Reference

```
#include <GenesysGUI.h>
```

Inheritance diagram for GenesysGUI:



Collaboration diagram for GenesysGUI:



Public Member Functions

- GenesysGUI ()
- GenesysGUI (const GenesysGUI &orig)
- virtual ∼GenesysGUI ()
- virtual int main (int argc, char ∗∗argv)

5.33.1 Constructor & Destructor Documentation

5.33.1.1 GenesysGUI::GenesysGUI ( )

5.33.1.2 GenesysGUI::GenesysGUI (const GenesysGUI & *orig*)

5.33.1.3 GenesysGUI::∼GenesysGUI ( ) `[virtual]`

5.33.2 Member Function Documentation

5.33.2.1 int GenesysGUI::main (int *argc*, char ∗∗ *argv*) `[virtual]`

Implements GenesysApplication_if.

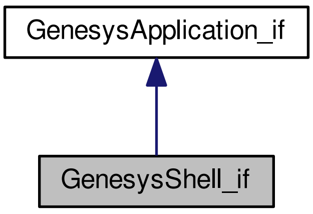
The documentation for this class was generated from the following files:

- GenesysGUI.h
- GenesysGUI.cpp

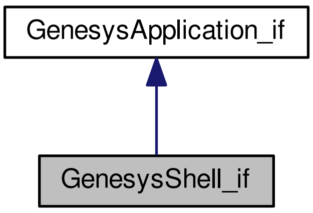
5.34 GenesysShell_if Class Reference

```
#include <GenesysShell_if.h>
```

Inheritance diagram for GenesysShell_if:



Collaboration diagram for GenesysShell_if:



Public Member Functions

- virtual void openModel (std::string filename)=0
- virtual void saveModelAs (std::string filename)=0
- virtual void saveModel ()=0
- virtual void listElements ()=0
- virtual void listComponents ()=0
- virtual void listHosts ()=0
- virtual void listPlugins ()=0
- virtual void deleteTraceFiles ()=0
- virtual void traceLevel (Util::TraceLevel tracelevel)=0
- virtual void addPlugin (std::string filename)=0
- virtual void addFromFile (std::string filename)=0
- virtual void readCommandsFromFile (std::string filename)=0
- virtual void redirectTrace (std::string trace, std::string dest, std::string filename)=0
- virtual void closeModel ()=0
- virtual void createModel ()=0
- virtual void execLinuxCommand (std::string command)=0
- virtual void verboseMode (bool on)=0
- virtual void check ()=0
- virtual void getGenesysInfo ()=0
- virtual void getCommandLine ()=0
- virtual void sendFile (std::string filename, std::string hostname, std::string portname)=0
- virtual void setActivationCode (std::string code)=0
- virtual void receiveFile (std::string filename)=0
- virtual void startSimulation ()=0
- virtual void stepSimulation ()=0
- virtual void stopSimulation ()=0
- virtual void showInit ()=0
- virtual void showHelp ()=0
- virtual void showHostName ()=0

5.34.1 Member Function Documentation

5.34.1.1 virtual void GenesysShell_if::addFromFile (std::string *filename*) `[pure virtual]`

5.34.1.2 virtual void GenesysShell_if::addPlugin (std::string *filename*) `[pure virtual]`

5.34.1.3 virtual void GenesysShell_if::check ( ) `[pure virtual]`

5.34.1.4 virtual void GenesysShell_if::closeModel ( ) `[pure virtual]`

5.34.1.5 virtual void GenesysShell_if::createModel ( ) `[pure virtual]`

5.34.1.6 virtual void GenesysShell_if::deleteTraceFiles ( ) `[pure virtual]`

5.34.1.7 virtual void GenesysShell_if::execLinuxCommand (std::string *command*) `[pure virtual]`

5.34.1.8 virtual void GenesysShell_if::getCommandLine ( ) `[pure virtual]`

5.34.1.9 virtual void GenesysShell_if::getGenesysInfo ( ) [pure virtual]

5.34.1.10 virtual void GenesysShell_if::listComponents ( ) [pure virtual]

5.34.1.11 virtual void GenesysShell_if::listElements ( ) [pure virtual]

5.34.1.12 virtual void GenesysShell_if::listHosts ( ) [pure virtual]

5.34.1.13 virtual void GenesysShell_if::listPlugins ( ) [pure virtual]

5.34.1.14 virtual void GenesysShell_if::openModel (std::string *filename*) [pure virtual]

5.34.1.15 virtual void GenesysShell_if::readCommandsFromFile (std::string *filename*) [pure virtual]

5.34.1.16 virtual void GenesysShell_if::receiveFile (std::string *filename*) [pure virtual]

5.34.1.17 virtual void GenesysShell_if::redirectTrace (std::string *trace,* std::string *dest,* std::string *filename*) [pure virtual]

5.34.1.18 virtual void GenesysShell_if::saveModel ( ) [pure virtual]

5.34.1.19 virtual void GenesysShell_if::saveModelAs (std::string *filename*) [pure virtual]

5.34.1.20 virtual void GenesysShell_if::sendFile (std::string *filename,* std::string *hostname,* std::string *portname*) [pure virtual]

5.34.1.21 virtual void GenesysShell_if::setActivationCode (std::string *code*) [pure virtual]

5.34.1.22 virtual void GenesysShell_if::showHelp ( ) [pure virtual]

5.34.1.23 virtual void GenesysShell_if::showHostName ( ) [pure virtual]

5.34.1.24 virtual void GenesysShell_if::showInit ( ) [pure virtual]

5.34.1.25 virtual void GenesysShell_if::startSimulation ( ) [pure virtual]

5.34.1.26 virtual void GenesysShell_if::stepSimulation ( ) [pure virtual]

5.34.1.27 virtual void GenesysShell_if::stopSimulation ( ) [pure virtual]

5.34.1.28 virtual void GenesysShell_if::traceLevel (Util::TraceLevel *tracelevel*) [pure virtual]

5.34.1.29 virtual void GenesysShell_if::verboseMode (bool *on*) [pure virtual]

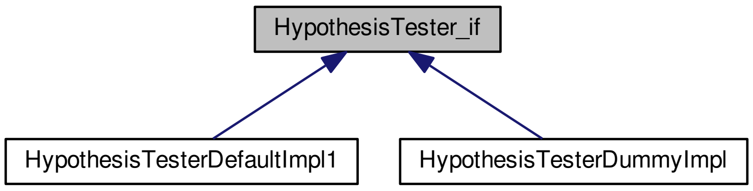
The documentation for this class was generated from the following file:

- GenesysShell_if.h

5.35 HypothesisTester_if Class Reference

```
#include <HypothesisTester_if.h>
```

Inheritance diagram for HypothesisTester_if:



Public Types

- enum H1Comparition { DIFFERENT = 1, LESS_THAN = 2, GREATER_THAN = 3 }

Public Member Functions

- virtual double testAverage (double confidencelevel, double avg, H1Comparition comp)=0
- virtual double testProportion (double confidencelevel, double prop, H1Comparition comp)=0
- virtual double testVariance (double confidencelevel, double var, H1Comparition comp)=0
- virtual double testAverage (double confidencelevel, std::string secondPopulationDataFilename, H1↩ Comparition comp)=0
- virtual double testProportion (double confidencelevel, std::string secondPopulationDataFilename, H1↩ Comparition comp)=0
- virtual double testVariance (double confidencelevel, std::string secondPopulationDataFilename, H1↩ Comparition comp)=0
- virtual void setDataFilename (std::string dataFilename)=0
- virtual std::string getDataFilename ()=0

5.35.1 Detailed Description

Interface for parametric hypothesis tests based on a datafile.

5.35.2 Member Enumeration Documentation

5.35.2.1 enum HypothesisTester_if::H1Comparition

Enumerator

DIFFERENT

LESS_THAN

GREATER_THAN

5.35.3 Member Function Documentation

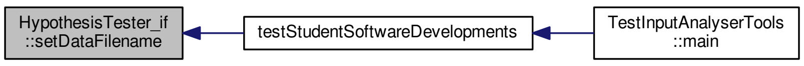
5.35.3.1 virtual std::string HypothesisTester_if::getDataFilename ( ) [pure virtual]

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

5.35.3.2 virtual void HypothesisTester_if::setDataFilename (std::string *dataFilename*) [pure virtual]

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

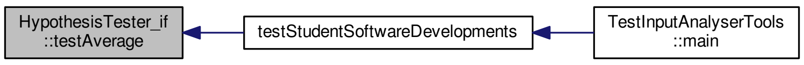
Here is the caller graph for this function:


HypothesisTester_if ::setDataFilename
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.35.3.3 virtual double HypothesisTester_if::testAverage (double *confidencelevel,* double *avg,* H1Comparition *comp*) [pure virtual]

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

Here is the caller graph for this function:


HypothesisTester_if ::testAverage
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.35.3.4 virtual double HypothesisTester_if::testAverage (double *confidencelevel,* std::string *secondPopulationDataFilename,* H1Comparition *comp*) [pure virtual]

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

5.35.3.5 virtual double HypothesisTester_if::testProportion (double *confidencelevel,* double *prop,* H1Comparition *comp*) [pure virtual]

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

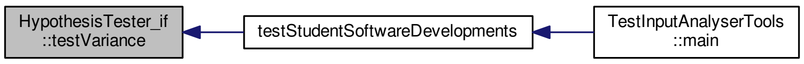
5.35.3.6 virtual double HypothesisTester_if::testProportion (double *confidencelevel,* std::string *secondPopulationDataFilename,* H1Comparition *comp*) `[pure virtual]`

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

5.35.3.7 virtual double HypothesisTester_if::testVariance (double *confidencelevel,* double *var,* H1Comparition *comp*) `[pure virtual]`

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

Here is the caller graph for this function:



5.35.3.8 virtual double HypothesisTester_if::testVariance (double *confidencelevel,* std::string *secondPopulationDataFilename,* H1Comparition *comp*) `[pure virtual]`

Implemented in HypothesisTesterDefaultImpl1, and HypothesisTesterDummyImpl.

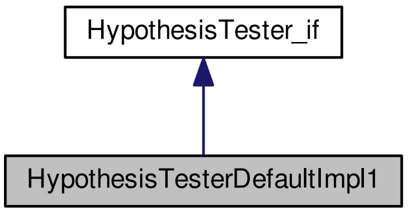
The documentation for this class was generated from the following file:

- HypothesisTester_if.h

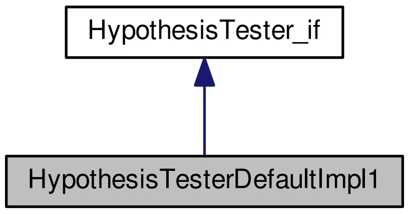
5.36 HypothesisTesterDefaultImpl1 Class Reference

```
#include <HypothesisTesterDefaultImpl1.h>
```

Inheritance diagram for HypothesisTesterDefaultImpl1:



Collaboration diagram for HypothesisTesterDefaultImpl1:



Public Member Functions

- HypothesisTesterDefaultImpl1 ()
- HypothesisTesterDefaultImpl1 (const HypothesisTesterDefaultImpl1 &orig)
- virtual ∼HypothesisTesterDefaultImpl1 ()
- virtual double testAverage (double confidencelevel, double avg, H1Comparition comp)
- virtual double testProportion (double confidencelevel, double prop, H1Comparition comp)
- virtual double testVariance (double confidencelevel, double var, H1Comparition comp)
- virtual double testAverage (double confidencelevel, std::string secondPopulationDataFilename, H1←Comparition comp)
- virtual double testProportion (double confidencelevel, std::string secondPopulationDataFilename, H1←Comparition comp)
- virtual double testVariance (double confidencelevel, std::string secondPopulationDataFilename, H1←Comparition comp)
- virtual void setDataFilename (std::string dataFilename)
- virtual std::string getDataFilename ()

Additional Inherited Members

5.36.1 Constructor & Destructor Documentation

5.36.1.1 HypothesisTesterDefaultImpl1::HypothesisTesterDefaultImpl1 ( )

5.36.1.2 HypothesisTesterDefaultImpl1::HypothesisTesterDefaultImpl1 (const HypothesisTesterDefaultImpl1 & *orig*)

5.36.1.3 HypothesisTesterDefaultImpl1::∼HypothesisTesterDefaultImpl1 ( ) [virtual]

5.36.2 Member Function Documentation

5.36.2.1 std::string HypothesisTesterDefaultImpl1::getDataFilename ( ) [virtual]

Implements HypothesisTester_if.

5.36.2.2 void HypothesisTesterDefaultImpl1::setDataFilename (std::string *dataFilename*) `[virtual]`

Implements HypothesisTester_if.

5.36.2.3 double HypothesisTesterDefaultImpl1::testAverage (double *confidencelevel*, double *avg*, H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.36.2.4 double HypothesisTesterDefaultImpl1::testAverage (double *confidencelevel*, std::string *secondPopulationDataFilename*, H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.36.2.5 double HypothesisTesterDefaultImpl1::testProportion (double *confidencelevel*, double *prop*, H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.36.2.6 double HypothesisTesterDefaultImpl1::testProportion (double *confidencelevel*, std::string *secondPopulationDataFilename*, H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.36.2.7 double HypothesisTesterDefaultImpl1::testVariance (double *confidencelevel*, double *var*, H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.36.2.8 double HypothesisTesterDefaultImpl1::testVariance (double *confidencelevel*, std::string *secondPopulationDataFilename*, H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

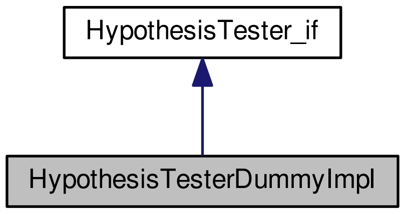
The documentation for this class was generated from the following files:

- HypothesisTesterDefaultImpl1.h
- HypothesisTesterDefaultImpl1.cpp

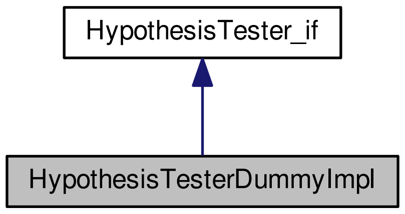
5.37 HypothesisTesterDummyImpl Class Reference

```
#include <HypothesisTesterDummyImpl.h>
```

Inheritance diagram for HypothesisTesterDummyImpl:



Collaboration diagram for HypothesisTesterDummyImpl:



Public Member Functions

- HypothesisTesterDummyImpl ()
- HypothesisTesterDummyImpl (const HypothesisTesterDummyImpl &orig)
- ∼HypothesisTesterDummyImpl ()
- virtual double testAverage (double confidencelevel, double avg, H1Comparition comp)
- virtual double testProportion (double confidencelevel, double prop, H1Comparition comp)
- virtual double testVariance (double confidencelevel, double var, H1Comparition comp)
- virtual double testAverage (double confidencelevel, std::string secondPopulationDataFilename, H1↩ Comparition comp)
- virtual double testProportion (double confidencelevel, std::string secondPopulationDataFilename, H1↩ Comparition comp)
- virtual double testVariance (double confidencelevel, std::string secondPopulationDataFilename, H1↩ Comparition comp)
- virtual void setDataFilename (std::string dataFilename)
- virtual std::string getDataFilename ()

Additional Inherited Members

5.37.1 Constructor & Destructor Documentation

5.37.1.1 HypothesisTesterDummyImpl::HypothesisTesterDummyImpl ( )

5.37.1.2 HypothesisTesterDummyImpl::HypothesisTesterDummyImpl (const HypothesisTesterDummyImpl & *orig*)

5.37.1.3 HypothesisTesterDummyImpl::∼HypothesisTesterDummyImpl ( )

5.37.2 Member Function Documentation

5.37.2.1 std::string HypothesisTesterDummyImpl::getDataFilename ( ) `[virtual]`

Implements HypothesisTester_if.

5.37.2.2 void HypothesisTesterDummyImpl::setDataFilename (std::string *dataFilename*) `[virtual]`

Implements HypothesisTester_if.

5.37.2.3 double HypothesisTesterDummyImpl::testAverage (double *confidencelevel,* double *avg,* H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.37.2.4 double HypothesisTesterDummyImpl::testAverage (double *confidencelevel,* std::string *secondPopulationDataFilename,* H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.37.2.5 double HypothesisTesterDummyImpl::testProportion (double *confidencelevel,* double *prop,* H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.37.2.6 double HypothesisTesterDummyImpl::testProportion (double *confidencelevel,* std::string *secondPopulationDataFilename,* H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.37.2.7 double HypothesisTesterDummyImpl::testVariance (double *confidencelevel,* double *var,* H1Comparition *comp*) `[virtual]`

Implements HypothesisTester_if.

5.37.2.8 double HypothesisTesterDummyImpl::testVariance (double *confidencelevel,* std::string *secondPopulationDataFilename,* H1Comparition *comp*) [virtual]

Implements HypothesisTester_if.

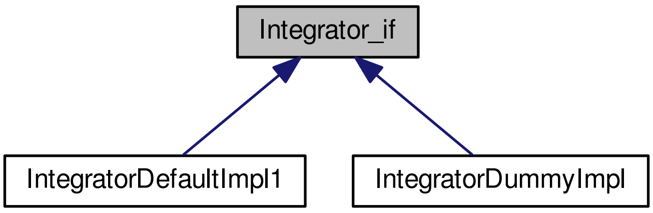
The documentation for this class was generated from the following files:

- HypothesisTesterDummyImpl.h
- HypothesisTesterDummyImpl.cpp

5.38 Integrator_if Class Reference

`#include <Integrator_if.h>`

Inheritance diagram for Integrator_if:



Public Member Functions

- virtual void setPrecision (double e)=0
- virtual double getPrecision ()=0
- virtual double integrate (double min, double max, double(∗f)(double, double), double p2)=0
- virtual double integrate (double min, double max, double(∗f)(double, double, double), double p2, double p3)=0
- virtual double integrate (double min, double max, double(∗f)(double, double, double, double), double p2, double p3, double p4)=0
- virtual double integrate (double min, double max, double(∗f)(double, double, double, double, double), double p2, double p3, double p4, double p5)=0

5.38.1 Detailed Description

Interface used by classes that perform the numerical integration of functions with one to four parameters. It is mainly used for calculating the probability of theoretical distributions, from its probability distribution functions.

5.38.2 Member Function Documentation

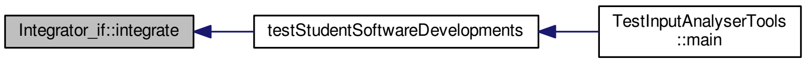
5.38.2.1 virtual double Integrator_if::getPrecision ( ) [pure virtual]

Implemented in IntegratorDefaultImpl1, and IntegratorDummyImpl.

5.38.2.2 virtual double Integrator_if::integrate (double *min*, double *max*, double(∗)(double, double) *f*, double *p2*) [pure virtual]

Implemented in IntegratorDefaultImpl1, and IntegratorDummyImpl.

Here is the caller graph for this function:


Integrator_if::integrate
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.38.2.3 virtual double Integrator_if::integrate (double *min*, double *max*, double(∗)(double, double, double) *f*, double *p2*, double *p3*) [pure virtual]

Implemented in IntegratorDefaultImpl1, and IntegratorDummyImpl.

5.38.2.4 virtual double Integrator_if::integrate (double *min*, double *max*, double(∗)(double, double, double, double) *f*, double *p2*, double *p3*, double *p4*) [pure virtual]

Implemented in IntegratorDefaultImpl1, and IntegratorDummyImpl.

5.38.2.5 virtual double Integrator_if::integrate (double *min*, double *max*, double(∗)(double, double, double, double, double) *f*, double *p2*, double *p3*, double *p4*, double *p5*) [pure virtual]

Implemented in IntegratorDefaultImpl1, and IntegratorDummyImpl.

5.38.2.6 virtual void Integrator_if::setPrecision (double *e*) [pure virtual]

Implemented in IntegratorDefaultImpl1, and IntegratorDummyImpl.

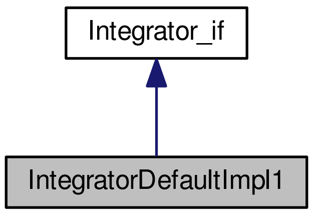
The documentation for this class was generated from the following file:

- Integrator_if.h

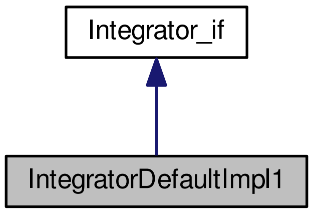
5.39 IntegratorDefaultImpl1 Class Reference

```
#include <IntegratorDefaultImpl1.h>
```

Inheritance diagram for IntegratorDefaultImpl1:



Collaboration diagram for IntegratorDefaultImpl1:



Public Member Functions

- IntegratorDefaultImpl1 ()

  `https://codereview.stackexchange.com/questions/200289/implementing-numerical-integration`
- IntegratorDefaultImpl1 (const IntegratorDefaultImpl1 &orig)
- virtual ∼IntegratorDefaultImpl1 ()
- virtual void setPrecision (double e)
- virtual double getPrecision ()
- virtual double integrate (double min, double max, double(∗f)(double, double), double p2)
- virtual double integrate (double min, double max, double(∗f)(double, double, double), double p2, double p3)
- virtual double integrate (double min, double max, double(∗f)(double, double, double, double), double p2, double p3, double p4)
- virtual double integrate (double min, double max, double(∗f)(double, double, double, double, double), double p2, double p3, double p4, double p5)

5.39.1 Constructor & Destructor Documentation

5.39.1.1 IntegratorDefaultImpl1::IntegratorDefaultImpl1 ( )

https://codereview.stackexchange.com/questions/200289/implementing-numerical-integration

5.39.1.2 IntegratorDefaultImpl1::IntegratorDefaultImpl1 (const IntegratorDefaultImpl1 & *orig*)

5.39.1.3 IntegratorDefaultImpl1::∼IntegratorDefaultImpl1 ( ) [virtual]

5.39.2 Member Function Documentation

5.39.2.1 double IntegratorDefaultImpl1::getPrecision ( ) [virtual]

Implements Integrator_if.

5.39.2.2 double IntegratorDefaultImpl1::integrate (double *min*, double *max*, double(∗)(double, double) *f*, double *p2*) [virtual]

Implements Integrator_if.

5.39.2.3 double IntegratorDefaultImpl1::integrate (double *min*, double *max*, double(∗)(double, double, double) *f*, double *p2*, double *p3*) [virtual]

Implements Integrator_if.

5.39.2.4 double IntegratorDefaultImpl1::integrate (double *min*, double *max*, double(∗)(double, double, double, double) *f*, double *p2*, double *p3*, double *p4*) [virtual]

Implements Integrator_if.

5.39.2.5 double IntegratorDefaultImpl1::integrate (double *min*, double *max*, double(∗)(double, double, double, double, double) *f*, double *p2*, double *p3*, double *p4*, double *p5*) [virtual]

Implements Integrator_if.

5.39.2.6 void IntegratorDefaultImpl1::setPrecision (double *e*) [virtual]

Implements Integrator_if.

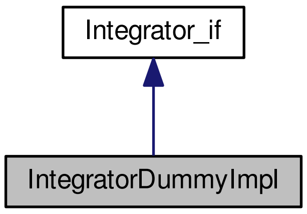
The documentation for this class was generated from the following files:

- IntegratorDefaultImpl1.h
- IntegratorDefaultImpl1.cpp

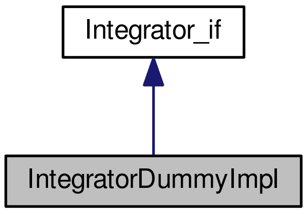
5.40 IntegratorDummyImpl Class Reference

```
#include <IntegratorDummyImpl.h>
```

Inheritance diagram for IntegratorDummyImpl:



Collaboration diagram for IntegratorDummyImpl:



Public Member Functions

- IntegratorDummyImpl ()
- IntegratorDummyImpl (const IntegratorDummyImpl &orig)
- ∼IntegratorDummyImpl ()
- void setPrecision (double e)
- double getPrecision ()
- double integrate (double min, double max, double(∗f)(double, double), double p2)
- double integrate (double min, double max, double(∗f)(double, double, double), double p2, double p3)
- double integrate (double min, double max, double(∗f)(double, double, double, double), double p2, double p3, double p4)
- double integrate (double min, double max, double(∗f)(double, double, double, double, double), double p2, double p3, double p4, double p5)

5.40.1 Constructor & Destructor Documentation

5.40.1.1 IntegratorDummyImpl::IntegratorDummyImpl ( )

5.40.1.2 IntegratorDummyImpl::IntegratorDummyImpl (const IntegratorDummyImpl & *orig*)

5.40.1.3 IntegratorDummyImpl::∼IntegratorDummyImpl ( )

5.40.2 Member Function Documentation

5.40.2.1 double IntegratorDummyImpl::getPrecision ( ) `[virtual]`

Implements Integrator_if.

5.40.2.2 double IntegratorDummyImpl::integrate (double *min*, double *max*, double(∗)(double, double) *f*, double *p2*) `[virtual]`

Implements Integrator_if.

5.40.2.3 double IntegratorDummyImpl::integrate (double *min*, double *max*, double(∗)(double, double, double) *f*, double *p2*, double *p3*) `[virtual]`

Implements Integrator_if.

5.40.2.4 double IntegratorDummyImpl::integrate (double *min*, double *max*, double(∗)(double, double, double, double) *f*, double *p2*, double *p3*, double *p4*) `[virtual]`

Implements Integrator_if.

5.40.2.5 double IntegratorDummyImpl::integrate (double *min*, double *max*, double(∗)(double, double, double, double, double) *f*, double *p2*, double *p3*, double *p4*, double *p5*) `[virtual]`

Implements Integrator_if.

5.40.2.6 void IntegratorDummyImpl::setPrecision (double *e*) `[virtual]`

Implements Integrator_if.

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

- IntegratorDummyImpl.h
- IntegratorDummyImpl.cpp

5.41 LicenceManager Class Reference

```
#include <LicenceManager.h>
```

Public Member Functions

- LicenceManager (Simulator ∗simulator)
- LicenceManager (const LicenceManager &orig)
- virtual ∼LicenceManager ()
- const std::string showLicence () const
- const std::string showLimits () const
- const std::string showActivationCode () const
- bool lookforActivationCode ()
- bool insertActivationCode ()
- void removeActivationCode ()
- unsigned int getModelComponentsLimit ()
- unsigned int getModelElementsLimit ()
- unsigned int getEntityLimit ()
- unsigned int getHostsLimit ()
- unsigned int getThreadsLimit ()

5.41.1 Constructor & Destructor Documentation

5.41.1.1 LicenceManager::LicenceManager (Simulator ∗ *simulator*)

5.41.1.2 LicenceManager::LicenceManager (const LicenceManager & *orig*)

5.41.1.3 LicenceManager::∼LicenceManager ( ) [virtual]

5.41.2 Member Function Documentation

5.41.2.1 unsigned int LicenceManager::getEntityLimit ( )

5.41.2.2 unsigned int LicenceManager::getHostsLimit ( )

5.41.2.3 unsigned int LicenceManager::getModelComponentsLimit ( )

5.41.2.4 unsigned int LicenceManager::getModelElementsLimit ( )

5.41.2.5 unsigned int LicenceManager::getThreadsLimit ( )

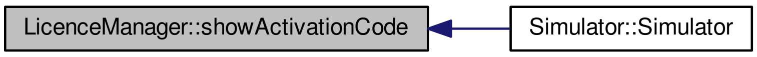
5.41.2.6 bool LicenceManager::insertActivationCode ( )

5.41.2.7 bool LicenceManager::lookforActivationCode ( )

5.41.2.8 void LicenceManager::removeActivationCode ( )

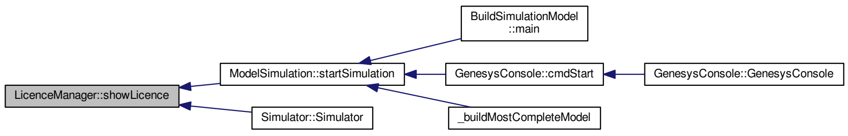
5.41.2.9 const std::string LicenceManager::showActivationCode ( ) const

Here is the caller graph for this function:



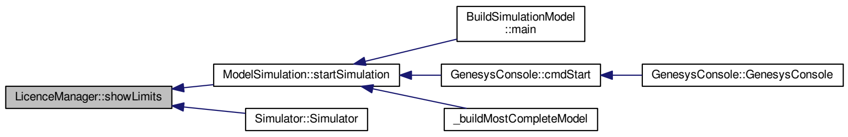
5.41.2.10 const std::string LicenceManager::showLicence ( ) const

Here is the caller graph for this function:


BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
LicenceManager::showLicence
Simulator::Simulator
_buildMostCompleteModel

5.41.2.11 const std::string LicenceManager::showLimits ( ) const

Here is the caller graph for this function:


BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
LicenceManager::showLimits
Simulator::Simulator
_buildMostCompleteModel

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

- LicenceManager.h
- LicenceManager.cpp

5.42 LinkedBy Class Reference

```
#include <LinkedBy.h>
```

Public Member Functions

- LinkedBy ()
- LinkedBy (const LinkedBy &orig)
- virtual ∼LinkedBy ()
- void addLink ()
- void removeLink ()
- bool isLinked ()

5.42.1 Constructor & Destructor Documentation

5.42.1.1 LinkedBy::LinkedBy ( )

5.42.1.2 LinkedBy::LinkedBy (const LinkedBy & *orig*)

5.42.1.3 LinkedBy::∼LinkedBy ( ) [virtual]

5.42.2 Member Function Documentation

5.42.2.1 void LinkedBy::addLink ( )

5.42.2.2 bool LinkedBy::isLinked ( )

5.42.2.3 void LinkedBy::removeLink ( )

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

- LinkedBy.h
- LinkedBy.cpp

5.43 List< T > Class Template Reference

```
#include <List.h>
```

Public Types

- using CompFunct = std::function< bool(const T, const T) >

Public Member Functions

- List ()
- List (const List &orig)
- virtual ~List ()
- unsigned int size ()
- bool empty ()
- void clear ()
- void pop_front ()
- template<class Compare >
  void sort (Compare comp)
- std::list< T > * getList () const
- T create ()
- template<typename U >
  T create (U arg)
- std::string show ()
- std::list< T >::iterator find (T element)
- void insert (T element)
- void remove (T element)
- void setAtRank (unsigned int rank, T element)
- T getAtRank (unsigned int rank)
- T next ()
- T front ()
- T last ()
- T previous ()
- T actual ()
- void setSortFunc (CompFunct _sortFunc)

5.43.1 Detailed Description

template<typename T>
class List< T >

List corresponds to an extended version of the list that must guarantee the consistency of the elements that make up the simulation model.

5.43.2 Member Typedef Documentation

5.43.2.1 template<typename T> using List< T >::CompFunct = std::function<bool(const T, const T) >

5.43.3 Constructor & Destructor Documentation

5.43.3.1 template<typename T > List< T >::List ( )

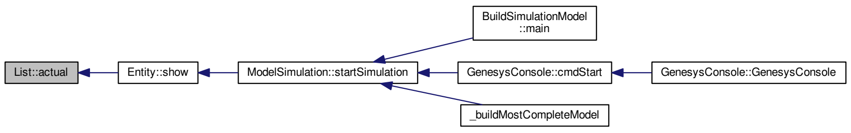
5.43.3.2 template<typename T > List< T >::List (const List< T > & *orig*)

5.43.3.3 template<typename T > List< T >::~List ( ) [virtual]

5.43.4 Member Function Documentation

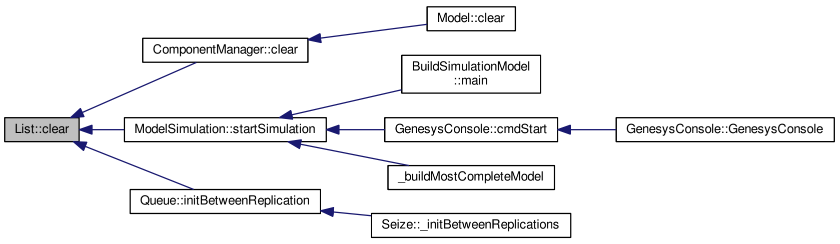
5.43.4.1 template<typename T > T List< T >::actual ( )

Here is the caller graph for this function:


List::actual
Entity::show
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.43.4.2 template<typename T > void List< T >::clear ( )

Here is the caller graph for this function:

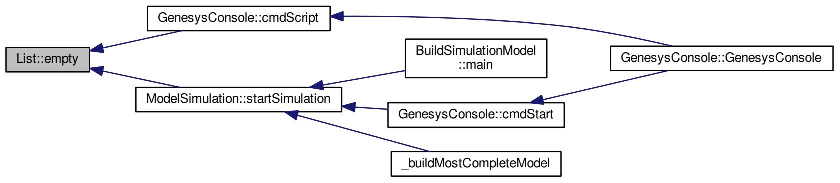

List::clear
ComponentManager::clear
Model::clear
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel
Queue::initBetweenReplication
Seize::_initBetweenReplications

5.43.4.3 template<typename T > T List< T >::create ( )

5.43.4.4 template<typename T > template<typename U > T List< T >::create (U *arg*)

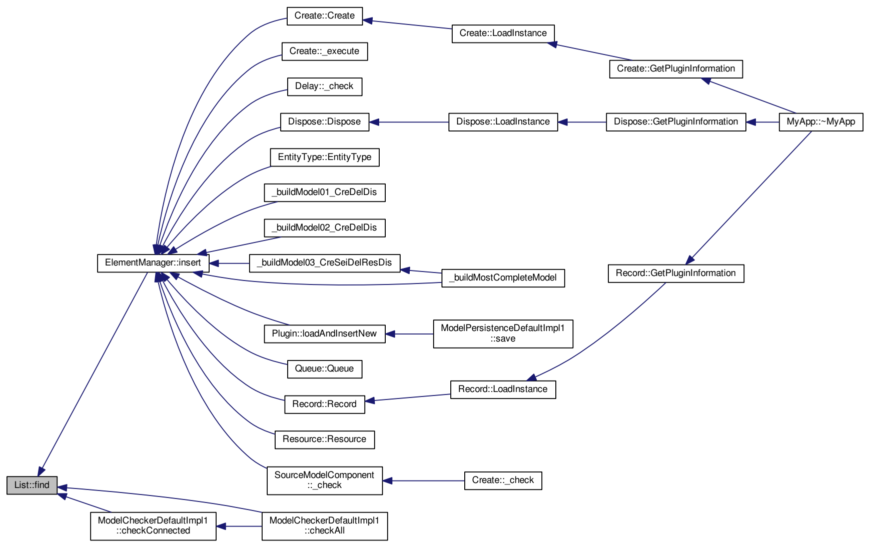
5.43.4.5 template<typename T > bool List< T >::empty ( )

Here is the caller graph for this function:


List::empty
GenesysConsole::cmdScript
GenesysConsole::GenesysConsole
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel

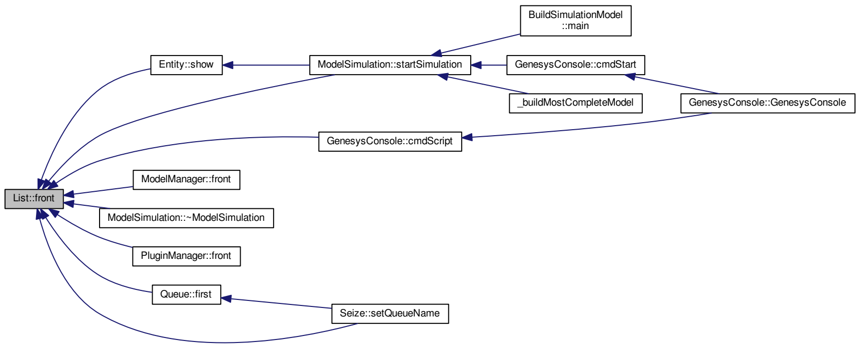
5.43.4.6 template<typename T> std::list< T >::iterator List< T >::find (T *element*)

Here is the caller graph for this function:



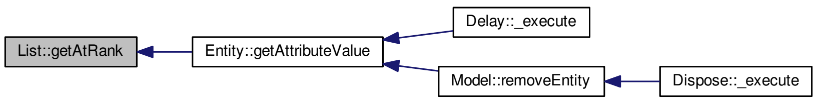
5.43.4.7 template<typename T > T List< T >::front ( )

Here is the caller graph for this function:


BuildSimulationModel ::main
Entity::show
ModelSimulation::startSimulation
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole
GenesysConsole::cmdScript
ModelManager::front
List::front
ModelSimulation::~ModelSimulation
PluginManager::front
Queue::first
Seize::setQueueName

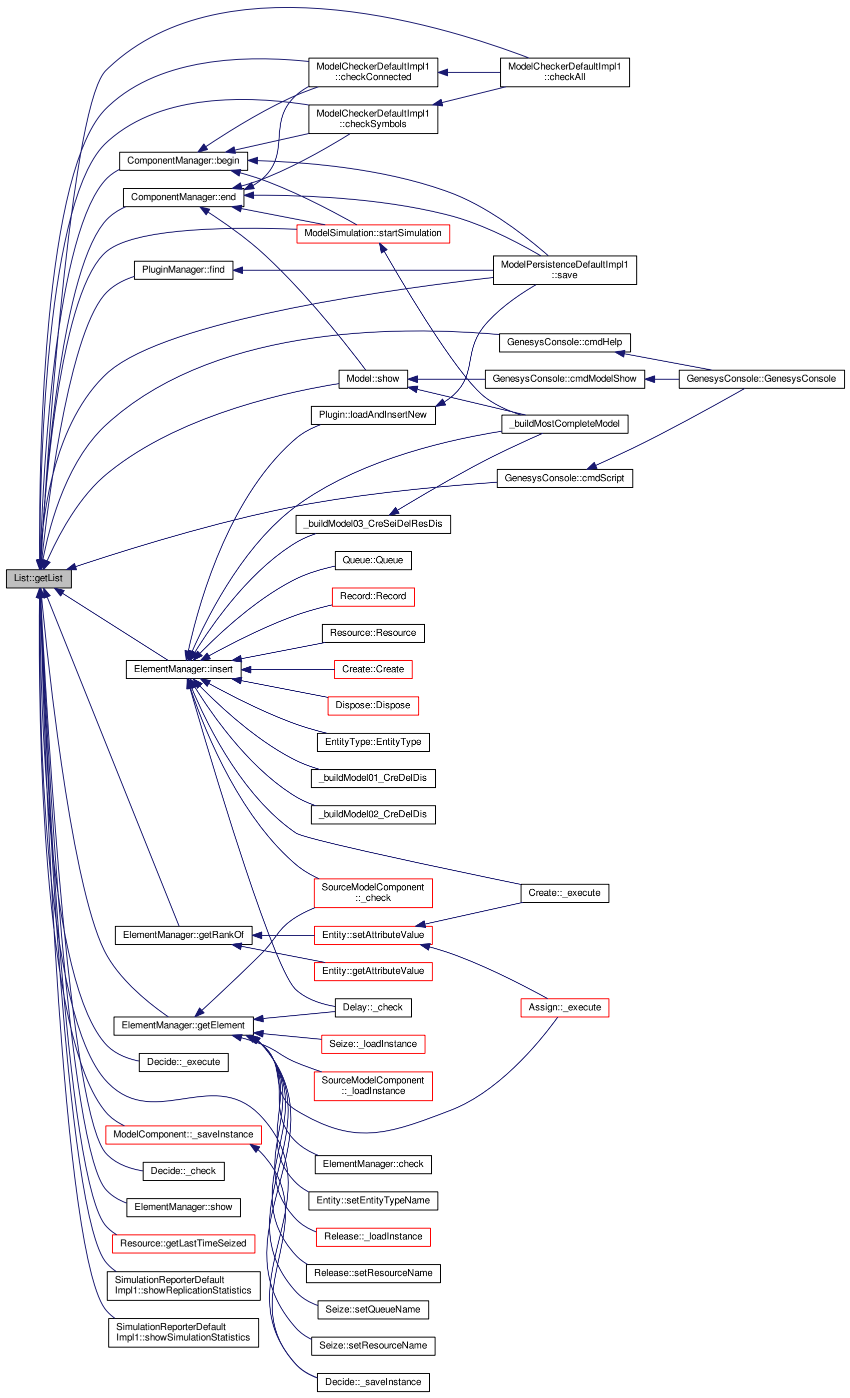
5.43.4.8 template<typename T > T List< T >::getAtRank (unsigned int *rank*)

Here is the caller graph for this function:



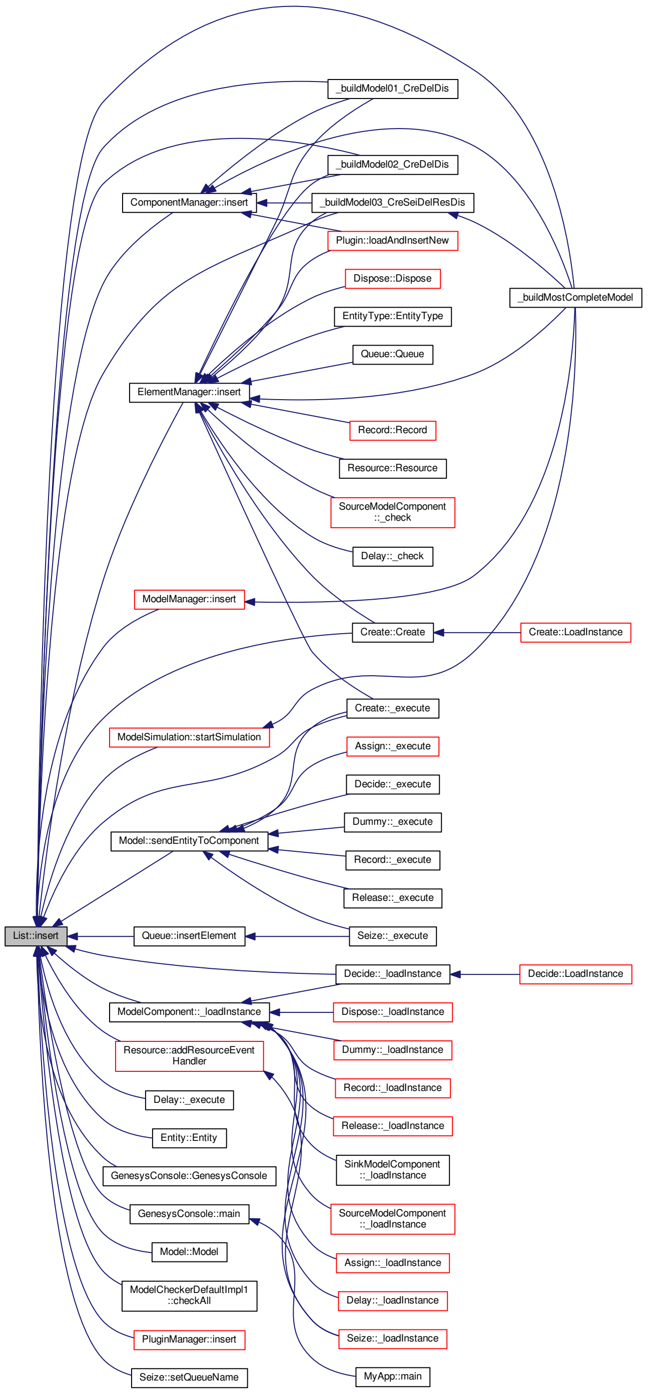
5.43.4.9 template<typename T > std::list< T > ∗ List< T >::getList ( ) const

Here is the caller graph for this function:


List::getList
ModelCheckerDefaultImpl1::checkConnected
ModelCheckerDefaultImpl1::checkAll
ModelCheckerDefaultImpl1::checkSymbols
ComponentManager::begin
ComponentManager::end
ModelSimulation::startSimulation
PluginManager::find
ModelPersistenceDefaultImpl1::save
GenesysConsole::cmdHelp
Model::show
GenesysConsole::cmdModelShow
GenesysConsole::GenesysConsole
Plugin::loadAndInsertNew
_buildMostCompleteModel
GenesysConsole::cmdScript
_buildModel03_CreSeiDelResDis
Queue::Queue
Record::Record
Resource::Resource
ElementManager::insert
Create::Create
Dispose::Dispose
EntityType::EntityType
_buildModel01_CreDelDis
_buildModel02_CreDelDis
SourceModelComponent::_check
Create::_execute
ElementManager::getRankOf
Entity::setAttributeValue
Entity::getAttributeValue
Delay::_check
Assign::_execute
ElementManager::getElement
Seize::_loadInstance
Decide::_execute
SourceModelComponent::_loadInstance
ModelComponent::_saveInstance
ElementManager::check
Decide::_check
Entity::setEntityTypeName
ElementManager::show
Release::_loadInstance
Resource::getLastTimeSeized
Release::setResourceName
SimulationReporterDefaultImpl1::showReplicationStatistics
Seize::setQueueName
SimulationReporterDefaultImpl1::showSimulationStatistics
Seize::setResourceName
Decide::_saveInstance

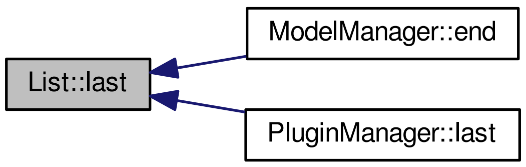
5.43.4.10 template<typename T> void List< T >::insert (T *element*)

Here is the caller graph for this function:



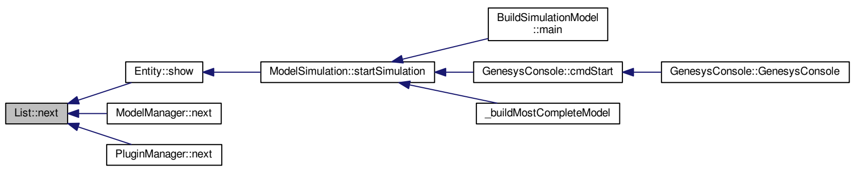
5.43.4.11 template< typename T > T List< T >::last ( )

Here is the caller graph for this function:



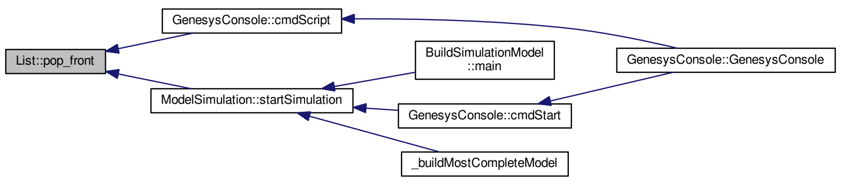
5.43.4.12 template< typename T > T List< T >::next ( )

Here is the caller graph for this function:



5.43.4.13 template< typename T > void List< T >::pop_front ( )

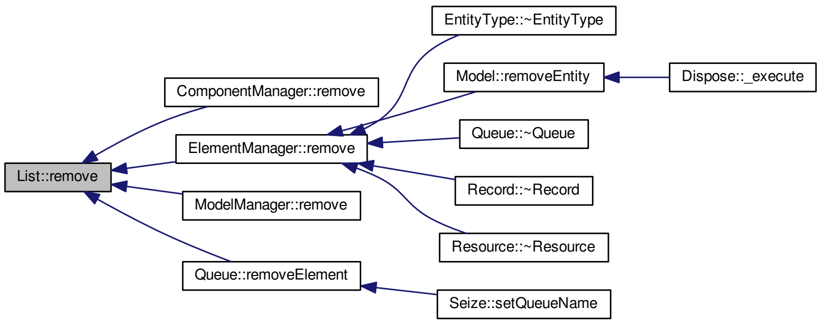
Here is the caller graph for this function:


List::pop_front
GenesysConsole::cmdScript
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

5.43.4.14 template<typename T > T List< T >::previous ( )

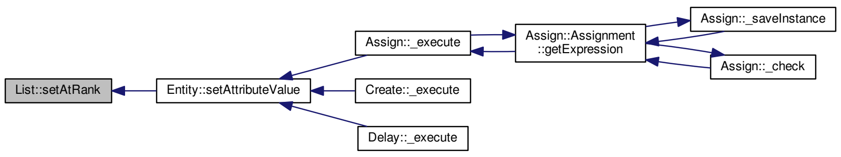
5.43.4.15 template<typename T> void List< T >::remove (T *element*)

Here is the caller graph for this function:



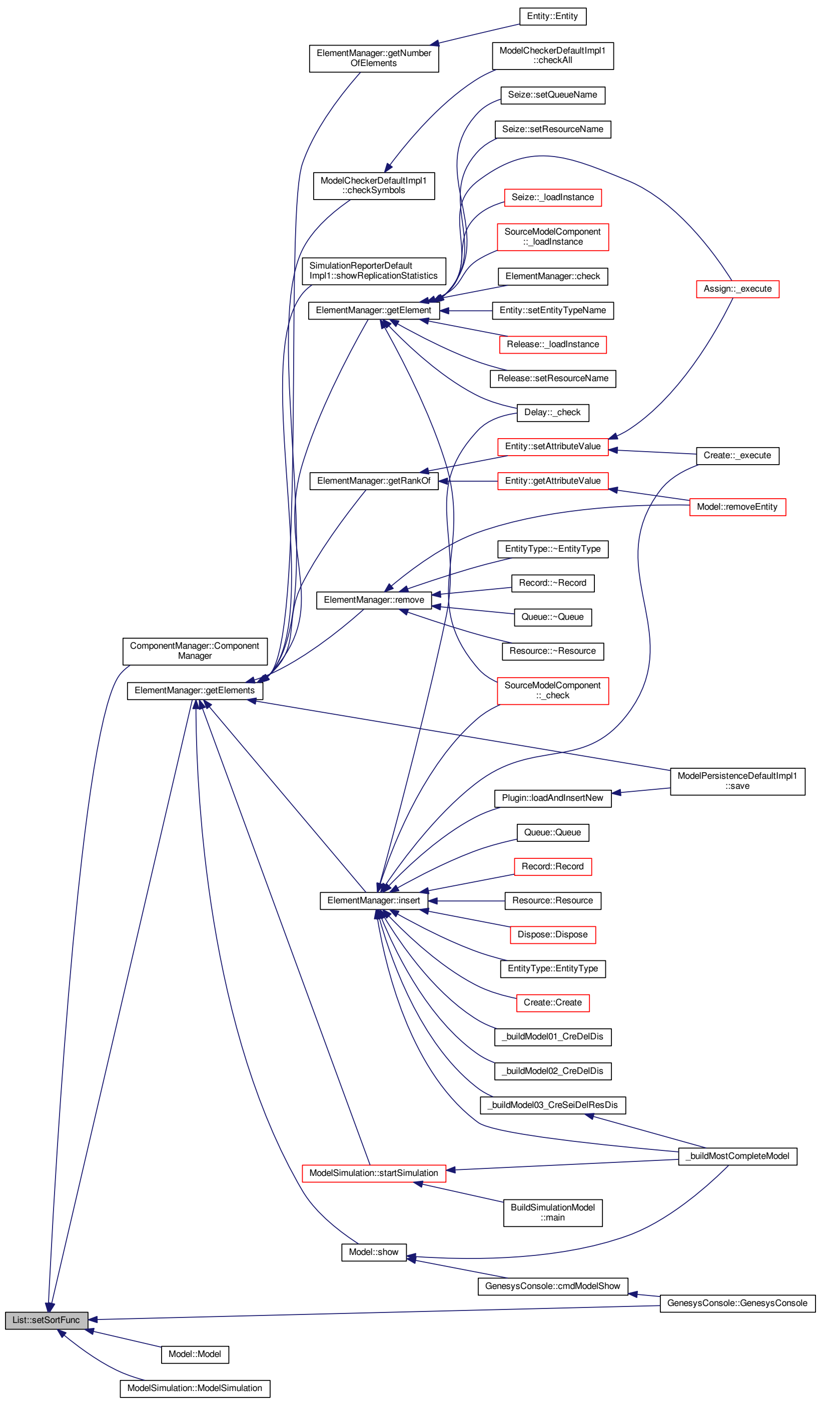
5.43.4.16 template<typename T> void List< T >::setAtRank (unsigned int *rank,* T *element*)

Here is the caller graph for this function:



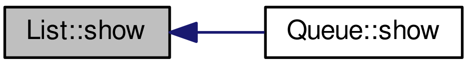
5.43.4.17 template<typename T > void List< T >::setSortFunc (CompFunct *_sortFunc*)

Here is the caller graph for this function:


Entity::Entity
ElementManager::getNumberOfElements
ModelCheckerDefaultImpl1::checkAll
Seize::setQueueName
Seize::setResourceName
ModelCheckerDefaultImpl1::checkSymbols
Seize::_loadInstance
SourceModelComponent::_loadInstance
SimulationReporterDefaultImpl1::showReplicationStatistics
ElementManager::check
Assign::_execute
ElementManager::getElement
Entity::setEntityTypeName
Release::_loadInstance
Release::setResourceName
Delay::_check
Entity::setAttributeValue
Create::_execute
ElementManager::getRankOf
Entity::getAttributeValue
Model::removeEntity
EntityType::~EntityType
Record::~Record
ElementManager::remove
Queue::~Queue
ComponentManager::ComponentManager
Resource::~Resource
SourceModelComponent::_check
ElementManager::getElements
ModelPersistenceDefaultImpl1::save
Plugin::loadAndInsertNew
Queue::Queue
Record::Record
ElementManager::insert
Resource::Resource
Dispose::Dispose
EntityType::EntityType
Create::Create
_buildModel01_CreDelDis
_buildModel02_CreDelDis
_buildModel03_CreSeiDelResDis
_buildMostCompleteModel
ModelSimulation::startSimulation
BuildSimulationModel::main
Model::show
GenesysConsole::cmdModelShow
List::setSortFunc
GenesysConsole::GenesysConsole
Model::Model
ModelSimulation::ModelSimulation

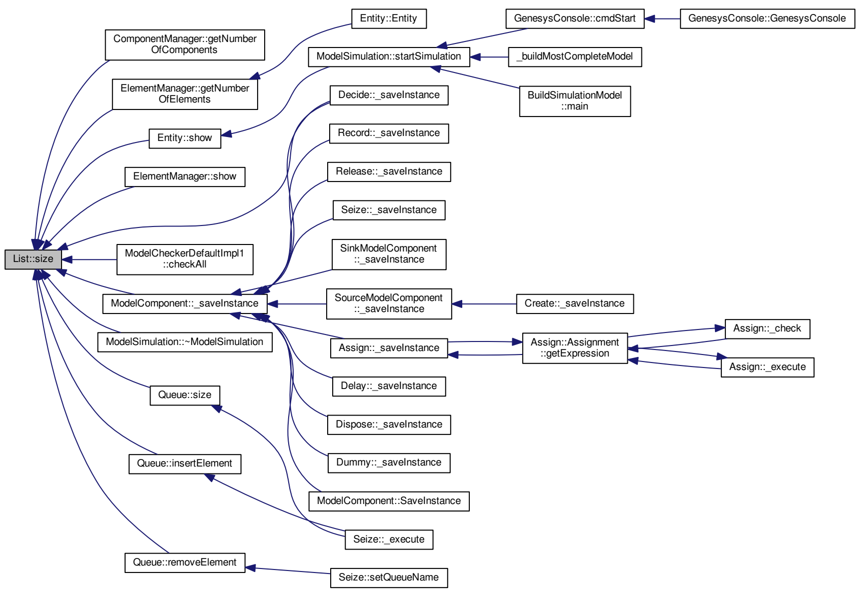
5.43.4.18 template<typename T > std::string List< T >::show ( )

Here is the caller graph for this function:



5.43.4.19 template<typename T > unsigned int List< T >::size ( )

Here is the caller graph for this function:



5.43.4.20 template<typename T > template<class Compare > void List< T >::sort (Compare *comp*)

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

- List.h

5.44 Model Class Reference

```
#include <Model.h>
```

Public Member Functions

- Model (Simulator ∗simulator)
- Model (const Model &orig)
- virtual ∼Model ()
- bool saveModel (std::string filename)
- bool loadModel (std::string filename)
- bool checkModel ()

    Checks the integrity and consistency of the model, possibly corrects some inconsistencies, and returns if the model is in position to the simulated.
- void clear ()
- void show ()
- void removeEntity (Entity ∗entity, bool collectStatistics)
- void sendEntityToComponent (Entity ∗entity, ModelComponent ∗component, double timeDelay)

    Used by components (ModelComponent) to send entities to another specific component, usually the next one connected to it, or used by the model itself, when processing an event (Event).
- double parseExpression (const std::string expression)
- double parseExpression (const std::string expression, bool ∗success, std::string ∗errorMessage)
- bool checkExpression (const std::string expression, const std::string expressionName, std::string ∗error←Message)
- Util::identitifcation getId () const
- List< SimulationControl ∗ > ∗ getControls () const

    Returns a list of values that can be externally controlled (changed). They usually correspond to input parameters in the simulation model that must be changed for an experimental design.
- List< SimulationResponse ∗ > ∗ getResponses () const

    Returns a list of exits or simulation results that can be read externally. They usually correspond to statistics resulting from the simulation that must be read for an experiment design.
- OnEventManager ∗ getOnEventManager () const
- ElementManager ∗ getElementManager () const

    Provides access to the class that manages the most basic elements of the simulation model (such as queues, resources, variables, etc.).
- ComponentManager ∗ getComponentManager () const

    The future events list chronologically sorted; Events are scheduled by components when processing other events, and a replication evolves over time by sequentially processing the very first event in this list. It's initialized with events first described by source components (SourceComponentModel).
- ModelInfo ∗ getInfos () const
- Simulator ∗ getParent () const
- ModelSimulation ∗ getSimulation () const

    Provides access to the class that manages the model simulation.
- List< Event ∗ > ∗ getEvents () const
- void setTraceManager (TraceManager ∗_traceManager)
- TraceManager ∗ getTraceManager () const

    Provides access to the class that performs the trace of simulation and replications.

5.44.1 Detailed Description

Model is probably the most important class of Genesys kernel. It represents a discrete event-driven simulation model. Each model is responsible for controlling its own simulation, ie, for sequentially processing events and collecting statistical results. A model is mainly represented by a collection of components (ModelComponent), adequately configurated and connected, and a collection of under layered element (ModelElement).

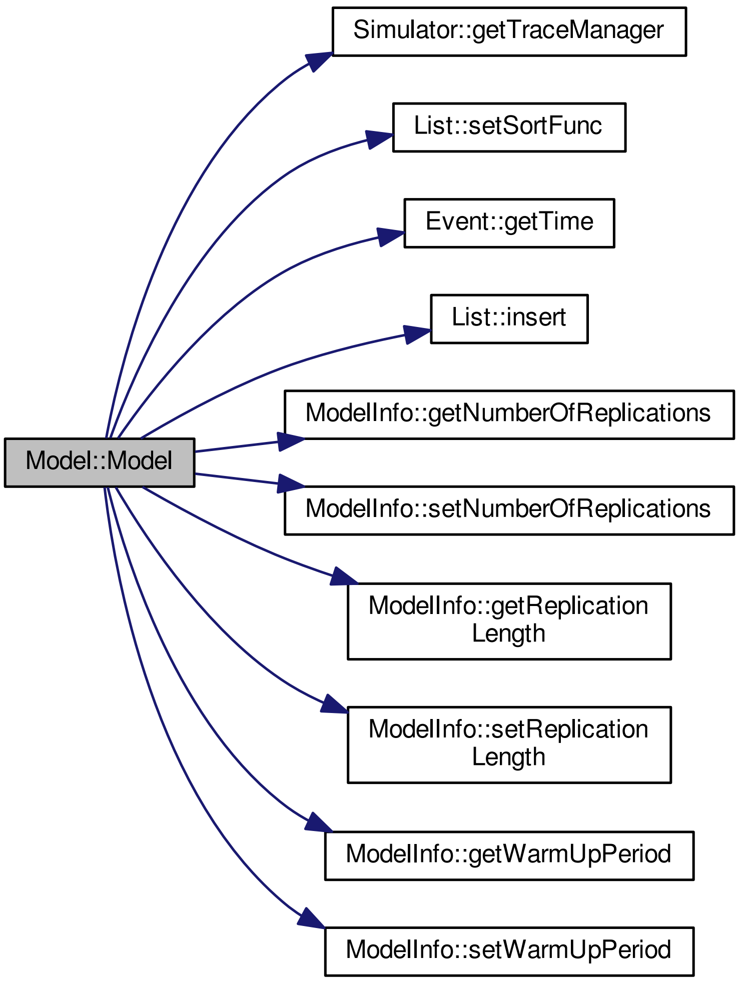
5.44.2 Constructor & Destructor Documentation

5.44.2.1 Model::Model (**Simulator** ∗ *simulator*)

The future events list must be chronologicaly sorted

Events are sorted chronologically

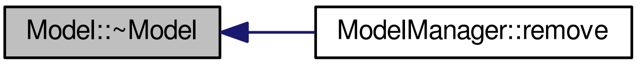
Here is the call graph for this function:


Model::Model
Simulator::getTraceManager
List::setSortFunc
Event::getTime
List::insert
ModelInfo::getNumberOfReplications
ModelInfo::setNumberOfReplications
ModelInfo::getReplication Length
ModelInfo::setReplication Length
ModelInfo::getWarmUpPeriod
ModelInfo::setWarmUpPeriod

5.44.2.2 Model::Model (const **Model** & *orig*)

5.44.2.3 Model::∼Model ( ) [virtual]

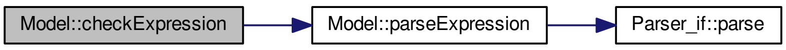
Here is the caller graph for this function:



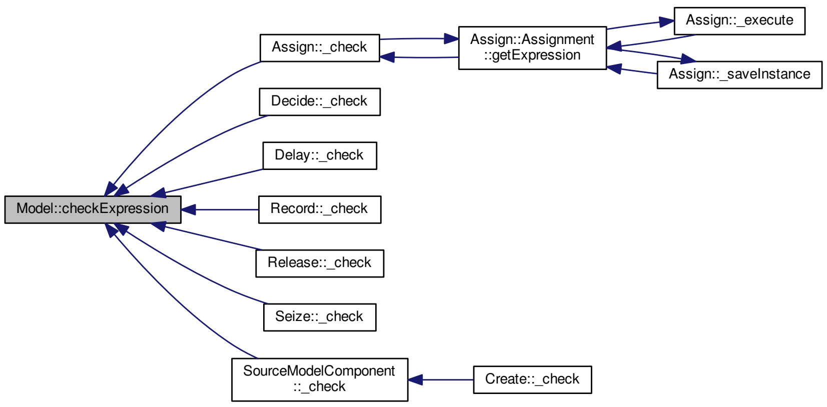
5.44.3 Member Function Documentation

5.44.3.1 bool Model::checkExpression (const std::string *expression*, const std::string *expressionName*, std::string ∗ *errorMessage*)

Here is the call graph for this function:



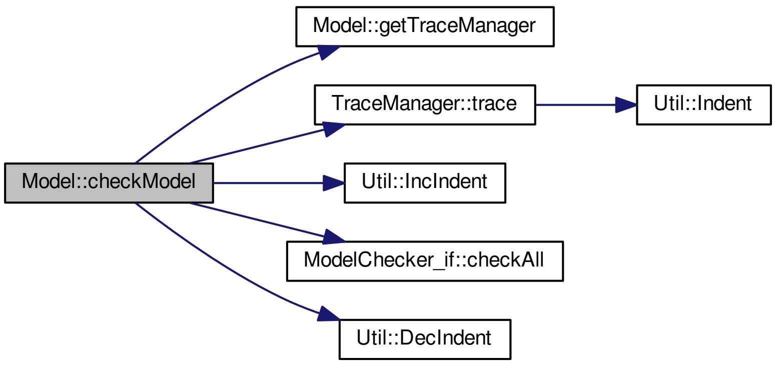
Here is the caller graph for this function:



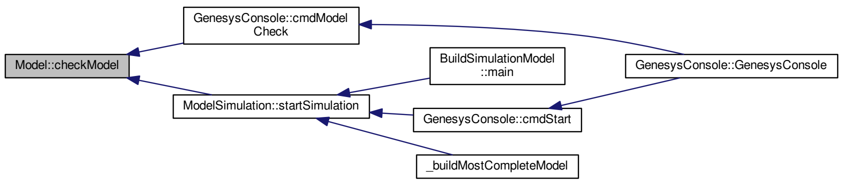
5.44.3.2 bool Model::checkModel ( )

Checks the integrity and consistency of the model, possibly corrects some inconsistencies, and returns if the model is in position to the simulated.

Here is the call graph for this function:

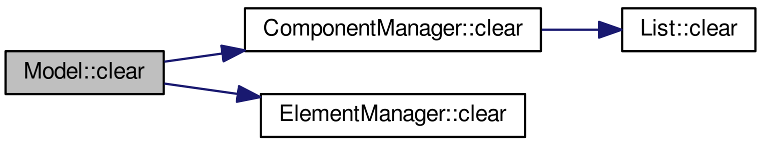


Here is the caller graph for this function:


GenesysConsole::cmdModel Check
Model::checkModel
BuildSimulationModel ::main
GenesysConsole::GenesysConsole
ModelSimulation::startSimulation
GenesysConsole::cmdStart
_buildMostCompleteModel

5.44.3.3 void Model::clear ( )

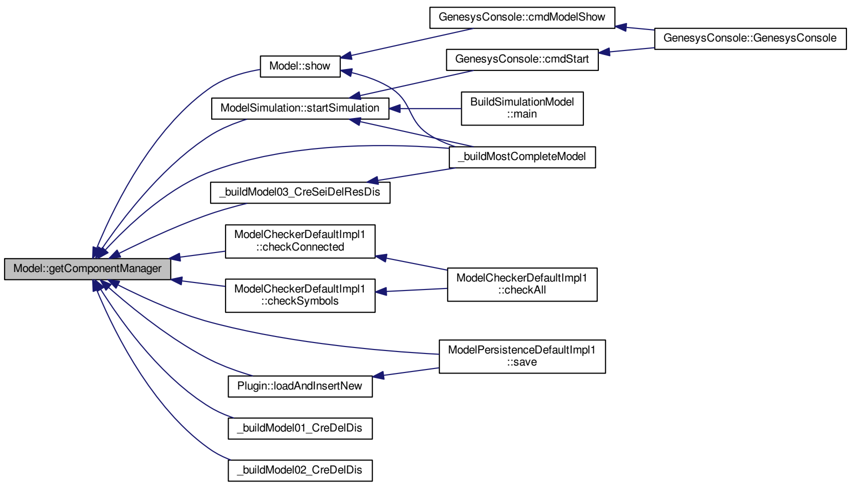
Here is the call graph for this function:



5.44.3.4 ComponentManager ∗ Model::getComponentManager ( ) const

The future events list chronologically sorted; Events are scheduled by components when processing other events, and a replication evolves over time by sequentially processing the very first event in this list. It's initialized with events first described by source components (SourceComponentModel).

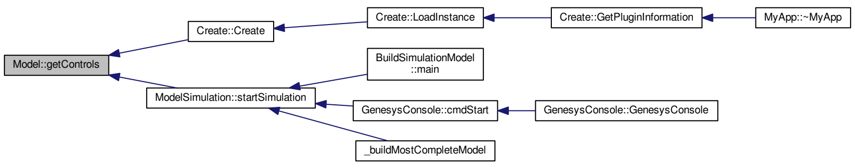
Here is the caller graph for this function:


Model::getComponentManager
Model::show
ModelSimulation::startSimulation
_buildModel03_CreSeiDelResDis
ModelCheckerDefaultImpl1 ::checkConnected
ModelCheckerDefaultImpl1 ::checkSymbols
Plugin::loadAndInsertNew
_buildModel01_CreDelDis
_buildModel02_CreDelDis
GenesysConsole::cmdModelShow
GenesysConsole::cmdStart
BuildSimulationModel ::main
_buildMostCompleteModel
ModelCheckerDefaultImpl1 ::checkAll
ModelPersistenceDefaultImpl1 ::save
GenesysConsole::GenesysConsole

5.44.3.5 List< SimulationControl ∗ > ∗ Model::getControls ( ) const

Returns a list of values that can be externally controlled (changed). They usually correspond to input parameters in the simulation model that must be changed for an experimental design.

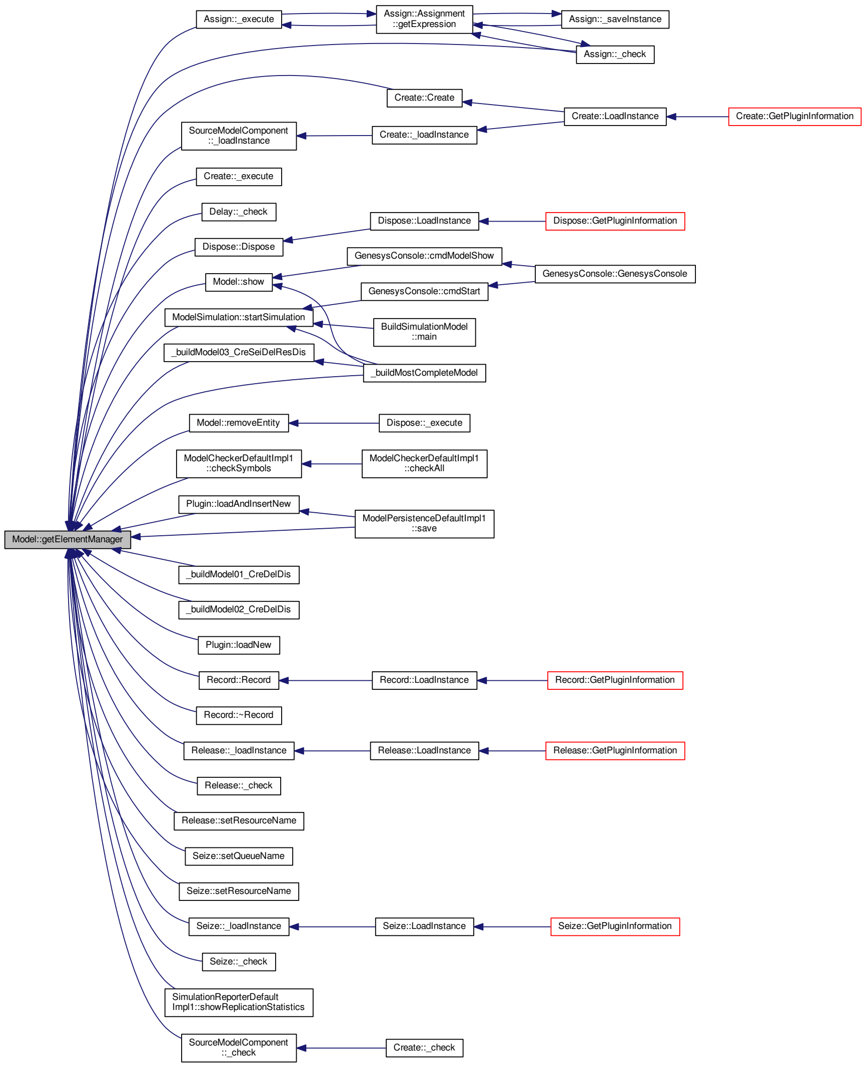
Here is the caller graph for this function:


Model::getControls
Create::Create
ModelSimulation::startSimulation
Create::LoadInstance
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
Create::GetPluginInformation
GenesysConsole::GenesysConsole
MyApp::~MyApp

5.44.3.6 ElementManager ∗ Model::getElementManager ( ) const

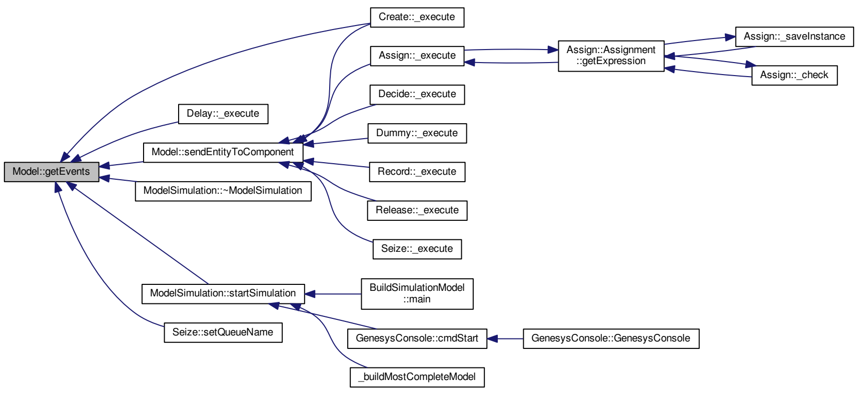
Provides access to the class that manages the most basic elements of the simulation model (such as queues, resources, variables, etc.).

Here is the caller graph for this function:



5.44.3.7 List< Event ∗ > ∗ Model::getEvents ( ) const

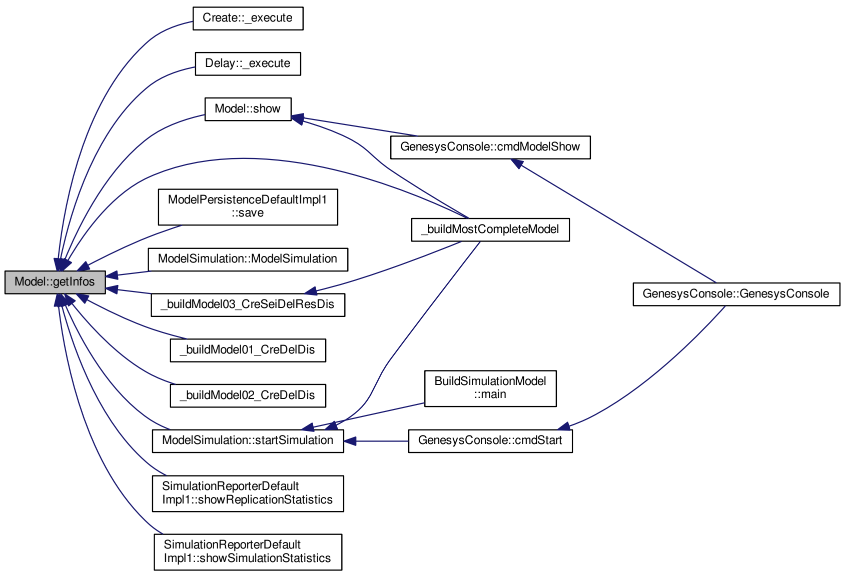
Here is the caller graph for this function:



5.44.3.8 Util::identitifcation Model::getId ( ) const

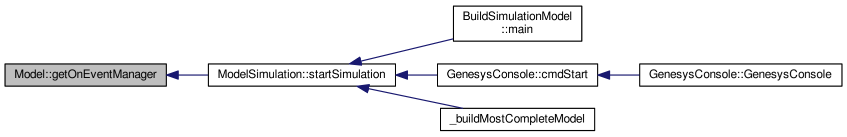
5.44.3.9 ModelInfo ∗ Model::getInfos ( ) const

Here is the caller graph for this function:


Model::getInfos
Create::_execute
Delay::_execute
Model::show
GenesysConsole::cmdModelShow
ModelPersistenceDefaultImpl1 ::save
_buildMostCompleteModel
ModelSimulation::ModelSimulation
_buildModel03_CreSeiDelResDis
GenesysConsole::GenesysConsole
_buildModel01_CreDelDis
_buildModel02_CreDelDis
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics

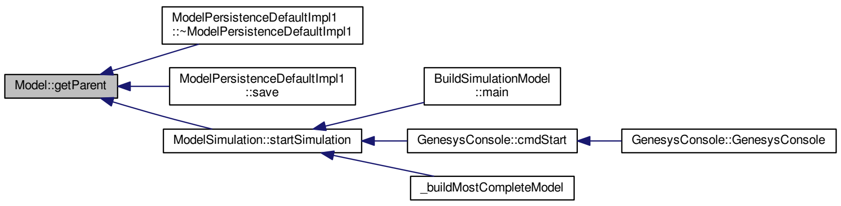
5.44.3.10 OnEventManager $*$ Model::getOnEventManager ( ) const

Here is the caller graph for this function:


Model::getOnEventManager
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.44.3.11 Simulator $*$ Model::getParent ( ) const

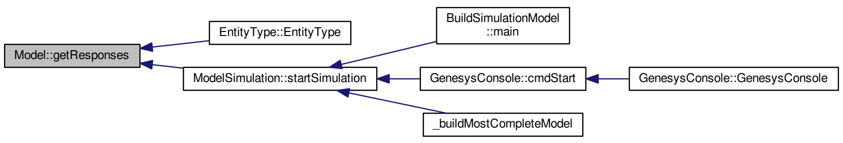
Here is the caller graph for this function:


Model::getParent
ModelPersistenceDefaultImpl1 ::~ModelPersistenceDefaultImpl1
ModelPersistenceDefaultImpl1 ::save
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.44.3.12 List$<$ SimulationResponse $*$ $>$ $*$ Model::getResponses ( ) const

Returns a list of exits or simulation results that can be read externally. They usually correspond to statistics resulting from the simulation that must be read for an experiment design.

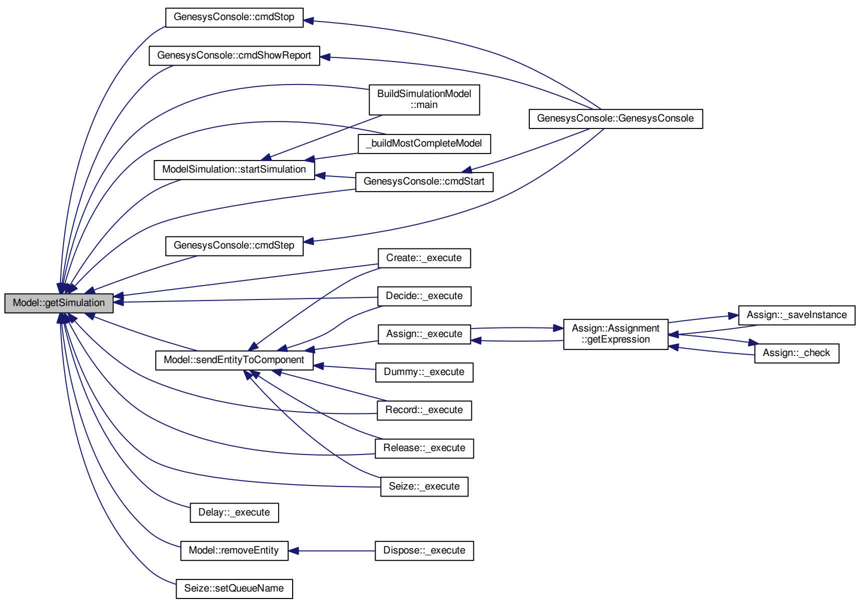
Here is the caller graph for this function:


Model::getResponses
EntityType::EntityType
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.44.3.13 ModelSimulation ∗ Model::getSimulation ( ) const

Provides access to the class that manages the model simulation.

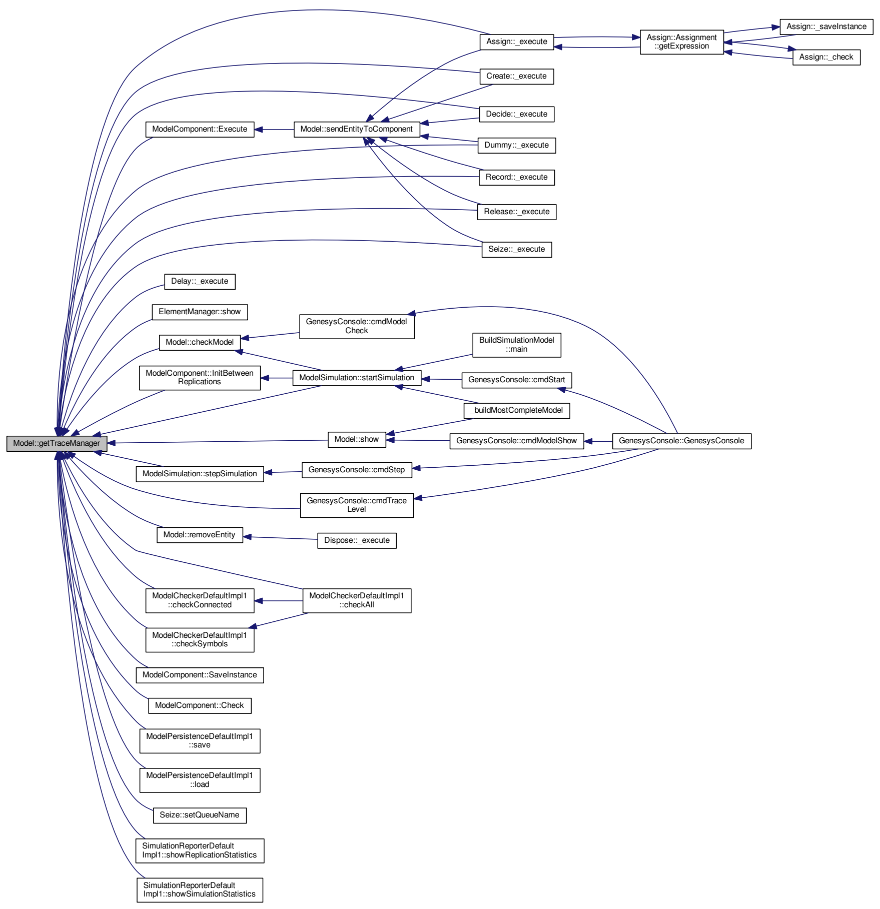
Here is the caller graph for this function:



5.44.3.14 TraceManager ∗ Model::getTraceManager ( ) const

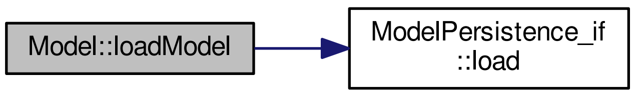
Provides access to the class that performs the trace of simulation and replications.

Here is the caller graph for this function:

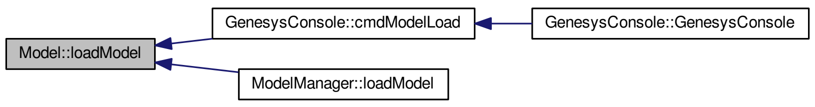

Model::getTraceManager
Assign::_execute
Assign::Assignment ::getExpression
Assign::_saveInstance
Assign::_check
Create::_execute
Decide::_execute
ModelComponent::Execute
Model::sendEntityToComponent
Dummy::_execute
Record::_execute
Release::_execute
Seize::_execute
Delay::_execute
ElementManager::show
GenesysConsole::cmdModel Check
Model::checkModel
BuildSimulationModel ::main
ModelComponent::InitBetween Replications
ModelSimulation::startSimulation
GenesysConsole::cmdStart
_buildMostCompleteModel
Model::show
GenesysConsole::cmdModelShow
GenesysConsole::GenesysConsole
ModelSimulation::stepSimulation
GenesysConsole::cmdStep
GenesysConsole::cmdTrace Level
Model::removeEntity
Dispose::_execute
ModelCheckerDefaultImpl1 ::checkConnected
ModelCheckerDefaultImpl1 ::checkAll
ModelCheckerDefaultImpl1 ::checkSymbols
ModelComponent::SaveInstance
ModelComponent::Check
ModelPersistenceDefaultImpl1 ::save
ModelPersistenceDefaultImpl1 ::load
Seize::setQueueName
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics

5.44.3.15 bool Model::loadModel (std::string *filename*)

Here is the call graph for this function:

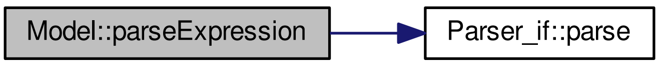


Here is the caller graph for this function:

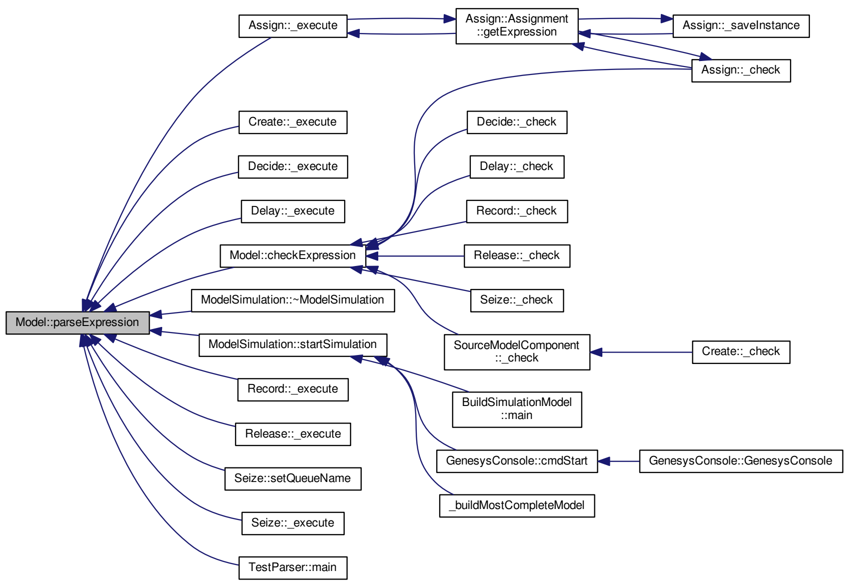


5.44.3.16 double Model::parseExpression (const std::string *expression*)

Here is the call graph for this function:

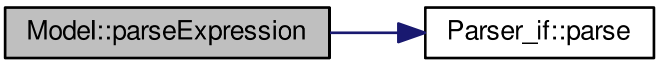


Here is the caller graph for this function:



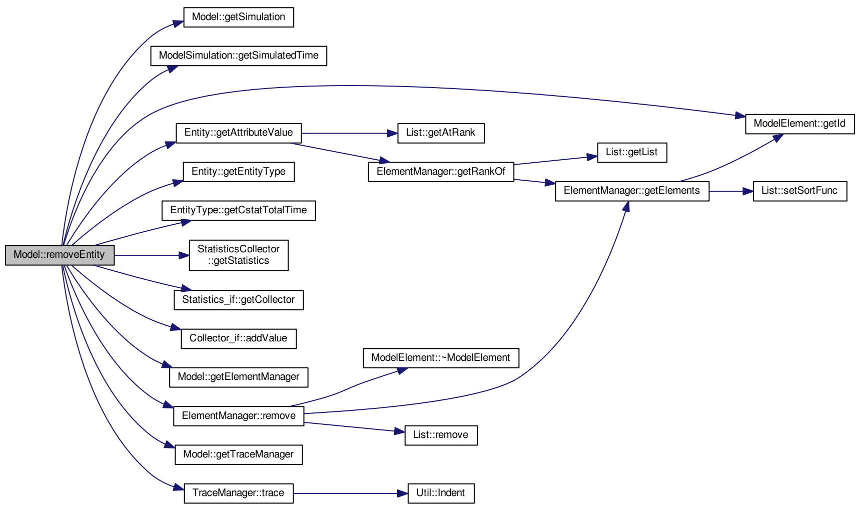
5.44.3.17 double Model::parseExpression (const std::string *expression,* bool ∗ *success,* std::string ∗ *errorMessage*)

Here is the call graph for this function:

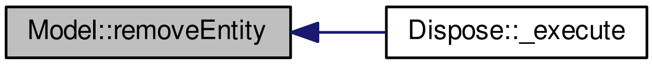


5.44.3.18 void Model::removeEntity (Entity ∗ *entity,* bool *collectStatistics*)

Here is the call graph for this function:

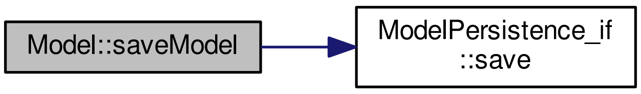


Here is the caller graph for this function:

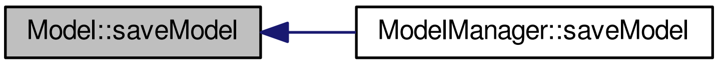


5.44.3.19 bool Model::saveModel (std::string *filename*)

Here is the call graph for this function:



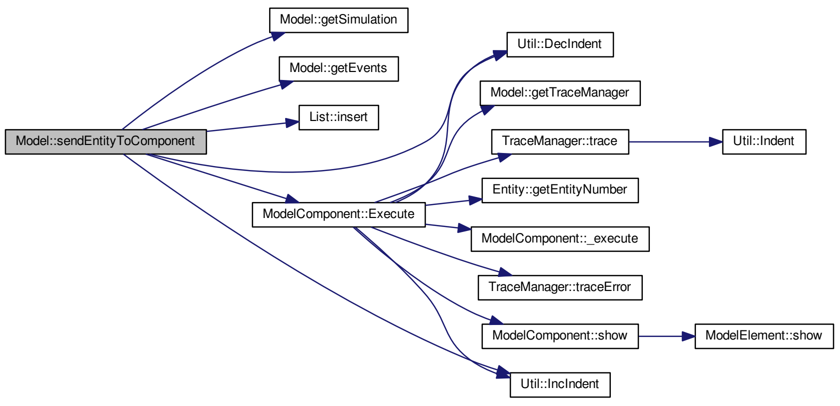
Here is the caller graph for this function:



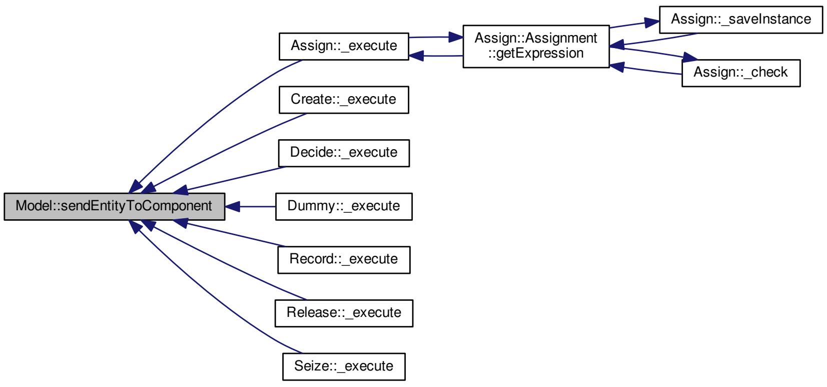
5.44.3.20 void Model::sendEntityToComponent (Entity ∗ *entity,* ModelComponent ∗ *component,* double *timeDelay*)

Used by components (ModelComponent) to send entities to another specific component, usually the next one connected to it, or used by the model itself, when processing an event (Event).

Here is the call graph for this function:



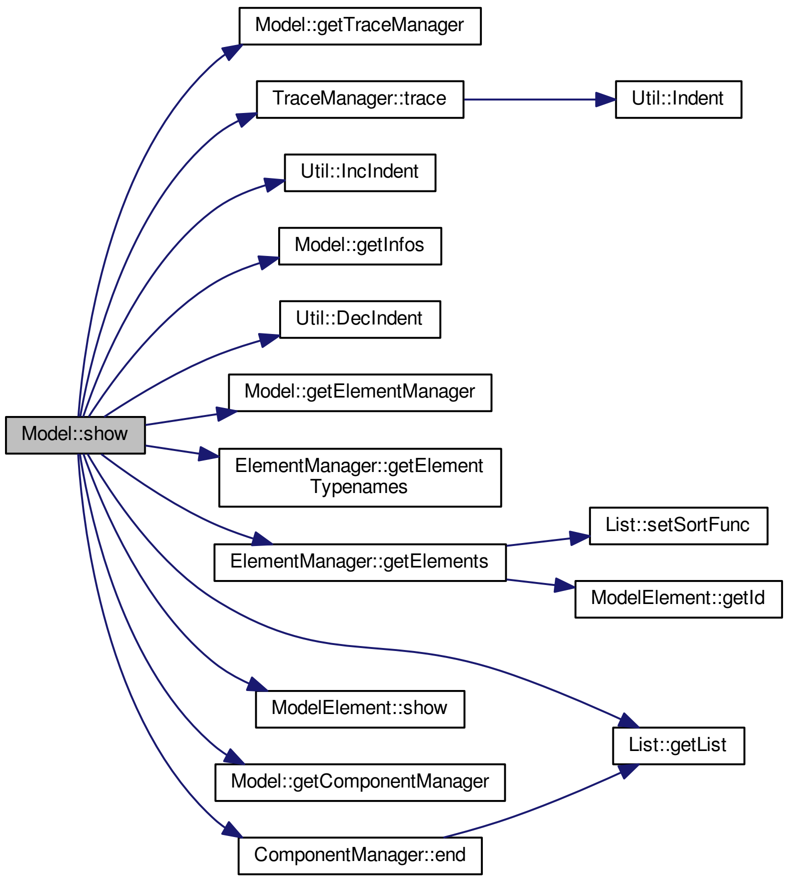
Here is the caller graph for this function:



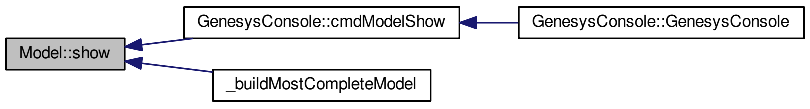
5.44.3.21 void Model::setTraceManager (**TraceManager** ∗ *_traceManager*)

5.44.3.22 void Model::show ( )

Here is the call graph for this function:



Here is the caller graph for this function:



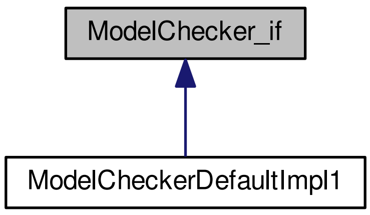
The documentation for this class was generated from the following files:

- Model.h
- Model.cpp

5.45 ModelChecker_if Class Reference

```
#include <ModelChecker_if.h>
```

Inheritance diagram for ModelChecker_if:



Public Member Functions

- virtual bool checkAll ()=0
- virtual bool checkConnected ()=0
- virtual bool checkSymbols ()=0
- virtual bool checkActivationCode ()=0

5.45.1 Detailed Description

The ModelChecker is responsable for verifying the model consistency, fixing inconsistencies wheneaver possible

5.45.2 Member Function Documentation

5.45.2.1 virtual bool ModelChecker_if::checkActivationCode ( ) [pure virtual]

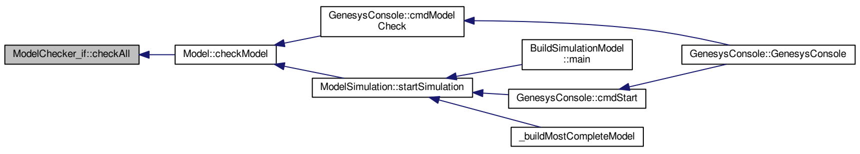
Checks if user-defined strings for symbols required by components, usually expressions or functions, are valid or references existing and valid elements.

Implemented in ModelCheckerDefaultImpl1.

5.45.2.2 virtual bool ModelChecker_if::checkAll ( ) [pure virtual]

Implemented in ModelCheckerDefaultImpl1.

Here is the caller graph for this function:


ModelChecker_if::checkAll
Model::checkModel
GenesysConsole::cmdModel Check
BuildSimulationModel ::main
GenesysConsole::GenesysConsole
ModelSimulation::startSimulation
GenesysConsole::cmdStart
_buildMostCompleteModel

5.45.2.3 virtual bool ModelChecker_if::checkConnected ( ) [pure virtual]

Invoques all other checks and returns true only if all of them returned true

Implemented in ModelCheckerDefaultImpl1.

5.45.2.4 virtual bool ModelChecker_if::checkSymbols ( ) [pure virtual]

Checks if components are consistently connected to other to form a valid process-oriented model, describing how entities proceed to the flow

Implemented in ModelCheckerDefaultImpl1.

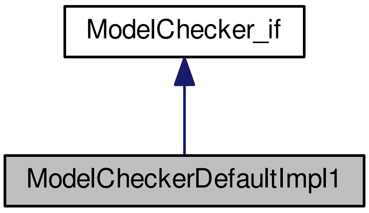
The documentation for this class was generated from the following file:

- ModelChecker_if.h

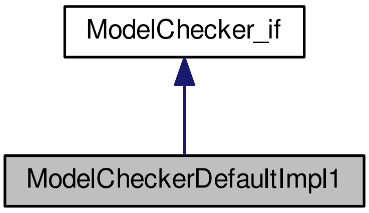
5.46 ModelCheckerDefaultImpl1 Class Reference

```
#include <ModelCheckerDefaultImpl1.h>
```

Inheritance diagram for ModelCheckerDefaultImpl1:



Collaboration diagram for ModelCheckerDefaultImpl1:



Public Member Functions

- ModelCheckerDefaultImpl1 (Model ∗model)
- ModelCheckerDefaultImpl1 (const ModelCheckerDefaultImpl1 &orig)
- virtual ∼ModelCheckerDefaultImpl1 ()
- virtual bool checkAll ()
- virtual bool checkConnected ()
- virtual bool checkSymbols ()
- virtual bool checkActivationCode ()

5.46.1 Constructor & Destructor Documentation

5.46.1.1 ModelCheckerDefaultImpl1::ModelCheckerDefaultImpl1 (**Model** ∗ *model*)

5.46.1.2 ModelCheckerDefaultImpl1::ModelCheckerDefaultImpl1 (const **ModelCheckerDefaultImpl1** & *orig*)

5.46.1.3 ModelCheckerDefaultImpl1::∼ModelCheckerDefaultImpl1 ( ) `[virtual]`

5.46.2 Member Function Documentation

5.46.2.1 bool ModelCheckerDefaultImpl1::checkActivationCode ( ) `[virtual]`

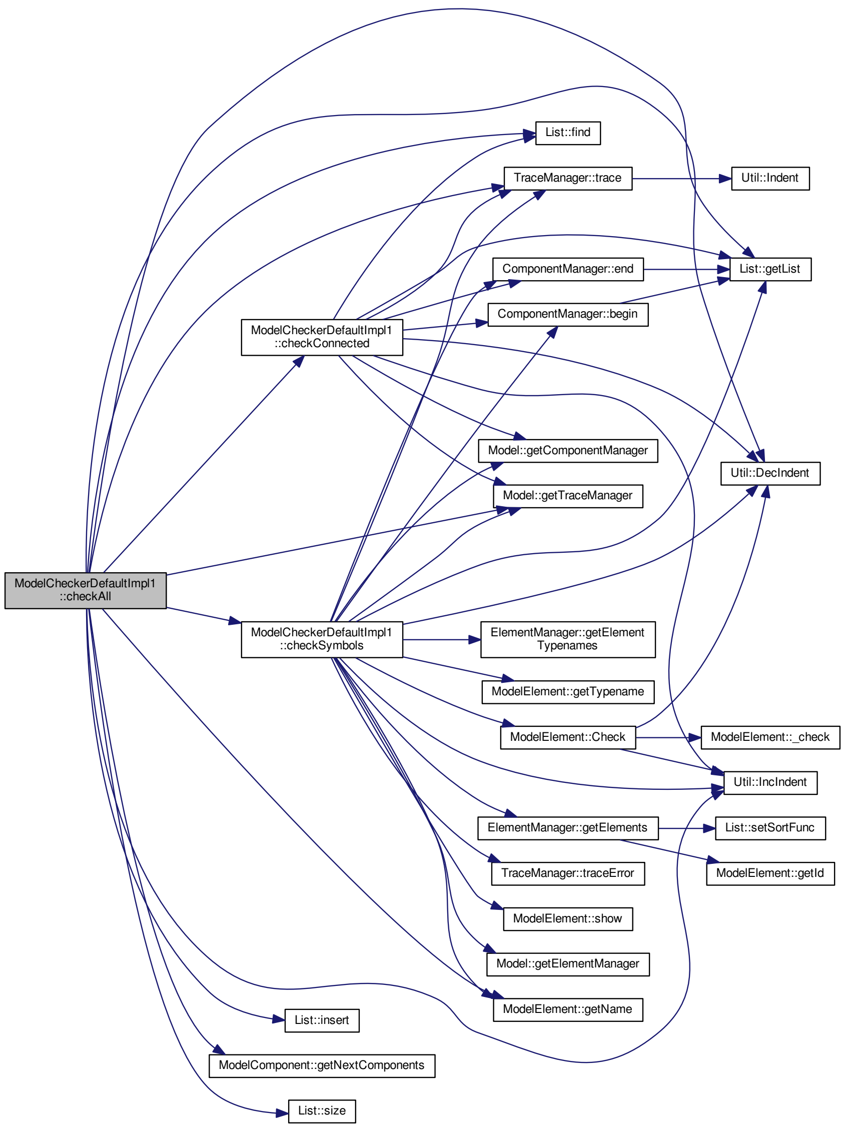
Checks if user-defined strings for symbols required by components, usually expressions or functions, are valid or references existing and valid elements.

Implements ModelChecker_if.

5.46.2.2 bool ModelCheckerDefaultImpl1::checkAll ( ) `[virtual]`

Implements ModelChecker_if.

Here is the call graph for this function:

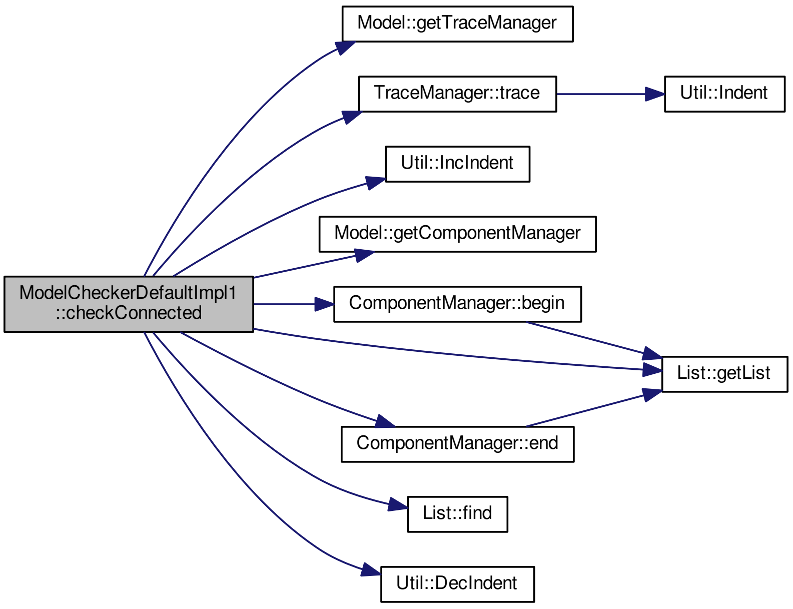


5.46.2.3 bool ModelCheckerDefaultImpl1::checkConnected ( ) [virtual]

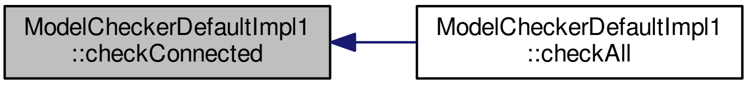
Invoques all other checks and returns true only if all of them returned true

Implements ModelChecker_if.

Here is the call graph for this function:



Here is the caller graph for this function:

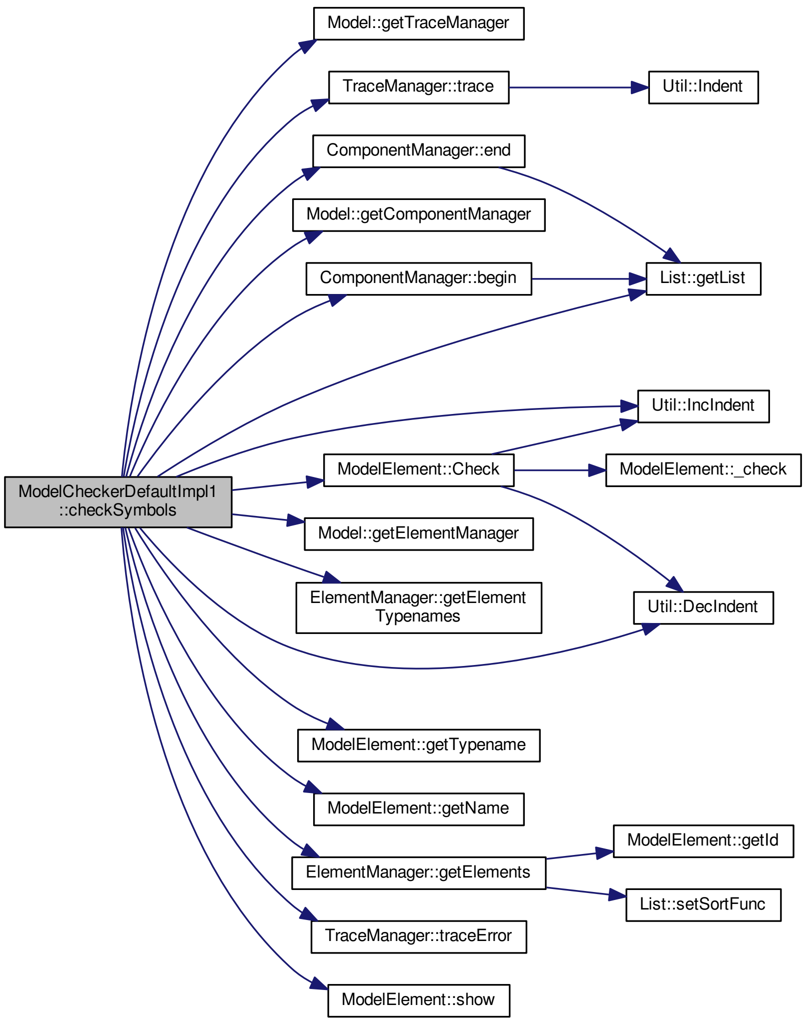

ModelCheckerDefaultImpl1 ::checkConnected
ModelCheckerDefaultImpl1 ::checkAll

5.46.2.4 bool ModelCheckerDefaultImpl1::checkSymbols ( ) [virtual]

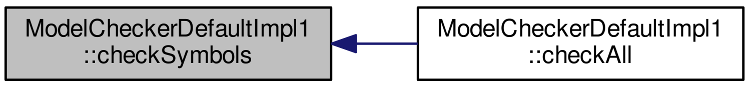
Checks if components are consistently connected to other to form a valid process-oriented model, describing how entities proceed to the flow

Implements ModelChecker_if.

Here is the call graph for this function:



Here is the caller graph for this function:


ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkAll

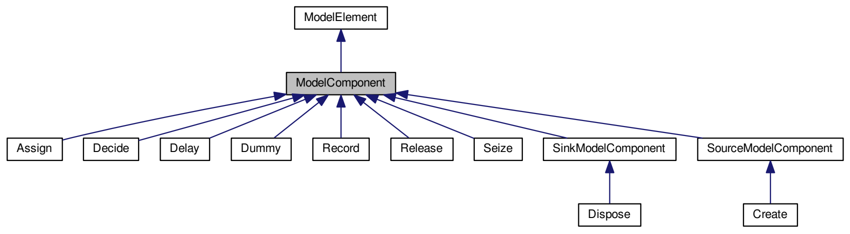
The documentation for this class was generated from the following files:

- ModelCheckerDefaultImpl1.h
- ModelCheckerDefaultImpl1.cpp

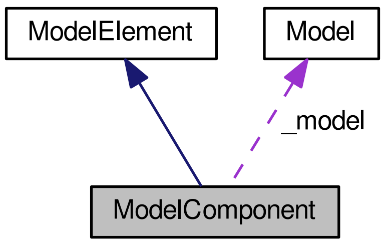
5.47 ModelComponent Class Reference

`#include <ModelComponent.h>`

Inheritance diagram for ModelComponent:



Collaboration diagram for ModelComponent:



Public Member Functions

- ModelComponent (Model ∗model, std::string componentTypename)
- ModelComponent (const ModelComponent &orig)
- virtual ∼ModelComponent ()
- virtual std::string show ()
- List< ModelComponent ∗ > ∗ getNextComponents () const

  Returns a list of components directly connected to the output. Usually the components have a single output, but they may have none (such as Dispose) or more than one (as Decide).

Static Public Member Functions

- static void Execute (Entity ∗entity, ModelComponent ∗component)

  This method triggers the simulation of the behavior of the component. It is invoked when an event (corresponding to this component) is taken from the list of future events or when an entity arrives at this component by connection.
- static void InitBetweenReplications (ModelComponent ∗component)
- static bool Check (ModelComponent ∗component)
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)
- static std::map< std::string, std::string > ∗ SaveInstance (ModelComponent ∗component)

Protected Member Functions

- virtual void _execute (Entity ∗entity)=0
- virtual void _initBetweenReplications ()=0
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)

Protected Attributes

- Model ∗ _model

5.47.1 Detailed Description

A component of the model is a block that represents a specific behavior to be simulated. The behavior is triggered when an entity arrives at the component, which corresponds to the occurrence of an event. A simulation model corresponds to a set of interconnected components to form the process by which the entity is submitted.

Parameters

model	The model this component belongs to

5.47.2 Constructor & Destructor Documentation

5.47.2.1 ModelComponent::ModelComponent (Model ∗ *model,* std::string *componentTypename*)

5.47.2.2 ModelComponent::ModelComponent (const ModelComponent & *orig*)

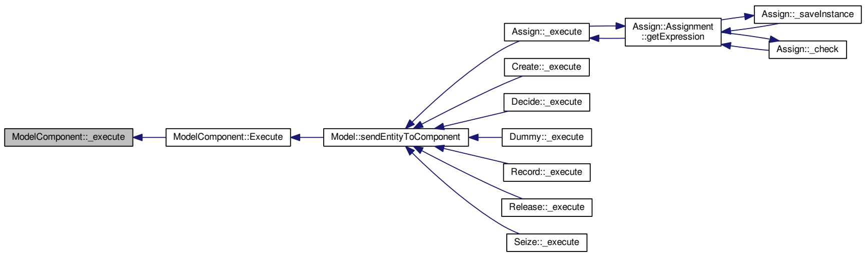
5.47.2.3 ModelComponent::∼ModelComponent ( ) [virtual]

5.47.3 Member Function Documentation

5.47.3.1 virtual void ModelComponent::_execute (Entity ∗ *entity*) [protected], [pure virtual]

Implemented in Assign, Seize, Release, Create, Record, Delay, Decide, Dispose, and Dummy.

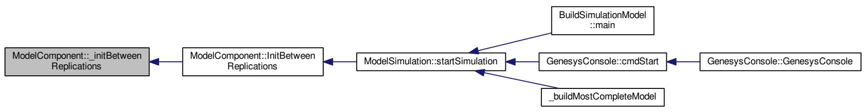
Here is the caller graph for this function:



5.47.3.2 virtual void ModelComponent::_initBetweenReplications ( ) [protected], [pure virtual]

Implemented in Assign, Seize, Release, SourceModelComponent, Create, Record, Delay, Decide, Dispose, Sink↩ModelComponent, and Dummy.

Here is the caller graph for this function:

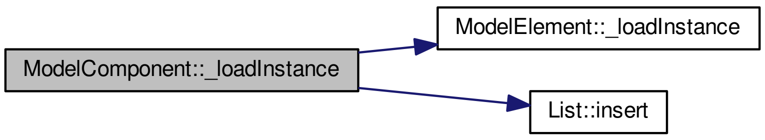

ModelComponent::_initBetween Replications
ModelComponent::InitBetween Replications
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.47.3.3 bool ModelComponent::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

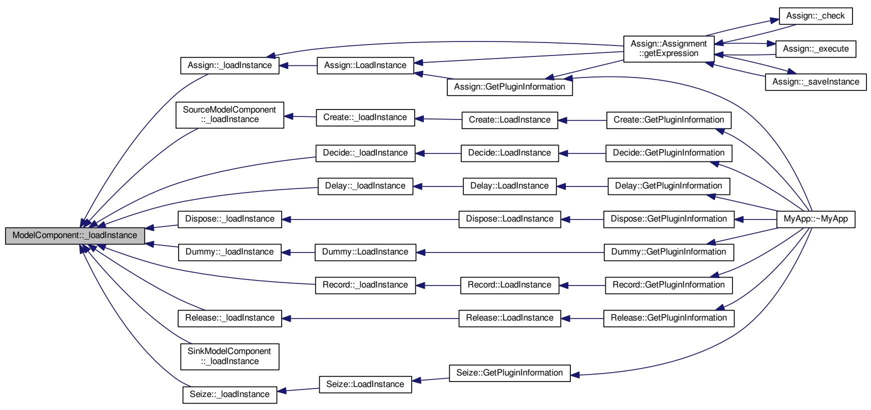
Reimplemented from ModelElement.

Reimplemented in Assign, Seize, Release, SourceModelComponent, Create, Delay, Record, Decide, Dispose, SinkModelComponent, and Dummy.

Here is the call graph for this function:



Here is the caller graph for this function:

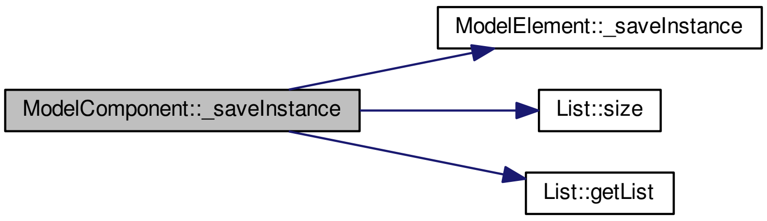


5.47.3.4 std::map< std::string, std::string > * ModelComponent::_saveInstance ( ) [protected], [virtual]

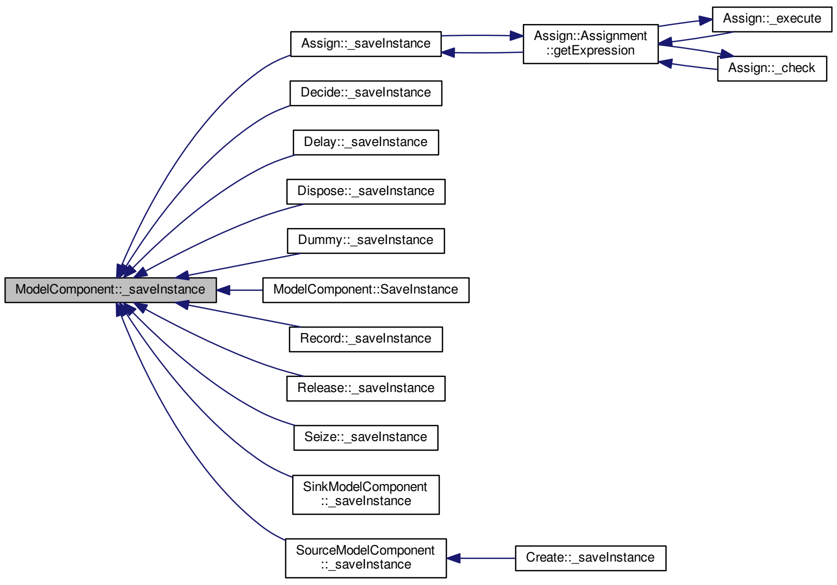
Reimplemented from ModelElement.

Reimplemented in Assign, Seize, Release, SourceModelComponent, Create, Record, Delay, Decide, Dispose, SinkModelComponent, and Dummy.

Here is the call graph for this function:

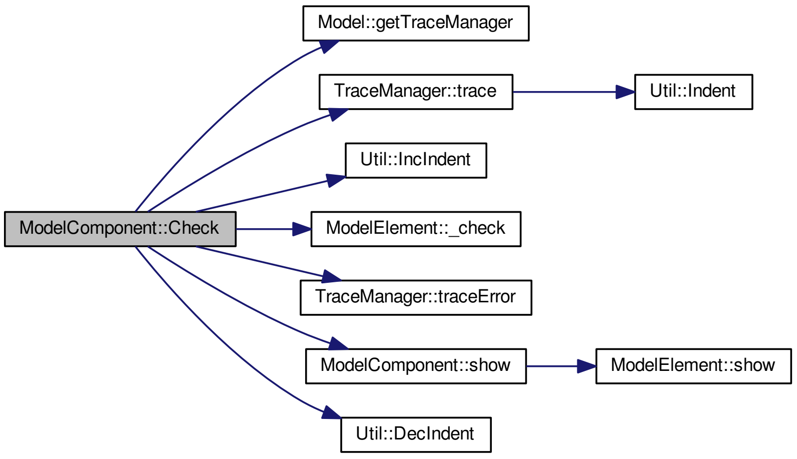


Here is the caller graph for this function:



5.47.3.5 bool ModelComponent::Check (ModelComponent ∗ *component*) [static]

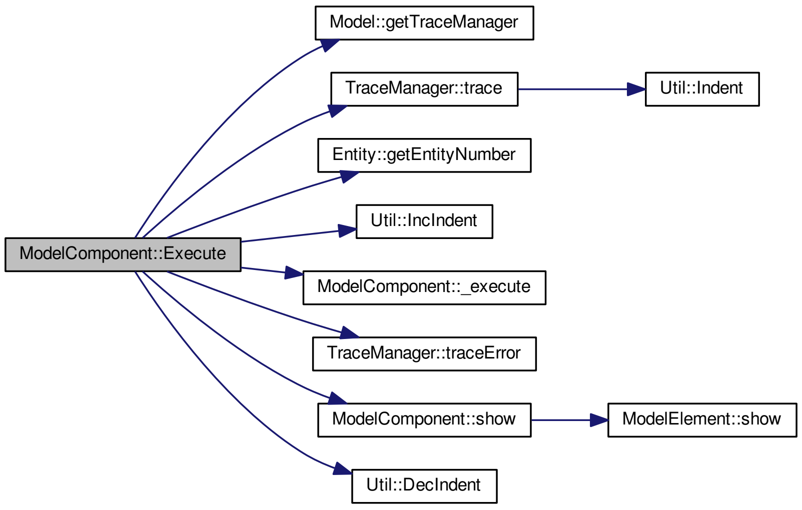
Here is the call graph for this function:



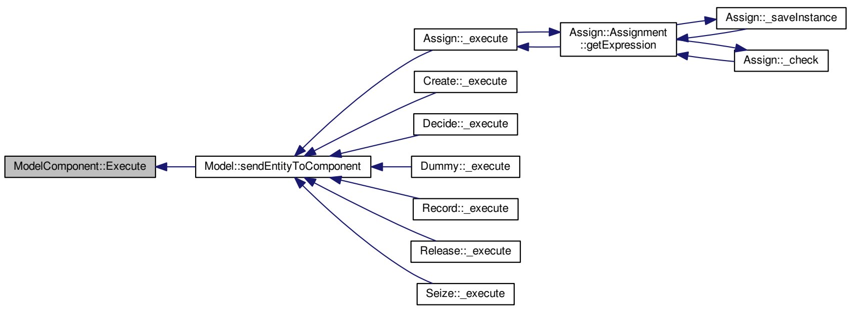
5.47.3.6 void ModelComponent::Execute (Entity ∗ *entity*, ModelComponent ∗ *component*) [static]

This method triggers the simulation of the behavior of the component. It is invoked when an event (corresponding to this component) is taken from the list of future events or when an entity arrives at this component by connection.

Here is the call graph for this function:



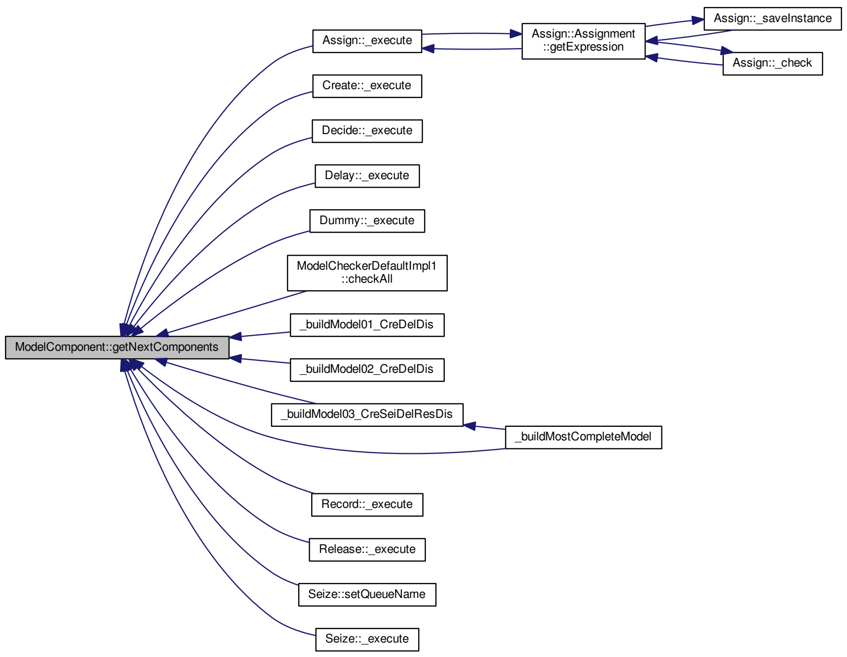
Here is the caller graph for this function:



5.47.3.7 List< ModelComponent ∗ > ∗ ModelComponent::getNextComponents ( ) const

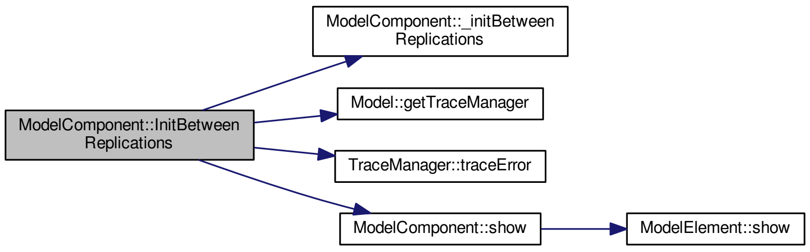
Returns a list of components directly connected to the output. Usually the components have a single output, but they may have none (such as Dispose) or more than one (as Decide).

Here is the caller graph for this function:

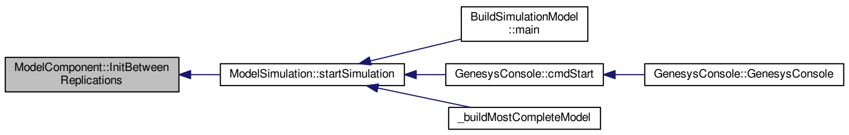


5.47.3.8 void ModelComponent::InitBetweenReplications (ModelComponent * *component*) [static]

Here is the call graph for this function:



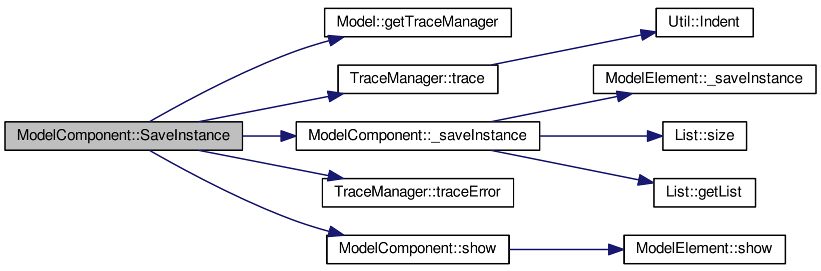
Here is the caller graph for this function:


BuildSimulationModel ::main
ModelComponent::InitBetween Replications
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.47.3.9 static ModelComponent* ModelComponent::LoadInstance (Model * *model*, std::map< std::string, std::string > * *fields*) [static]

5.47.3.10 std::map< std::string, std::string > * ModelComponent::SaveInstance (ModelComponent * *component*) [static]

Here is the call graph for this function:

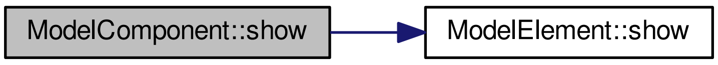


5.47.3.11 std::string ModelComponent::show ( ) [virtual]

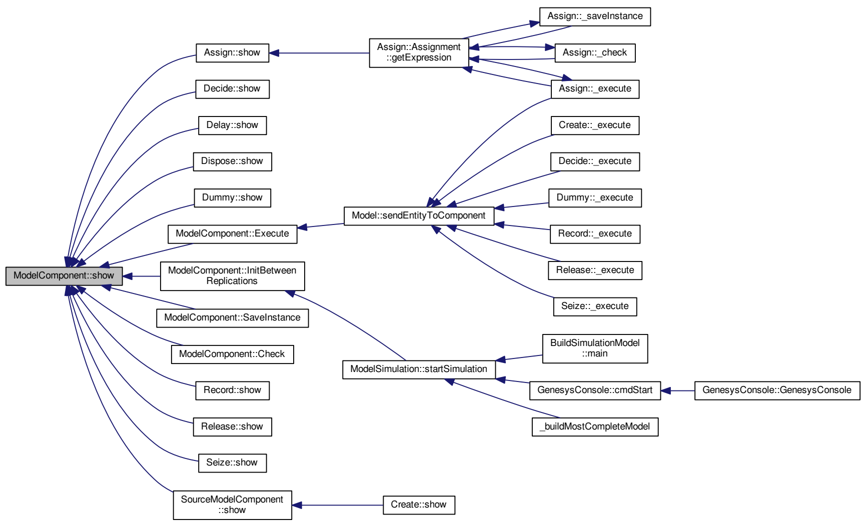
Reimplemented from ModelElement.

Reimplemented in Assign, SourceModelComponent, Create, Record, Seize, Delay, Release, Decide, Dispose, and Dummy.

Here is the call graph for this function:



Here is the caller graph for this function:



5.47.4 Member Data Documentation

5.47.4.1 Model∗ ModelComponent::_model [protected]

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

- ModelComponent.h
- ModelComponent.cpp

5.48 ModelComponentManager_if Class Reference

```
#include <ModelComponentManager_if.h>
```

Public Member Functions

- ModelComponentManager_if ()
- ModelComponentManager_if (const ModelComponentManager_if &orig)
- virtual ∼ModelComponentManager_if ()

5.48.1 Constructor & Destructor Documentation

5.48.1.1 ModelComponentManager_if::ModelComponentManager_if ( )

5.48.1.2 ModelComponentManager_if::ModelComponentManager_if (const **ModelComponentManager_if** & *orig*)

5.48.1.3 virtual ModelComponentManager_if::∼ModelComponentManager_if ( ) `[virtual]`

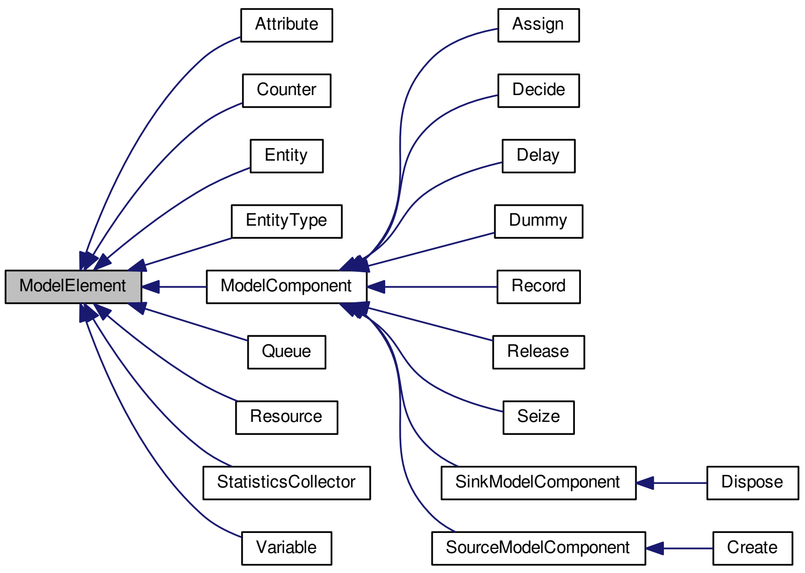
The documentation for this class was generated from the following file:

- ModelComponentManager_if.h

5.49 ModelElement Class Reference

```
#include <ModelElement.h>
```

Inheritance diagram for ModelElement:



Public Member Functions

- ModelElement (std::string elementTypename)
- ModelElement (const ModelElement &orig)
- virtual ∼ModelElement ()
- virtual std::string show ()
- Util::identitifcation getId () const
- void setName (std::string _name)
- std::string getName () const
- std::string getTypename () const

Static Public Member Functions

- static ModelElement ∗ LoadInstance (std::map< std::string, std::string > ∗fields)
- static std::map< std::string, std::string > ∗ SaveInstance (ModelElement ∗element)
- static bool Check (ModelElement ∗element, std::string ∗errorMessage)

Protected Member Functions

- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Protected Attributes

- Util::identitifcation _id
- std::string _name
- std::string _typename

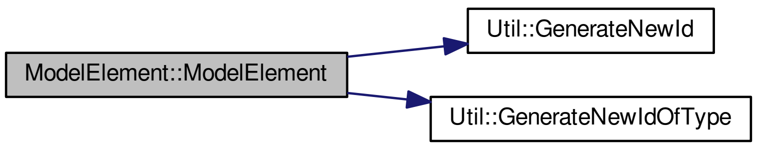
5.49.1 Detailed Description

This class is the basis for any element of the model (such as Queue, Resource, Variable, etc.) and also for any component of the model. It has the infrastructure to read and write on file and to verify symbols.

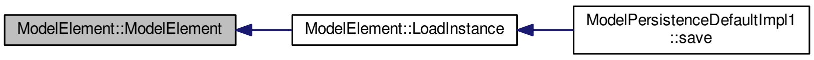
5.49.2 Constructor & Destructor Documentation

5.49.2.1 ModelElement::ModelElement (std::string *elementTypename*)

Here is the call graph for this function:



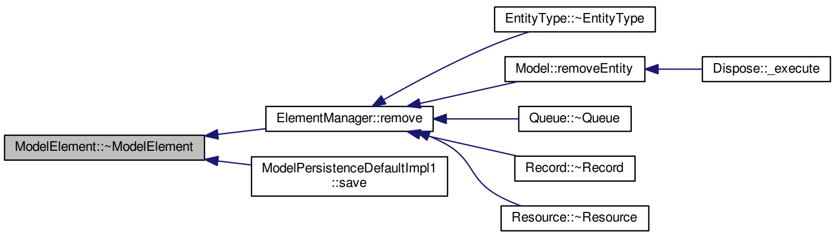
Here is the caller graph for this function:


ModelElement::ModelElement
ModelElement::LoadInstance
ModelPersistenceDefaultImpl1 ::save

5.49.2.2 ModelElement::ModelElement (const **ModelElement** & *orig*)

5.49.2.3 ModelElement::∼ModelElement ( ) [virtual]

Here is the caller graph for this function:

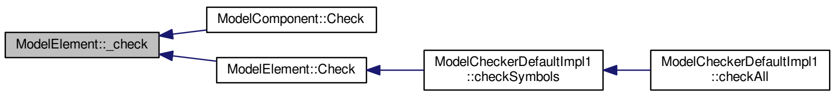


5.49.3 Member Function Documentation

5.49.3.1 bool ModelElement::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented in Assign, Resource, Seize, Queue, EntityType, Release, SourceModelComponent, Entity, Counter, Create, Record, Delay, StatisticsCollector, Variable, Attribute, Decide, Dispose, SinkModelComponent, and Dummy.

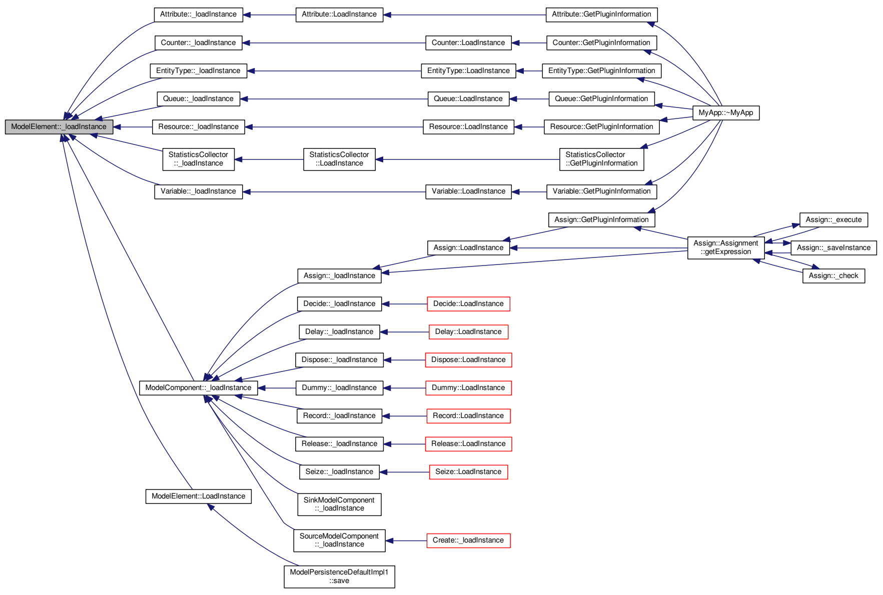
Here is the caller graph for this function:


ModelComponent::Check
ModelElement::_check
ModelElement::Check
ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkAll

5.49.3.2 bool ModelElement::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented in Assign, Resource, Seize, Queue, EntityType, ModelComponent, Release, SourceModel←Component, Entity, Counter, Create, Delay, Record, StatisticsCollector, Variable, Attribute, Decide, Dispose, Sink←ModelComponent, and Dummy.

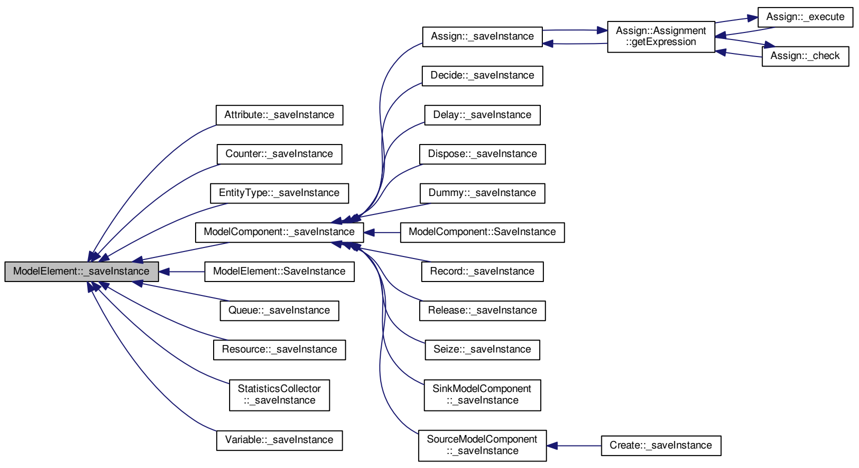
Here is the caller graph for this function:



5.49.3.3 std::map< std::string, std::string > * ModelElement::_saveInstance ( ) [protected], [virtual]

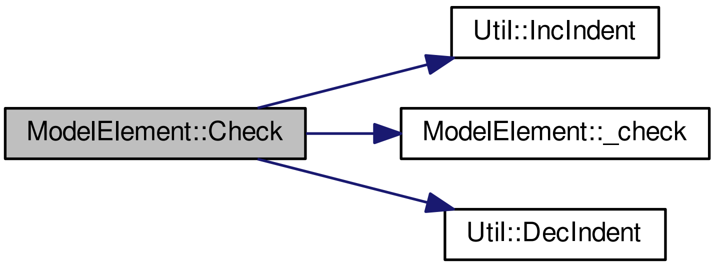
Reimplemented in Assign, Resource, Seize, Queue, EntityType, Release, ModelComponent, SourceModel←Component, Entity, Create, Record, Counter, Delay, StatisticsCollector, Variable, Attribute, Decide, Dispose, Sink←ModelComponent, and Dummy.

Here is the caller graph for this function:

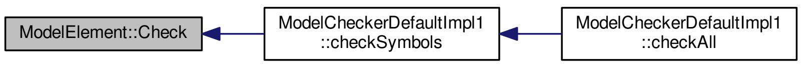


5.49.3.4 bool ModelElement::Check (ModelElement ∗ *element*, std::string ∗ *errorMessage*) [static]

Here is the call graph for this function:

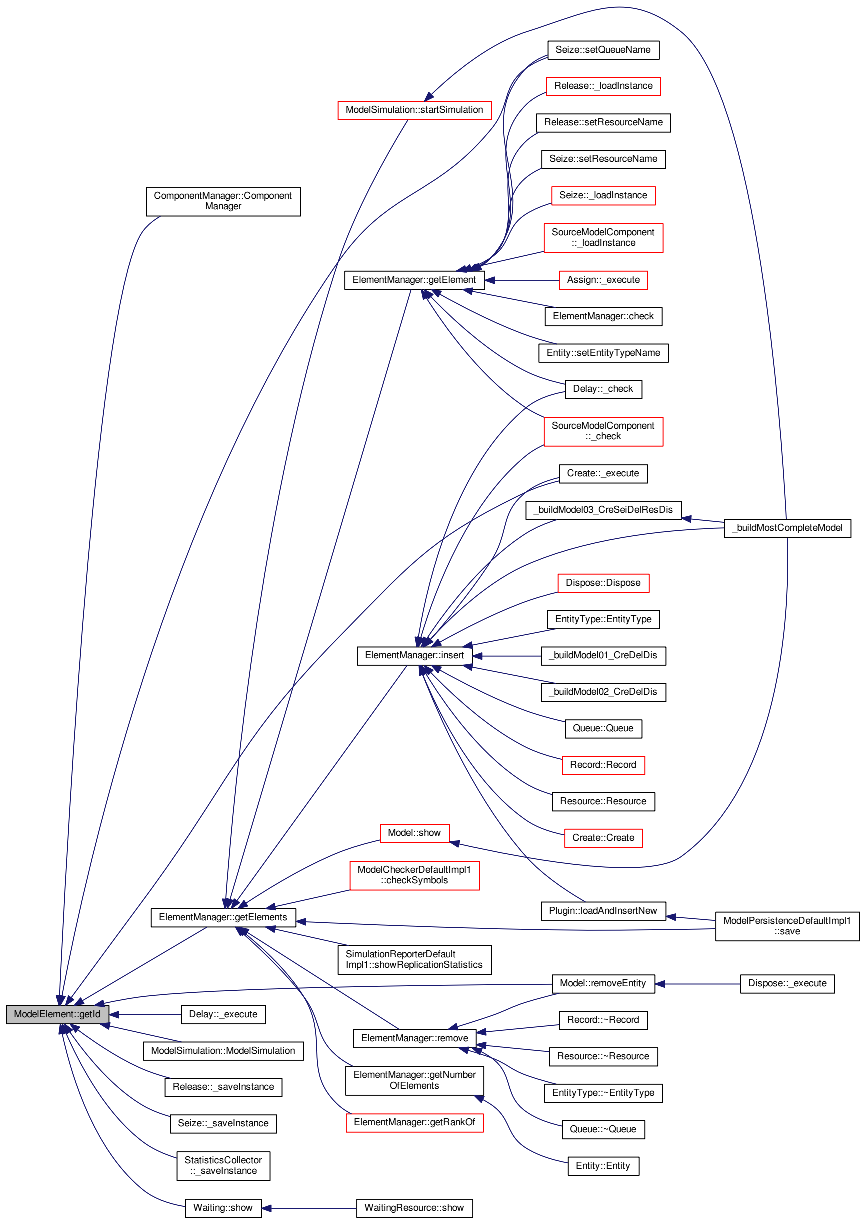


Here is the caller graph for this function:


ModelElement::Check
ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkAll

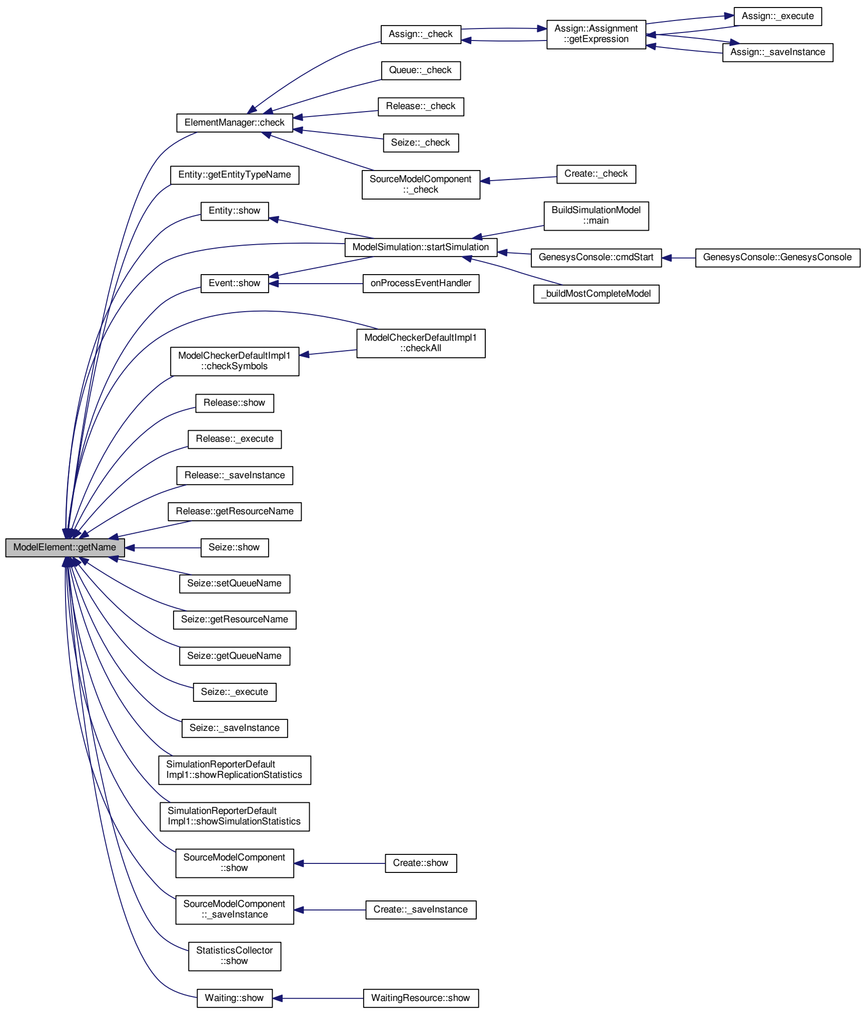
5.49.3.5 Util::identitifcation ModelElement::getId ( ) const

Here is the caller graph for this function:



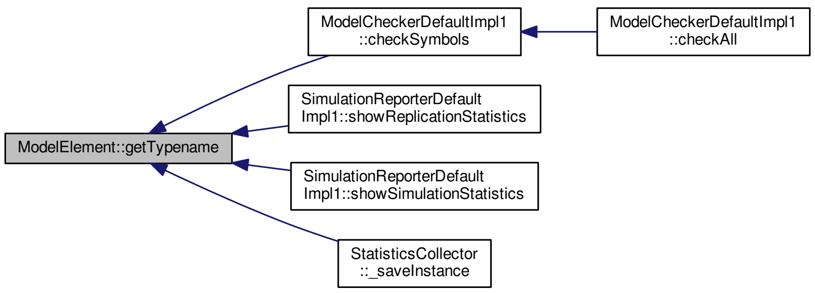
5.49.3.6 std::string ModelElement::getName ( ) const

Here is the caller graph for this function:



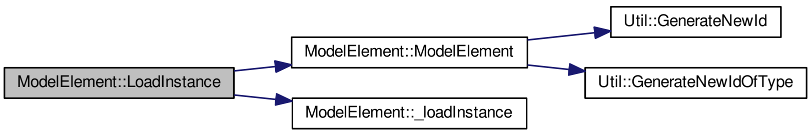
5.49.3.7 std::string ModelElement::getTypename ( ) const

Here is the caller graph for this function:

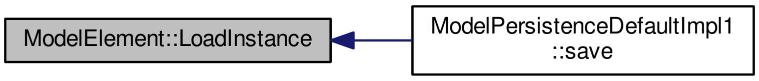

ModelElement::getTypename
ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkAll
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
StatisticsCollector ::_saveInstance

5.49.3.8 ModelElement $*$ ModelElement::LoadInstance (std::map$<$ std::string, std::string $>$ $*$ *fields*) [static]

Here is the call graph for this function:

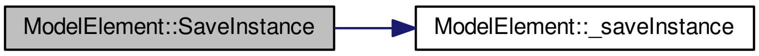

ModelElement::LoadInstance
ModelElement::ModelElement
ModelElement::_loadInstance
Util::GenerateNewId
Util::GenerateNewIdOfType

Here is the caller graph for this function:


ModelElement::LoadInstance
ModelPersistenceDefaultImpl1 ::save

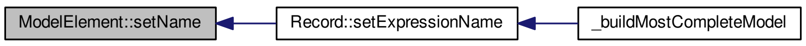
5.49.3.9 std::map$<$ std::string, std::string $>$ $*$ ModelElement::SaveInstance (ModelElement $*$ *element*) [static]

Here is the call graph for this function:



5.49.3.10 void ModelElement::setName (std::string *_name*)

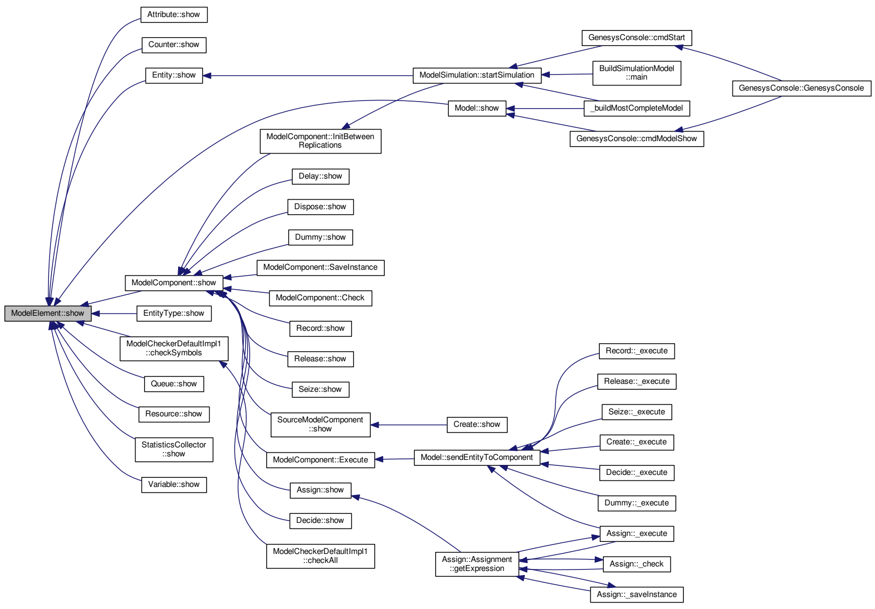
Here is the caller graph for this function:



5.49.3.11 std::string ModelElement::show ( ) [virtual]

Reimplemented in Assign, Resource, SourceModelComponent, Queue, ModelComponent, Create, Record, Seize, Delay, EntityType, Attribute, Entity, StatisticsCollector, Counter, Release, Variable, Decide, Dispose, and Dummy.

Here is the caller graph for this function:



5.49.4 Member Data Documentation

5.49.4.1 Util::identitifcation ModelElement::_id [protected]

5.49.4.2 std::string ModelElement::_name [protected]

5.49.4.3 std::string ModelElement::_typename [protected]

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

- ModelElement.h
- ModelElement.cpp

5.50 ModelInfo Class Reference

```
#include <ModelInfo.h>
```

Public Member Functions

- ModelInfo ()
- ModelInfo (const ModelInfo &orig)
- virtual ∼ModelInfo ()
- std::string show ()
- void setName (std::string _name)
- std::string getName () const
- void setAnalystName (std::string _analystName)
- std::string getAnalystName () const
- void setDescription (std::string _description)
- std::string getDescription () const
- void setProjectTitle (std::string _projectTitle)
- std::string getProjectTitle () const
- void setVersion (std::string _version)
- std::string getVersion () const
- void setNumberOfReplications (unsigned int _numberOfReplications)
- unsigned int getNumberOfReplications () const
- void setReplicationLength (double _replicationLength)
- double getReplicationLength () const
- void setReplicationLengthTimeUnit (Util::TimeUnit _replicationLengthTimeUnit)
- Util::TimeUnit getReplicationLengthTimeUnit () const
- void setWarmUpPeriod (double _warmUpPeriod)
- double getWarmUpPeriod () const
- void setWarmUpPeriodTimeUnit (Util::TimeUnit _warmUpPeriodTimeUnit)
- Util::TimeUnit getWarmUpPeriodTimeUnit () const
- void setTerminatingCondition (std::string _terminatingCondition)
- std::string getTerminatingCondition () const
- void loadInstance (std::map< std::string, std::string > ∗fields)
- std::map< std::string, std::string > ∗ saveInstance ()

5.50.1 Detailed Description

ModelInfo stores basic model project information.

5.50.2 Constructor & Destructor Documentation

5.50.2.1 ModelInfo::ModelInfo ( )

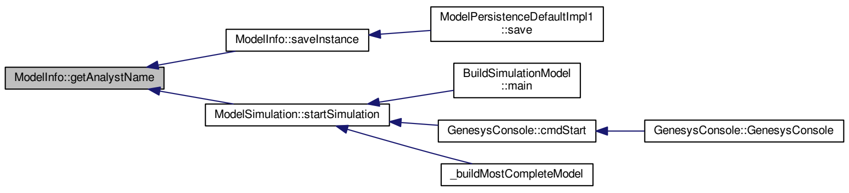
5.50.2.2 ModelInfo::ModelInfo (const ModelInfo & *orig*)

5.50.2.3 ModelInfo::∼ModelInfo ( ) `[virtual]`

5.50.3 Member Function Documentation

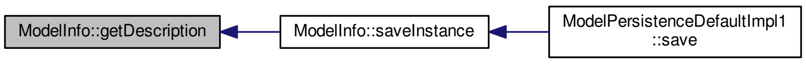
5.50.3.1 std::string ModelInfo::getAnalystName ( ) const

Here is the caller graph for this function:


ModelInfo::getAnalystName
ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

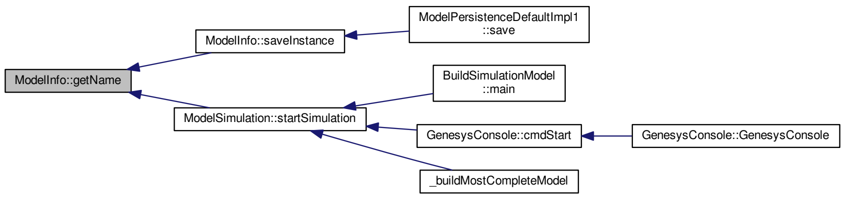
5.50.3.2 std::string ModelInfo::getDescription ( ) const

Here is the caller graph for this function:


ModelInfo::getDescription
ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save

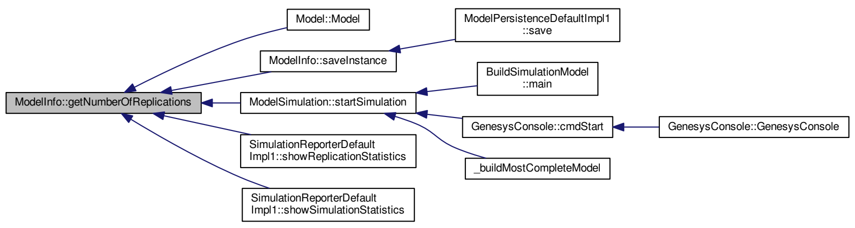
5.50.3.3 std::string ModelInfo::getName ( ) const

Here is the caller graph for this function:


ModelInfo::getName
ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

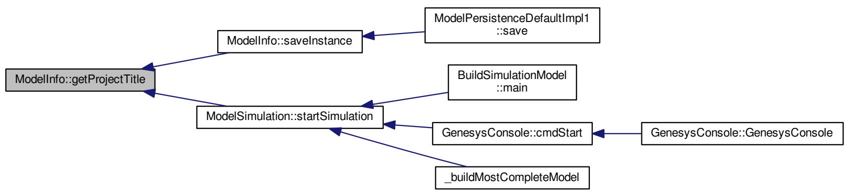
5.50.3.4 unsigned int ModelInfo::getNumberOfReplications ( ) const

Here is the caller graph for this function:


ModelInfo::getNumberOfReplications
Model::Model
ModelInfo::saveInstance
ModelSimulation::startSimulation
SimulationReporterDefaultImpl1::showReplicationStatistics
SimulationReporterDefaultImpl1::showSimulationStatistics
ModelPersistenceDefaultImpl1::save
BuildSimulationModel::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

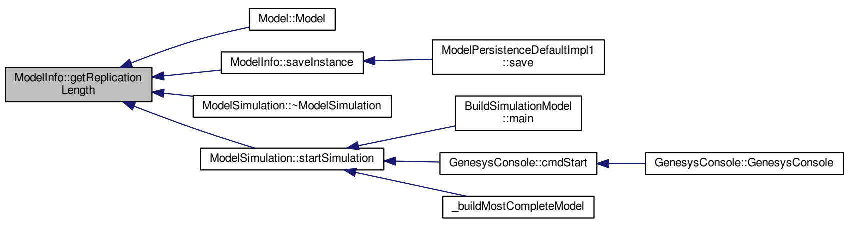
5.50.3.5 std::string ModelInfo::getProjectTitle ( ) const

Here is the caller graph for this function:


ModelInfo::getProjectTitle
ModelInfo::saveInstance
ModelSimulation::startSimulation
ModelPersistenceDefaultImpl1::save
BuildSimulationModel::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

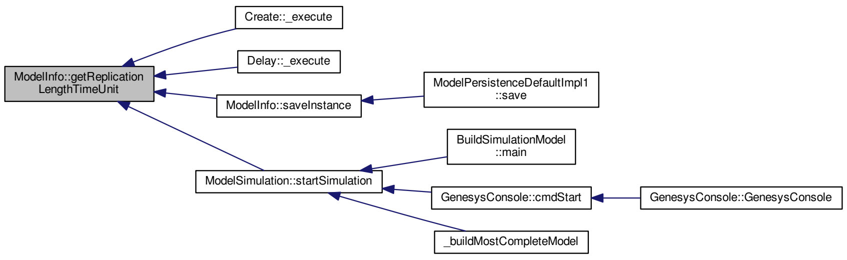
5.50.3.6 double ModelInfo::getReplicationLength ( ) const

Here is the caller graph for this function:


ModelInfo::getReplicationLength
Model::Model
ModelInfo::saveInstance
ModelSimulation::~ModelSimulation
ModelSimulation::startSimulation
ModelPersistenceDefaultImpl1::save
BuildSimulationModel::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

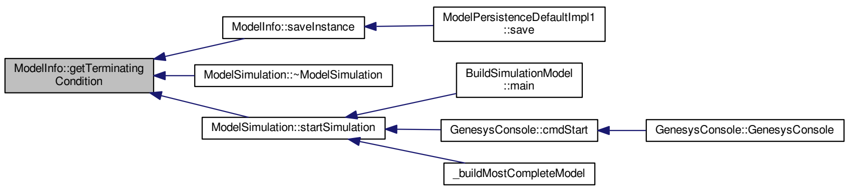
5.50.3.7 Util::TimeUnit ModelInfo::getReplicationLengthTimeUnit ( ) const

Here is the caller graph for this function:


Create::_execute
Delay::_execute
ModelInfo::getReplication LengthTimeUnit
ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

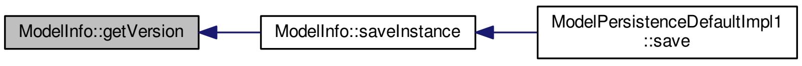
5.50.3.8 std::string ModelInfo::getTerminatingCondition ( ) const

Here is the caller graph for this function:


ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save
ModelInfo::getTerminating Condition
ModelSimulation::~ModelSimulation
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

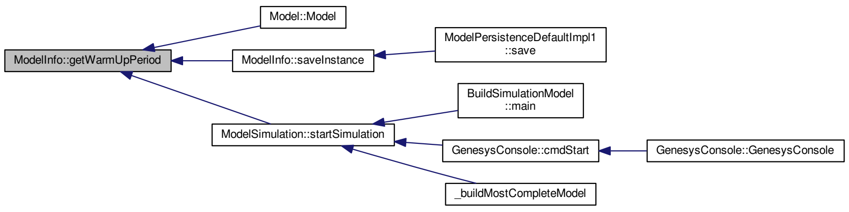
5.50.3.9 std::string ModelInfo::getVersion ( ) const

Here is the caller graph for this function:



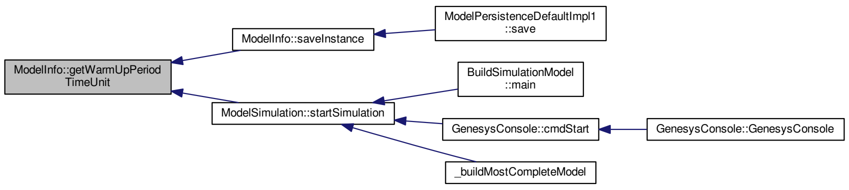
5.50.3.10 double ModelInfo::getWarmUpPeriod ( ) const

Here is the caller graph for this function:


ModelInfo::getWarmUpPeriod
Model::Model
ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

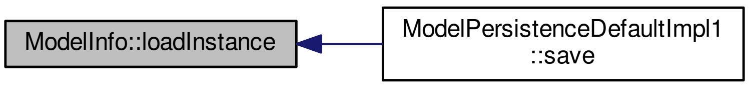
5.50.3.11 Util::TimeUnit ModelInfo::getWarmUpPeriodTimeUnit ( ) const

Here is the caller graph for this function:


ModelInfo::getWarmUpPeriod TimeUnit
ModelInfo::saveInstance
ModelPersistenceDefaultImpl1 ::save
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

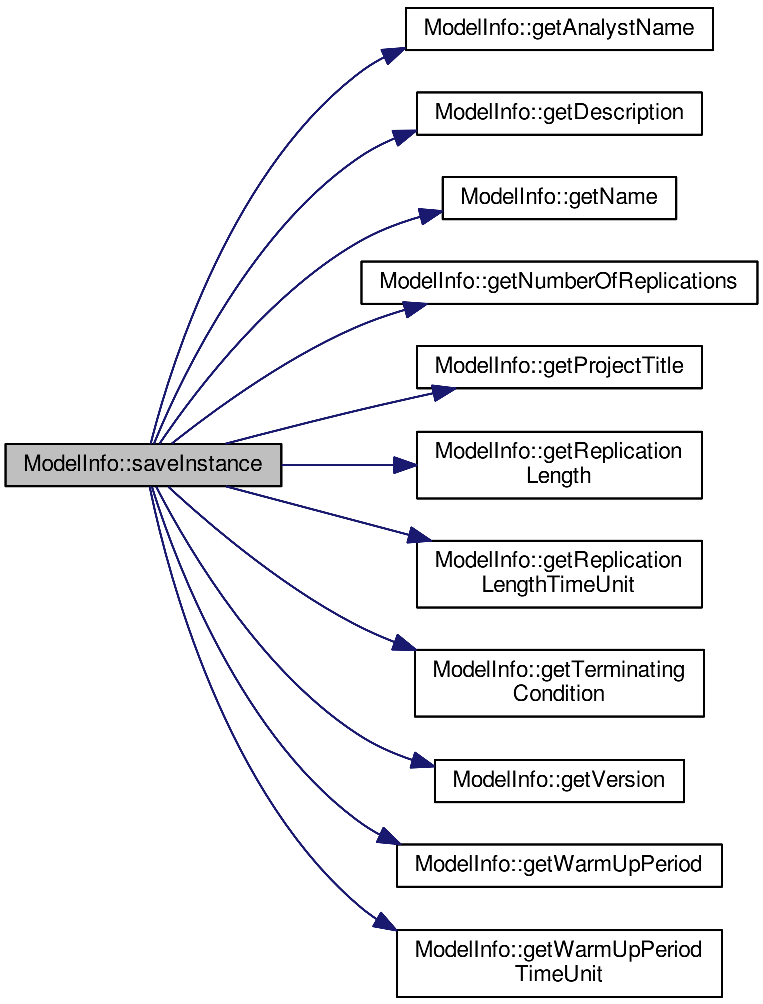
5.50.3.12 void ModelInfo::loadInstance (std::map< std::string, std::string > ∗ *fields*)

Here is the caller graph for this function:

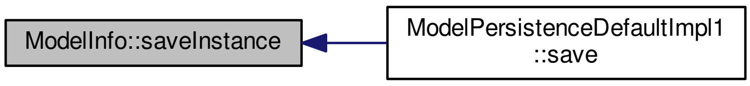


5.50.3.13 std::map< std::string, std::string > ∗ ModelInfo::saveInstance ( )

Here is the call graph for this function:

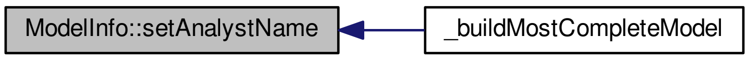

ModelInfo::saveInstance
ModelInfo::getAnalystName
ModelInfo::getDescription
ModelInfo::getName
ModelInfo::getNumberOfReplications
ModelInfo::getProjectTitle
ModelInfo::getReplication Length
ModelInfo::getReplication LengthTimeUnit
ModelInfo::getTerminating Condition
ModelInfo::getVersion
ModelInfo::getWarmUpPeriod
ModelInfo::getWarmUpPeriod TimeUnit

Here is the caller graph for this function:



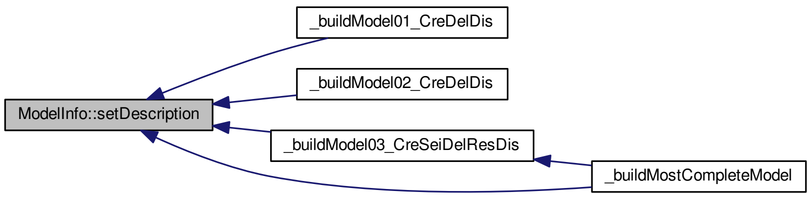
5.50.3.14 void ModelInfo::setAnalystName (std::string *_analystName*)

Here is the caller graph for this function:



5.50.3.15 void ModelInfo::setDescription (std::string *_description*)

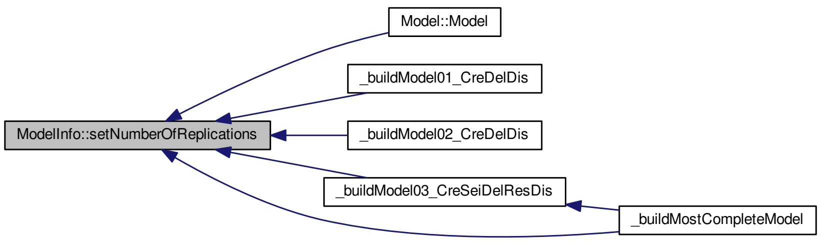
Here is the caller graph for this function:



5.50.3.16 void ModelInfo::setName (std::string *_name*)

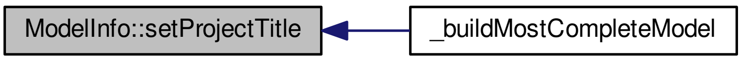
5.50.3.17 void ModelInfo::setNumberOfReplications (unsigned int *_numberOfReplications*)

Here is the caller graph for this function:



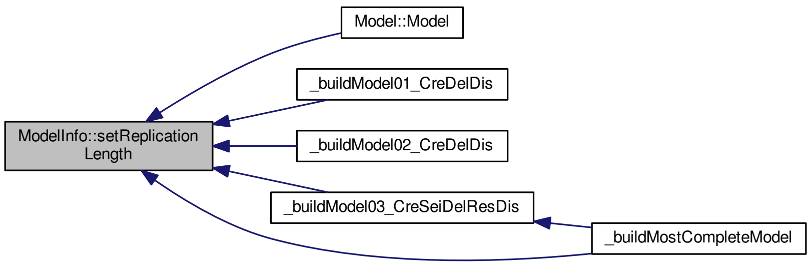
5.50.3.18 void ModelInfo::setProjectTitle (std::string *_projectTitle*)

Here is the caller graph for this function:



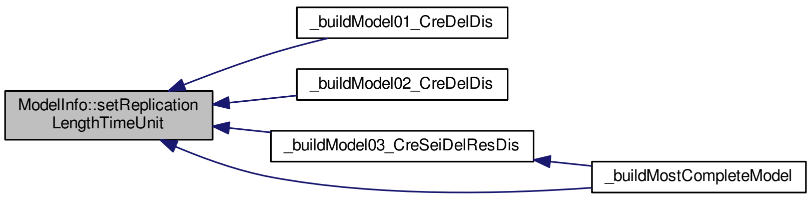
5.50.3.19 void ModelInfo::setReplicationLength (double *_replicationLength*)

Here is the caller graph for this function:



5.50.3.20 void ModelInfo::setReplicationLengthTimeUnit (Util::TimeUnit *_replicationLengthTimeUnit*)

Here is the caller graph for this function:

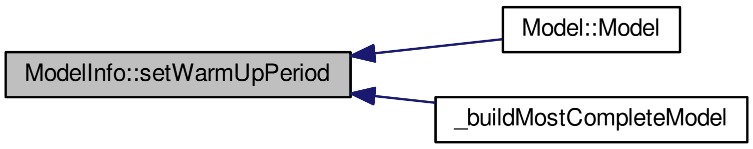

_buildModel01_CreDelDis
_buildModel02_CreDelDis
ModelInfo::setReplication LengthTimeUnit
_buildModel03_CreSeiDelResDis
_buildMostCompleteModel

5.50.3.21 void ModelInfo::setTerminatingCondition (std::string *_terminatingCondition*)

5.50.3.22 void ModelInfo::setVersion (std::string *_version*)

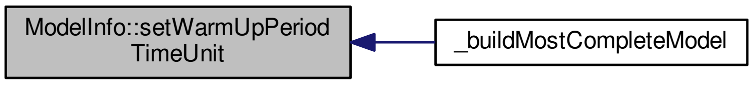
5.50.3.23 void ModelInfo::setWarmUpPeriod (double *_warmUpPeriod*)

Here is the caller graph for this function:



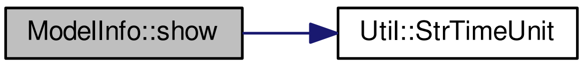
5.50.3.24 void ModelInfo::setWarmUpPeriodTimeUnit (Util::TimeUnit *_warmUpPeriodTimeUnit*)

Here is the caller graph for this function:



5.50.3.25 std::string ModelInfo::show ( )

Here is the call graph for this function:



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

- ModelInfo.h
- ModelInfo.cpp

5.51 ModelManager Class Reference

```
#include <ModelManager.h>
```

Public Member Functions

- ModelManager (Simulator ∗simulator)
- ModelManager (const ModelManager &orig)
- virtual ∼ModelManager ()
- void insert (Model ∗model)
- void remove (Model ∗model)
- void setCurrent (Model ∗model)
- bool saveModel (std::string filename)
- bool loadModel (std::string filename)
- Model ∗ front ()
- Model ∗ current ()
- Model ∗ next ()
- Model ∗ end ()

5.51.1 Constructor & Destructor Documentation

5.51.1.1 ModelManager::ModelManager (**Simulator** ∗ *simulator*)

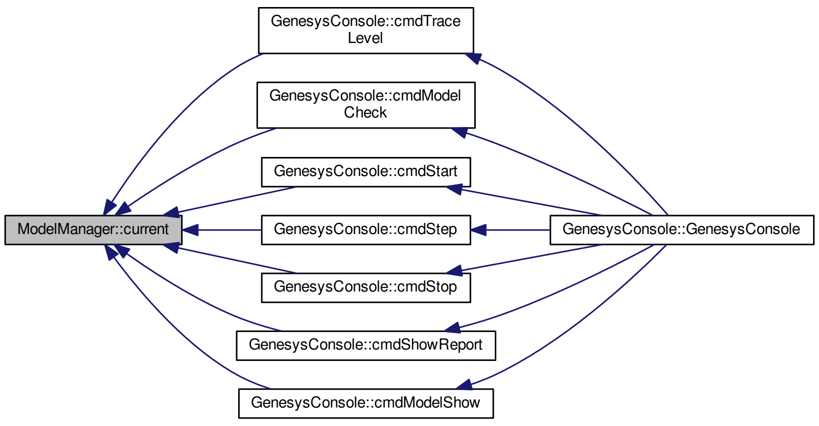
5.51.1.2 ModelManager::ModelManager (const **ModelManager** & *orig*)

5.51.1.3 ModelManager::∼ModelManager ( ) `[virtual]`

5.51.2 Member Function Documentation

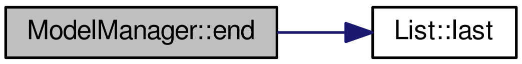
5.51.2.1 **Model** ∗ ModelManager::current ( )

Here is the caller graph for this function:


GenesysConsole::cmdTrace Level
GenesysConsole::cmdModel Check
GenesysConsole::cmdStart
ModelManager::current
GenesysConsole::cmdStep
GenesysConsole::GenesysConsole
GenesysConsole::cmdStop
GenesysConsole::cmdShowReport
GenesysConsole::cmdModelShow

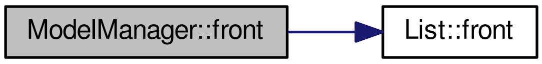
5.51.2.2 Model ∗ ModelManager::end ( )

Here is the call graph for this function:



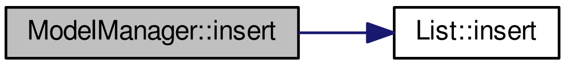
5.51.2.3 Model ∗ ModelManager::front ( )

Here is the call graph for this function:

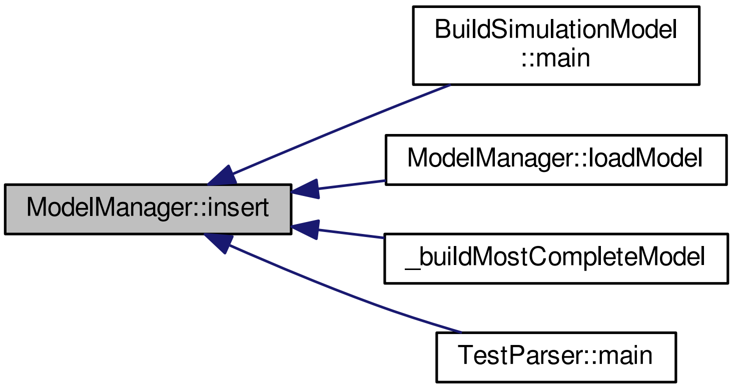


5.51.2.4 void ModelManager::insert (Model ∗ *model*)

Here is the call graph for this function:

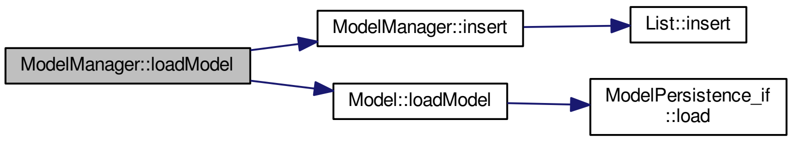


Here is the caller graph for this function:



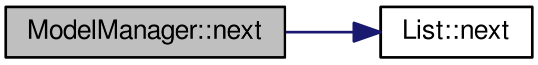
5.51.2.5 bool ModelManager::loadModel (std::string *filename*)

Here is the call graph for this function:



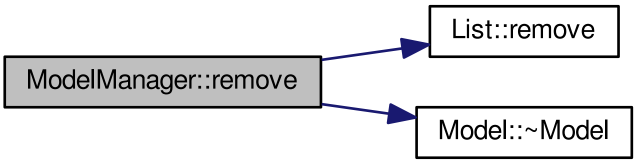
5.51.2.6 Model ∗ ModelManager::next ( )

Here is the call graph for this function:



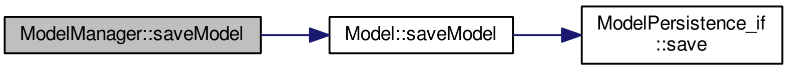
5.51.2.7 void ModelManager::remove (Model ∗ *model*)

Here is the call graph for this function:



5.51.2.8 bool ModelManager::saveModel (std::string *filename*)

Here is the call graph for this function:



5.51.2.9 void ModelManager::setCurrent (Model ∗ *model*)

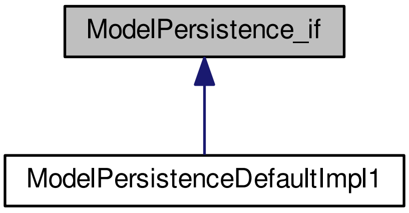
The documentation for this class was generated from the following files:

- ModelManager.h
- ModelManager.cpp

5.52 ModelPersistence_if Class Reference

```
#include <ModelPersistence_if.h>
```

Inheritance diagram for ModelPersistence_if:



Public Member Functions

- virtual bool save (std::string filename)=0
- virtual bool load (std::string filename)=0
- virtual bool isSaved ()=0

5.52.1 Detailed Description

First and inadequate interface for model persistence. It should use the best pattern for the DAO approach

5.52.2 Member Function Documentation

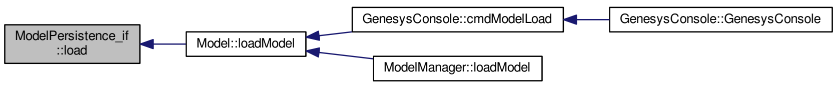
5.52.2.1 virtual bool ModelPersistence_if::isSaved ( ) [pure virtual]

Implemented in ModelPersistenceDefaultImpl1.

5.52.2.2 virtual bool ModelPersistence_if::load (std::string *filename*) [pure virtual]

Implemented in ModelPersistenceDefaultImpl1.

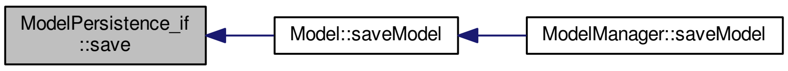
Here is the caller graph for this function:


ModelPersistence_if ::load
Model::loadModel
GenesysConsole::cmdModelLoad
GenesysConsole::GenesysConsole
ModelManager::loadModel

5.52.2.3 virtual bool ModelPersistence_if::save (std::string *filename*) [pure virtual]

Implemented in ModelPersistenceDefaultImpl1.

Here is the caller graph for this function:



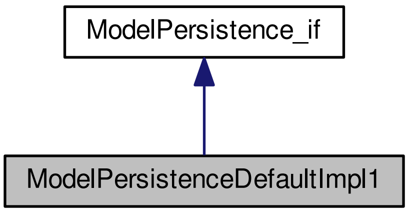
The documentation for this class was generated from the following file:

- ModelPersistence_if.h

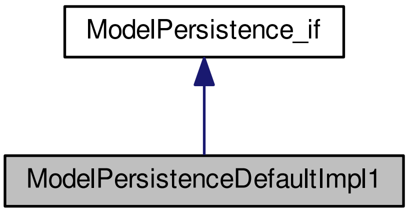
5.53 ModelPersistenceDefaultImpl1 Class Reference

```
#include <ModelPersistenceDefaultImpl1.h>
```

Inheritance diagram for ModelPersistenceDefaultImpl1:



Collaboration diagram for ModelPersistenceDefaultImpl1:



Public Member Functions

- ModelPersistenceDefaultImpl1 (Model ∗model)
- ModelPersistenceDefaultImpl1 (const ModelPersistenceDefaultImpl1 &orig)
- virtual ∼ModelPersistenceDefaultImpl1 ()
- virtual bool save (std::string filename)
- virtual bool load (std::string filename)
- virtual bool isSaved ()

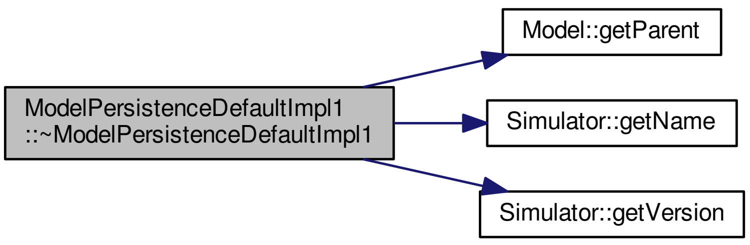
5.53.1 Constructor & Destructor Documentation

5.53.1.1 ModelPersistenceDefaultImpl1::ModelPersistenceDefaultImpl1 (**Model** ∗ *model*)

5.53.1.2 ModelPersistenceDefaultImpl1::ModelPersistenceDefaultImpl1 (const **ModelPersistenceDefaultImpl1** & *orig*)

5.53.1.3 ModelPersistenceDefaultImpl1::∼ModelPersistenceDefaultImpl1 ( ) [virtual]

Here is the call graph for this function:


ModelPersistenceDefaultImpl1 ::~ModelPersistenceDefaultImpl1
Model::getParent
Simulator::getName
Simulator::getVersion

5.53.2 Member Function Documentation

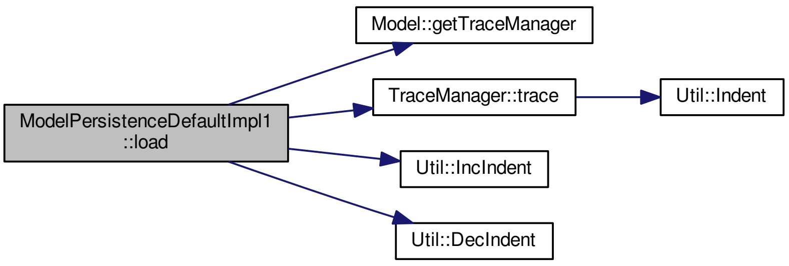
5.53.2.1 bool ModelPersistenceDefaultImpl1::isSaved ( ) [virtual]

Implements ModelPersistence_if.

5.53.2.2 bool ModelPersistenceDefaultImpl1::load (std::string *filename*) [virtual]

Implements ModelPersistence_if.

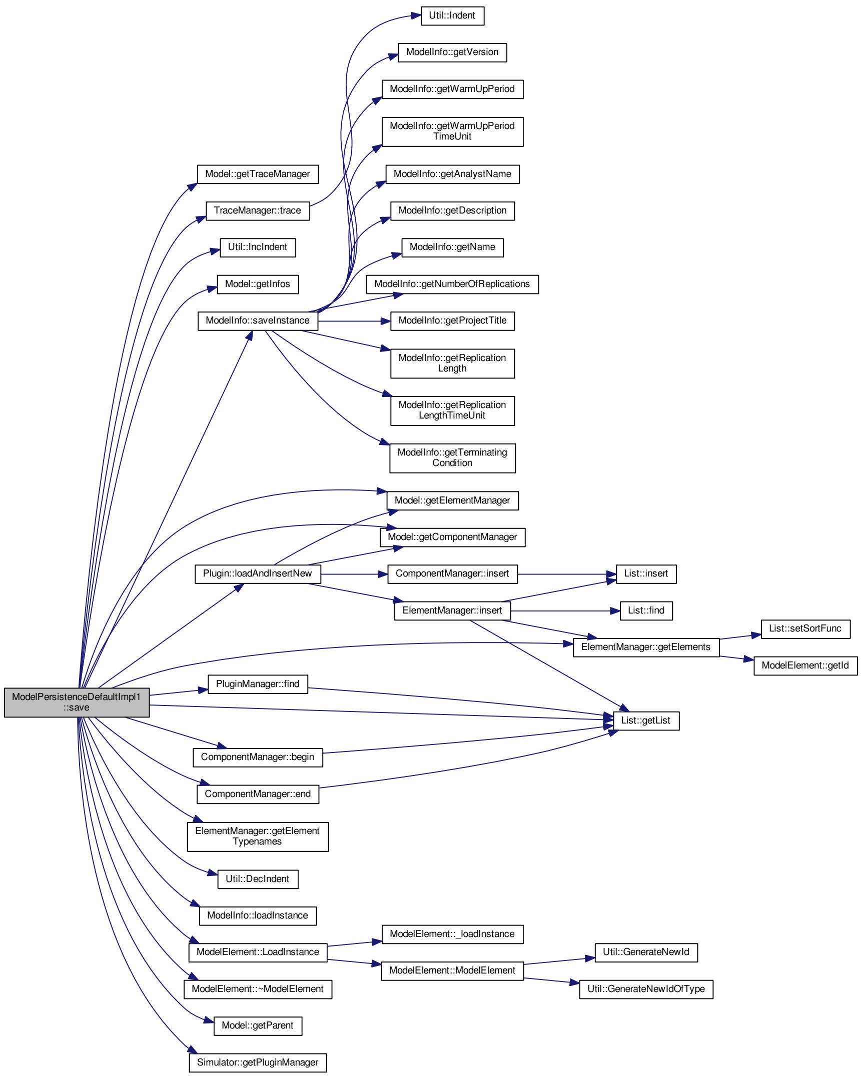
Here is the call graph for this function:



5.53.2.3 bool ModelPersistenceDefaultImpl1::save (std::string *filename*) [virtual]

Implements ModelPersistence_if.

Here is the call graph for this function:


Util::Indent
ModelInfo::getVersion
ModelInfo::getWarmUpPeriod
ModelInfo::getWarmUpPeriod TimeUnit
Model::getTraceManager
ModelInfo::getAnalystName
TraceManager::trace
ModelInfo::getDescription
Util::IncIndent
ModelInfo::getName
Model::getInfos
ModelInfo::getNumberOfReplications
ModelInfo::saveInstance
ModelInfo::getProjectTitle
ModelInfo::getReplication Length
ModelInfo::getReplication LengthTimeUnit
ModelInfo::getTerminating Condition
Model::getElementManager
Model::getComponentManager
Plugin::loadAndInsertNew
ComponentManager::insert
List::insert
ElementManager::insert
List::find
List::setSortFunc
ElementManager::getElements
ModelElement::getId
ModelPersistenceDefaultImpl1 ::save
PluginManager::find
List::getList
ComponentManager::begin
ComponentManager::end
ElementManager::getElement Typenames
Util::DecIndent
ModelInfo::loadInstance
ModelElement::_loadInstance
ModelElement::LoadInstance
Util::GenerateNewId
ModelElement::ModelElement
ModelElement::~ModelElement
Util::GenerateNewIdOfType
Model::getParent
Simulator::getPluginManager

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

- ModelPersistenceDefaultImpl1.h
- ModelPersistenceDefaultImpl1.cpp

5.54 ModelSimulation Class Reference

```
#include <ModelSimulation.h>
```

Public Member Functions

- ModelSimulation (Model ∗model)
- ModelSimulation (const ModelSimulation &orig)

- virtual ∼ModelSimulation ()
- void startSimulation ()

  Starts a sequential execution of a simulation, ie, a set of replications of this model.
- void pauseSimulation ()
- void stepSimulation ()

  Executes the processing of a single event, the next one in the future events list.
- void stopSimulation ()
- void restartSimulation ()
- void setPauseOnEvent (bool _pauseOnEvent)
- bool isPauseOnEvent () const
- void setStepByStep (bool _stepByStep)
- bool isStepByStep () const
- void setInitializeStatistics (bool _initializeStatistics)
- bool isInitializeStatistics () const
- void setInitializeSystem (bool _initializeSystem)
- bool isInitializeSystem () const
- void setPauseOnReplication (bool _pauseBetweenReplications)
- bool isPauseOnReplication () const
- double getSimulatedTime () const
- bool isRunning () const
- unsigned int getCurrentReplicationNumber () const
- ModelComponent ∗ getCurrentComponent () const
- Entity ∗ getCurrentEntity () const
- SimulationReporter_if ∗ getSimulationReporter () const

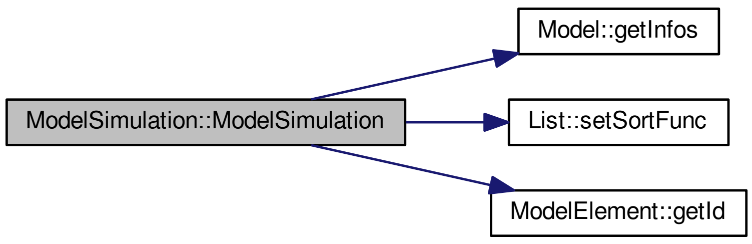
5.54.1 Detailed Description

The ModelSimulation controls the simulation of a model, alowing to start, pause, resume e stop a simulation, composed by a set of replications.

5.54.2 Constructor & Destructor Documentation

5.54.2.1 ModelSimulation::ModelSimulation (Model ∗ *model*)

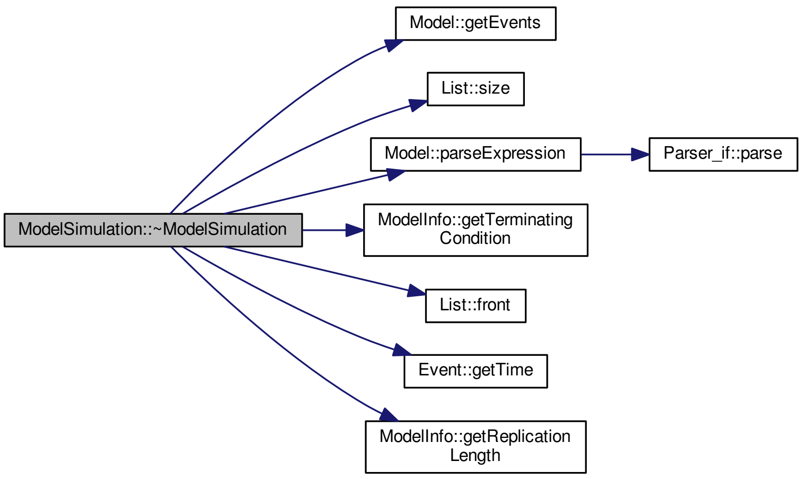
Here is the call graph for this function:



5.54.2.2 ModelSimulation::ModelSimulation (const ModelSimulation & *orig*)

5.54.2.3 ModelSimulation::∼ModelSimulation ( ) [virtual]

Here is the call graph for this function:


Model::getEvents
List::size
Model::parseExpression
Parser_if::parse
ModelSimulation::~ModelSimulation
ModelInfo::getTerminating Condition
List::front
Event::getTime
ModelInfo::getReplication Length

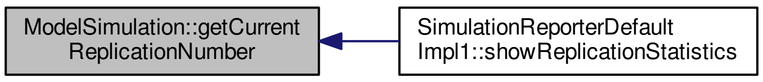
5.54.3 Member Function Documentation

5.54.3.1 ModelComponent ∗ ModelSimulation::getCurrentComponent ( ) const

5.54.3.2 Entity ∗ ModelSimulation::getCurrentEntity ( ) const

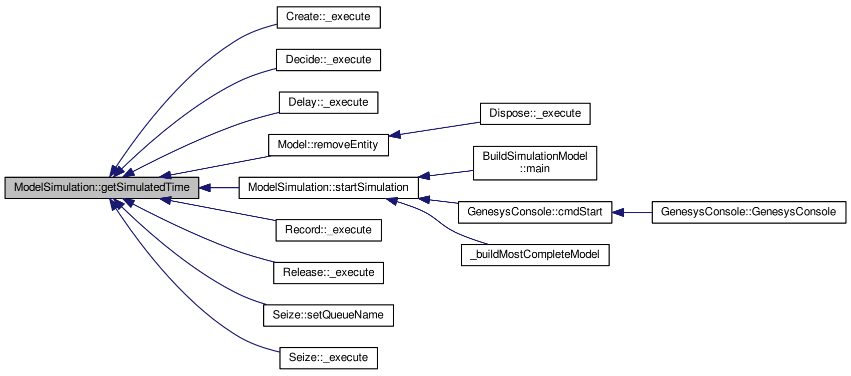
5.54.3.3 unsigned int ModelSimulation::getCurrentReplicationNumber ( ) const

Here is the caller graph for this function:


ModelSimulation::getCurrent ReplicationNumber
SimulationReporterDefault Impl1::showReplicationStatistics

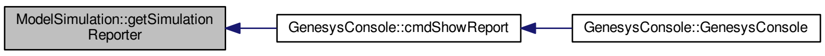
5.54.3.4 double ModelSimulation::getSimulatedTime ( ) const

Here is the caller graph for this function:



5.54.3.5 SimulationReporter_if ∗ ModelSimulation::getSimulationReporter ( ) const

Here is the caller graph for this function:


ModelSimulation::getSimulation Reporter
GenesysConsole::cmdShowReport
GenesysConsole::GenesysConsole

5.54.3.6 bool ModelSimulation::isInitializeStatistics ( ) const

5.54.3.7 bool ModelSimulation::isInitializeSystem ( ) const

5.54.3.8 bool ModelSimulation::isPauseOnEvent ( ) const

5.54.3.9 bool ModelSimulation::isPauseOnReplication ( ) const

5.54.3.10 bool ModelSimulation::isRunning ( ) const

The current time in the model being simulated, i.e., the instant when the current event was triggered

5.54.3.11 bool ModelSimulation::isStepByStep ( ) const

5.54.3.12 void ModelSimulation::pauseSimulation ( )

5.54.3.13 void ModelSimulation::restartSimulation ( )

5.54.3.14 void ModelSimulation::setInitializeStatistics (bool *_initializeStatistics*)

5.54.3.15 void ModelSimulation::setInitializeSystem (bool *_initializeSystem*)

5.54.3.16 void ModelSimulation::setPauseOnEvent (bool *_pauseOnEvent*)

5.54.3.17 void ModelSimulation::setPauseOnReplication (bool *_pauseBetweenReplications*)

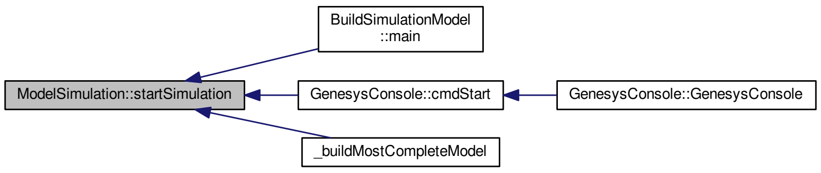
5.54.3.18 void ModelSimulation::setStepByStep (bool *_stepByStep*)

5.54.3.19 void ModelSimulation::startSimulation ( )

Starts a sequential execution of a simulation, ie, a set of replications of this model.

Checks the model and if ok then initialize the simulation, execute repeatedly each replication and then show simulation statistics

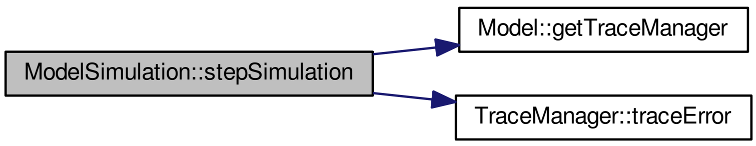
Here is the caller graph for this function:


BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

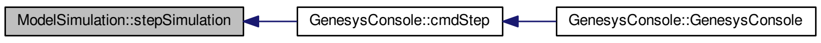
5.54.3.20 void ModelSimulation::stepSimulation ( )

Executes the processing of a single event, the next one in the future events list.

Here is the call graph for this function:

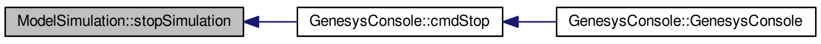


Here is the caller graph for this function:


ModelSimulation::stepSimulation
GenesysConsole::cmdStep
GenesysConsole::GenesysConsole

5.54.3.21 void ModelSimulation::stopSimulation ( )

Here is the caller graph for this function:


ModelSimulation::stopSimulation
GenesysConsole::cmdStop
GenesysConsole::GenesysConsole

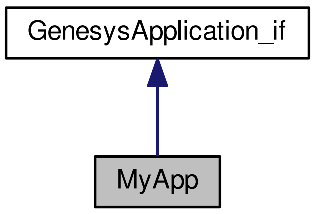
The documentation for this class was generated from the following files:

- ModelSimulation.h
- ModelSimulation.cpp

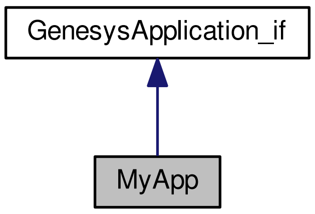
5.55 MyApp Class Reference

```
#include <MyApp.h>
```

Inheritance diagram for MyApp:



Collaboration diagram for MyApp:



Public Member Functions

- MyApp ()
- MyApp (const MyApp &orig)
- virtual ∼MyApp ()
- virtual int main (int argc, char ∗∗argv)

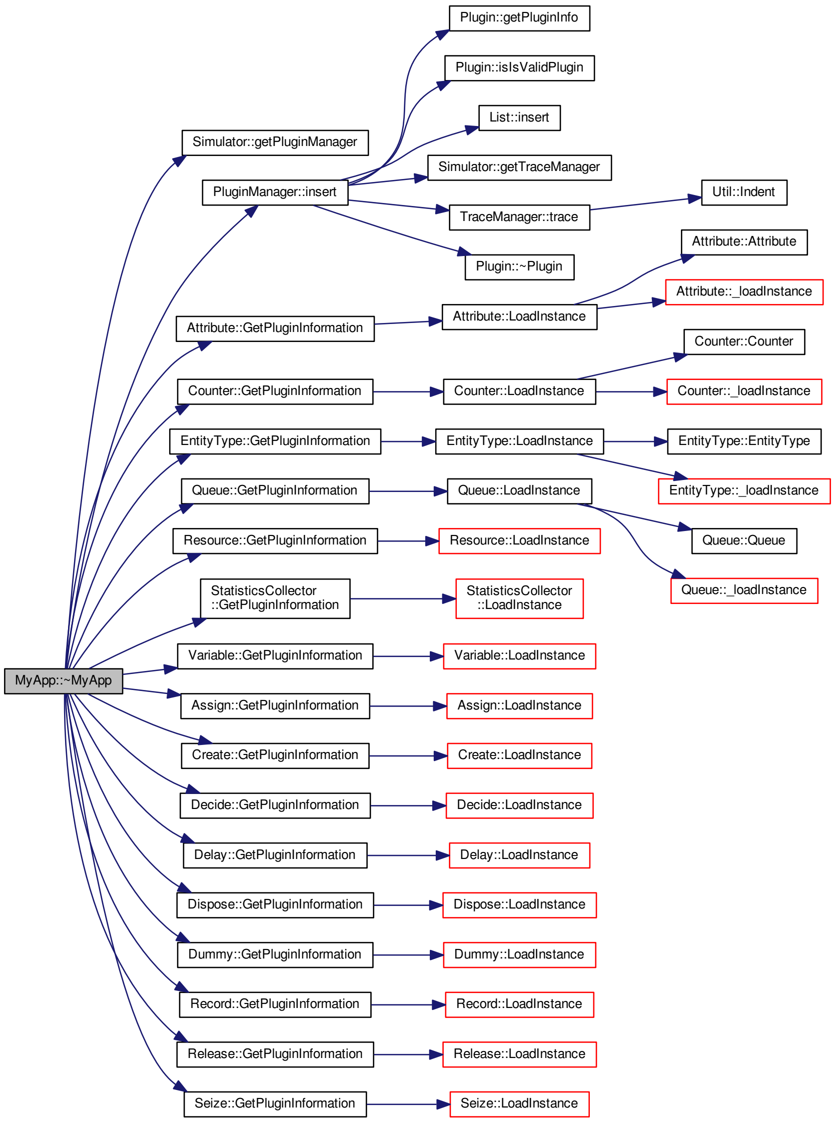
5.55.1 Constructor & Destructor Documentation

5.55.1.1 MyApp::MyApp ( )

5.55.1.2 MyApp::MyApp (const **MyApp** & *orig*)

5.55.1.3 MyApp::∼MyApp ( ) [virtual]

Here is the call graph for this function:

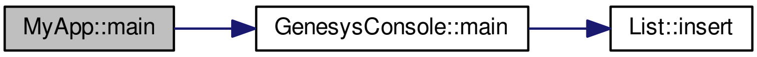

MyApp::~MyApp
Simulator::getPluginManager
PluginManager::insert
Plugin::getPluginInfo
Plugin::isIsValidPlugin
List::insert
Simulator::getTraceManager
TraceManager::trace
Util::Indent
Plugin::~Plugin
Attribute::GetPluginInformation
Attribute::LoadInstance
Attribute::Attribute
Attribute::_loadInstance
Counter::GetPluginInformation
Counter::LoadInstance
Counter::Counter
Counter::_loadInstance
EntityType::GetPluginInformation
EntityType::LoadInstance
EntityType::EntityType
EntityType::_loadInstance
Queue::GetPluginInformation
Queue::LoadInstance
Queue::Queue
Queue::_loadInstance
Resource::GetPluginInformation
Resource::LoadInstance
StatisticsCollector ::GetPluginInformation
StatisticsCollector ::LoadInstance
Variable::GetPluginInformation
Variable::LoadInstance
Assign::GetPluginInformation
Assign::LoadInstance
Create::GetPluginInformation
Create::LoadInstance
Decide::GetPluginInformation
Decide::LoadInstance
Delay::GetPluginInformation
Delay::LoadInstance
Dispose::GetPluginInformation
Dispose::LoadInstance
Dummy::GetPluginInformation
Dummy::LoadInstance
Record::GetPluginInformation
Record::LoadInstance
Release::GetPluginInformation
Release::LoadInstance
Seize::GetPluginInformation
Seize::LoadInstance

5.55.2 Member Function Documentation

5.55.2.1 int MyApp::main (int *argc*, char ∗∗ *argv*) [virtual]

Implements GenesysApplication_if.

Here is the call graph for this function:



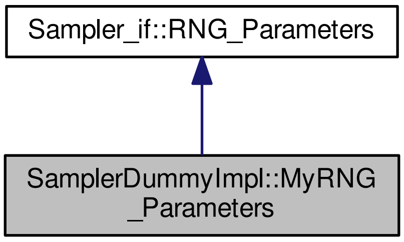
The documentation for this class was generated from the following files:

- MyApp.h
- MyApp.cpp

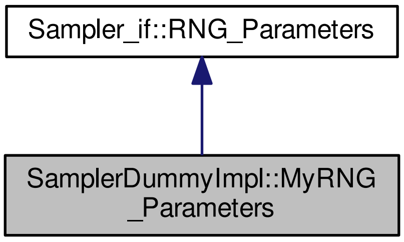
5.56 SamplerDummyImpl::MyRNG_Parameters Struct Reference

```
#include <SamplerDummyImpl.h>
```

Inheritance diagram for SamplerDummyImpl::MyRNG_Parameters:



Collaboration diagram for SamplerDummyImpl::MyRNG_Parameters:



Public Member Functions

- ∼MyRNG_Parameters ()=default

Public Attributes

- unsigned int seed
- unsigned int module
- unsigned int multiplier

5.56.1 Constructor & Destructor Documentation

5.56.1.1 SamplerDummyImpl::MyRNG_Parameters::∼MyRNG_Parameters ( ) [default]

5.56.2 Member Data Documentation

5.56.2.1 unsigned int SamplerDummyImpl::MyRNG_Parameters::module

5.56.2.2 unsigned int SamplerDummyImpl::MyRNG_Parameters::multiplier

5.56.2.3 unsigned int SamplerDummyImpl::MyRNG_Parameters::seed

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

- SamplerDummyImpl.h

5.57 OnEventManager Class Reference

```
#include <OnEventManager.h>
```

Public Member Functions

- OnEventManager ()
- OnEventManager (const OnEventManager &orig)
- virtual ∼OnEventManager ()
- void addOnReplicationStartHandler (simulationEventHandler EventHandler)
- void addOnReplicationStepHandler (simulationEventHandler EventHandler)
- void addOnReplicationEndHandler (simulationEventHandler EventHandler)
- void addOnProcessEventHandler (simulationEventHandler EventHandler)
- void addOnSimulationStartHandler (simulationEventHandler EventHandler)
- void addOnSimulationEndHandler (simulationEventHandler EventHandler)
- void addOnEntityRemoveHandler (simulationEventHandler EventHandler)
- void NotifyReplicationStartHandlers (SimulationEvent ∗se)
- void NotifyReplicationStepHandlers (SimulationEvent ∗se)
- void NotifyReplicationEndHandlers (SimulationEvent ∗se)
- void NotifyProcessEventHandlers (SimulationEvent ∗se)
- void NotifySimulationStartHandlers (SimulationEvent ∗se)
- void NotifySimulationEndHandlers (SimulationEvent ∗se)

5.57.1 Detailed Description

OnEventManager allows external methods to hook interval simulation events as listeners (or observers) of pecific events. All methods added as listeners of an event will be invovked when that event is triggered.

5.57.2 Constructor & Destructor Documentation

5.57.2.1 OnEventManager::OnEventManager ( )

5.57.2.2 OnEventManager::OnEventManager (const OnEventManager & *orig*)

5.57.2.3 OnEventManager::∼OnEventManager ( ) [virtual]

5.57.3 Member Function Documentation

5.57.3.1 void OnEventManager::addOnEntityRemoveHandler (simulationEventHandler *EventHandler*)

5.57.3.2 void OnEventManager::addOnProcessEventHandler (simulationEventHandler *EventHandler*)

5.57.3.3 void OnEventManager::addOnReplicationEndHandler (simulationEventHandler *EventHandler*)

5.57.3.4 void OnEventManager::addOnReplicationStartHandler (simulationEventHandler *EventHandler*)

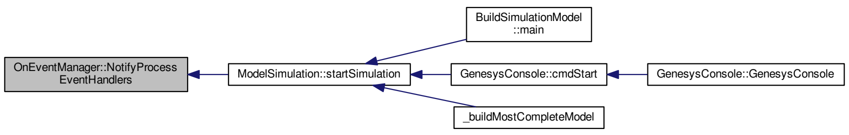
5.57.3.5 void OnEventManager::addOnReplicationStepHandler (simulationEventHandler *EventHandler*)

5.57.3.6 void OnEventManager::addOnSimulationEndHandler (simulationEventHandler *EventHandler*)

5.57.3.7 void OnEventManager::addOnSimulationStartHandler (simulationEventHandler *EventHandler*)

5.57.3.8 void OnEventManager::NotifyProcessEventHandlers (SimulationEvent ∗ *se*)

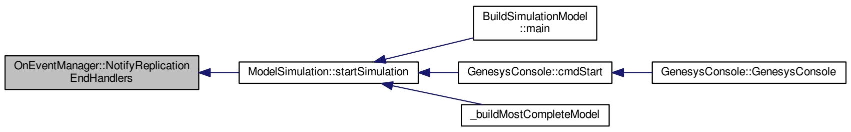
Here is the caller graph for this function:



OnEventManager::NotifyProcess EventHandlers
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

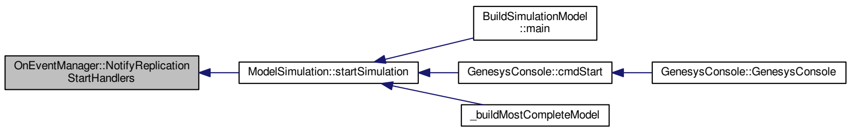
5.57.3.9 void OnEventManager::NotifyReplicationEndHandlers (SimulationEvent * *se*)

Here is the caller graph for this function:


OnEventManager::NotifyReplication EndHandlers
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

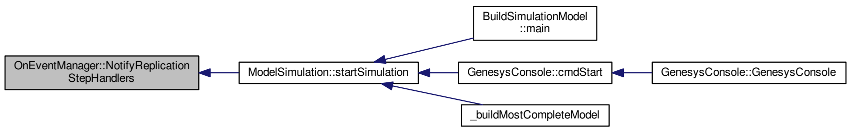
5.57.3.10 void OnEventManager::NotifyReplicationStartHandlers (SimulationEvent * *se*)

Here is the caller graph for this function:


OnEventManager::NotifyReplication StartHandlers
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

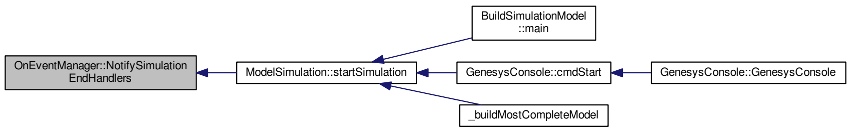
5.57.3.11 void OnEventManager::NotifyReplicationStepHandlers (SimulationEvent * *se*)

Here is the caller graph for this function:


OnEventManager::NotifyReplication StepHandlers
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

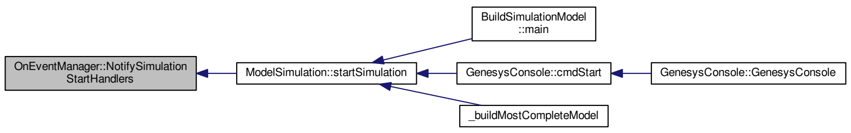
5.57.3.12 void OnEventManager::NotifySimulationEndHandlers (SimulationEvent * *se*)

Here is the caller graph for this function:


OnEventManager::NotifySimulation EndHandlers
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

5.57.3.13 void OnEventManager::NotifySimulationStartHandlers (SimulationEvent ∗ *se*)

Here is the caller graph for this function:


OnEventManager::NotifySimulation StartHandlers
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

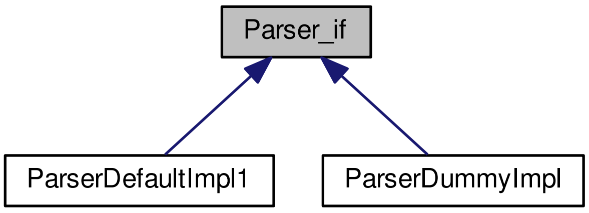
The documentation for this class was generated from the following files:

- OnEventManager.h
- OnEventManager.cpp

5.58 Parser_if Class Reference

```
#include <Parser_if.h>
```

Inheritance diagram for Parser_if:



Public Member Functions

- virtual double parse (const std::string expression)=0
- virtual double parse (const std::string expression, bool ∗success, std::string ∗errorMessage)=0
- virtual std::string ∗ getErrorMessage ()=0

5.58.1 Member Function Documentation

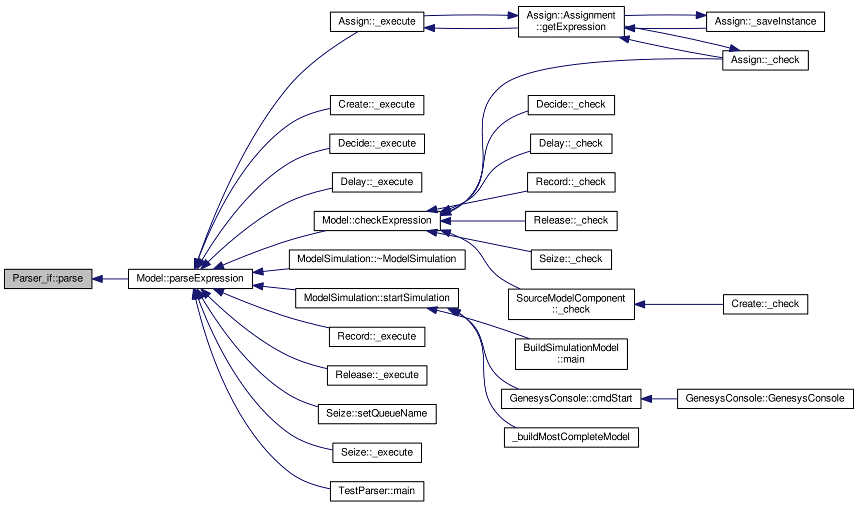
5.58.1.1 virtual std::string∗ Parser_if::getErrorMessage ( ) [pure virtual]

Implemented in ParserDummyImpl, and ParserDefaultImpl1.

5.58.1.2 virtual double Parser_if::parse (const std::string *expression*) [pure virtual]

Implemented in ParserDummyImpl, and ParserDefaultImpl1.

Here is the caller graph for this function:



5.58.1.3 virtual double Parser_if::parse (const std::string *expression,* bool ∗ *success,* std::string ∗ *errorMessage*) [pure virtual]

Implemented in ParserDummyImpl, and ParserDefaultImpl1.

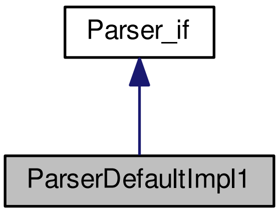
The documentation for this class was generated from the following file:

- Parser_if.h

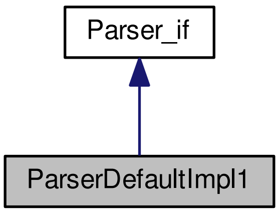
5.59 ParserDefaultImpl1 Class Reference

`#include <ParserDefaultImpl1.h>`

Inheritance diagram for ParserDefaultImpl1:



Collaboration diagram for ParserDefaultImpl1:



Public Member Functions

- ParserDefaultImpl1 (Model ∗model)
- ParserDefaultImpl1 (const ParserDefaultImpl1 &orig)
- virtual ∼ParserDefaultImpl1 ()
- double parse (const std::string expression)
- double parse (const std::string expression, bool ∗success, std::string ∗errorMessage)
- std::string ∗ getErrorMessage ()

5.59.1 Constructor & Destructor Documentation

5.59.1.1 ParserDefaultImpl1::ParserDefaultImpl1 (Model ∗ *model*)

5.59.1.2 ParserDefaultImpl1::ParserDefaultImpl1 (const ParserDefaultImpl1 & *orig*)

5.59.1.3 ParserDefaultImpl1::∼ParserDefaultImpl1 ( ) [virtual]

5.59.2 Member Function Documentation

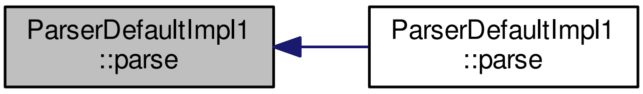
5.59.2.1 std::string ∗ ParserDefaultImpl1::getErrorMessage ( ) [virtual]

Implements Parser_if.

5.59.2.2 double ParserDefaultImpl1::parse (const std::string *expression*) [virtual]

Implements Parser_if.

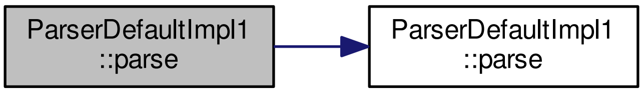
Here is the caller graph for this function:



5.59.2.3 double ParserDefaultImpl1::parse (const std::string *expression*, bool * *success*, std::string * *errorMessage*) [virtual]

Implements Parser_if.

Here is the call graph for this function:



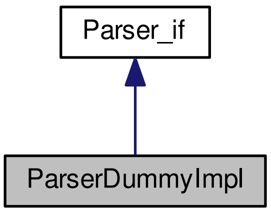
The documentation for this class was generated from the following files:

- ParserDefaultImpl1.h
- ParserDefaultImpl1.cpp

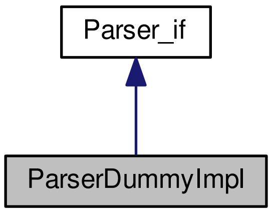
5.60 ParserDummyImpl Class Reference

```
#include <ParserDummyImpl.h>
```

Inheritance diagram for ParserDummyImpl:



Collaboration diagram for ParserDummyImpl:



Public Member Functions

- ParserDummyImpl (Model ∗model)
- ParserDummyImpl (const ParserDummyImpl &orig)
- virtual ∼ParserDummyImpl ()
- double parse (const std::string expression)
- double parse (const std::string expression, bool ∗success, std::string ∗errorMessage)
- std::string ∗ getErrorMessage ()

5.60.1 Constructor & Destructor Documentation

5.60.1.1 ParserDummyImpl::ParserDummyImpl (**Model** ∗ *model*)

5.60.1.2 ParserDummyImpl::ParserDummyImpl (const **ParserDummyImpl** & *orig*)

5.60.1.3 ParserDummyImpl::∼ParserDummyImpl ( ) `[virtual]`

5.60.2 Member Function Documentation

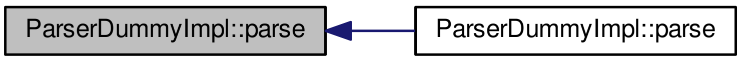
5.60.2.1 std::string ∗ ParserDummyImpl::getErrorMessage ( ) `[virtual]`

Implements Parser_if.

5.60.2.2 double ParserDummyImpl::parse (const std::string *expression*) `[virtual]`

Implements Parser_if.

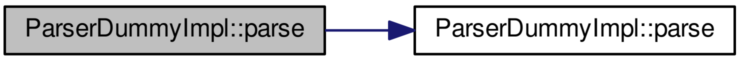
Here is the caller graph for this function:



5.60.2.3 double ParserDummyImpl::parse (const std::string *expression*, bool ∗ *success*, std::string ∗ *errorMessage*) [virtual]

Implements Parser_if.

Here is the call graph for this function:



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

- ParserDummyImpl.h
- ParserDummyImpl.cpp

5.61 Plugin Class Reference

```
#include <Plugin.h>
```

Public Member Functions

- Plugin (StaticGetPluginInformation getInformation)
- Plugin (const Plugin &orig)
- virtual ∼Plugin ()
- bool isIsValidPlugin () const
- PluginInformation ∗ getPluginInfo () const
- ModelElement ∗ loadNew (Model ∗model, std::map< std::string, std::string > ∗fields)
- bool loadAndInsertNew (Model ∗model, std::map< std::string, std::string > ∗fields)

5.61.1 Detailed Description

A Plugin represents a dynamically linked component class (ModelComponent) or element class (ModelElement); It gives access to a ModelComponent so it can be used by the model. Classes like Create, Delay, and Dispose are examples of PlugIns. It corresponds directly to the "Expansible" part (the capitalized 'E') of the GenESyS acronymous PlugIns are NOT implemented yet

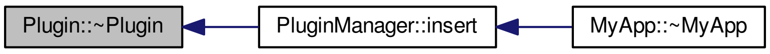
5.61.2 Constructor & Destructor Documentation

5.61.2.1 Plugin::Plugin (**StaticGetPluginInformation** *getInformation*)

5.61.2.2 Plugin::Plugin (const **Plugin** & *orig*)

5.61.2.3 Plugin::∼Plugin ( ) `[virtual]`

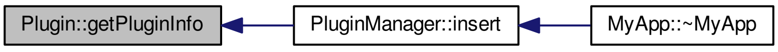
Here is the caller graph for this function:



5.61.3 Member Function Documentation

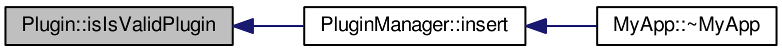
5.61.3.1 **PluginInformation** ∗ Plugin::getPluginInfo ( ) const

Here is the caller graph for this function:



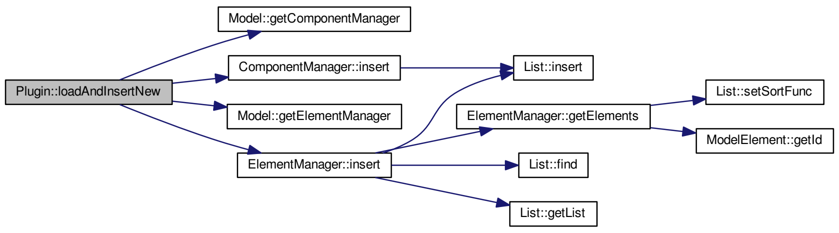
5.61.3.2 bool Plugin::isIsValidPlugin ( ) const

Here is the caller graph for this function:

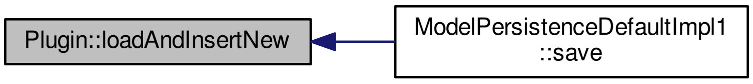


5.61.3.3 bool Plugin::loadAndInsertNew (**Model** ∗ *model,* std::map< std::string, std::string > ∗ *fields*)

Here is the call graph for this function:

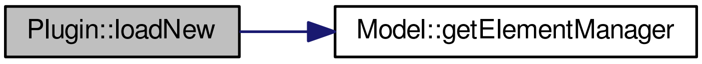


Here is the caller graph for this function:



5.61.3.4 **ModelElement** ∗ Plugin::loadNew (**Model** ∗ *model,* std::map< std::string, std::string > ∗ *fields*)

Here is the call graph for this function:



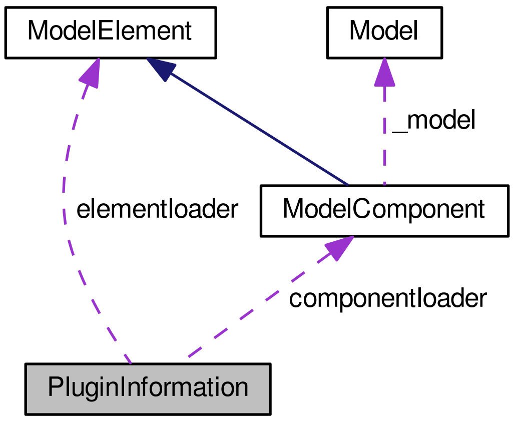
The documentation for this class was generated from the following files:

- Plugin.h
- Plugin.cpp

5.62 PluginInformation Class Reference

```
#include <Plugin.h>
```

Collaboration diagram for PluginInformation:



Public Member Functions

- PluginInformation (std::string pluginTypename, StaticLoaderComponentInstance componentloader)
- PluginInformation (std::string pluginTypename, StaticLoaderElementInstance elementloader)

Public Attributes

- std::string pluginTypename
- std::string author = ""
- std::string date = ""
- std::string observation = ""
- bool isSource
- bool isSink
- bool isComponent
- bool generateReport = false
- StaticLoaderComponentInstance componentloader
- StaticLoaderElementInstance elementloader
- std::list< PluginInformation > dependencies

5.62.1 Constructor & Destructor Documentation

5.62.1.1 PluginInformation::PluginInformation (std::string *pluginTypename*, StaticLoaderComponentInstance *componentloader*) [inline]

5.62.1.2 PluginInformation::PluginInformation (std::string *pluginTypename*, StaticLoaderElementInstance *elementloader*) [inline]

5.62.2 Member Data Documentation

5.62.2.1 std::string PluginInformation::author = ""

5.62.2.2 StaticLoaderComponentInstance PluginInformation::componentloader

5.62.2.3 std::string PluginInformation::date = ""

5.62.2.4 std::list<PluginInformation> PluginInformation::dependencies

5.62.2.5 StaticLoaderElementInstance PluginInformation::elementloader

5.62.2.6 bool PluginInformation::generateReport = false

5.62.2.7 bool PluginInformation::isComponent

5.62.2.8 bool PluginInformation::isSink

5.62.2.9 bool PluginInformation::isSource

5.62.2.10 std::string PluginInformation::observation = ""

5.62.2.11 std::string PluginInformation::pluginTypename

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

- Plugin.h

5.63 PluginManager Class Reference

`#include <PluginManager.h>`

Public Member Functions

- PluginManager (Simulator ∗simulator)
- PluginManager (const PluginManager &orig)
- virtual ∼PluginManager ()
- bool insert (Plugin ∗plugin)
- void remove (Plugin ∗plugin)
- Plugin ∗ find (std::string pluginTypeName)
- Plugin ∗ front ()
- Plugin ∗ next ()
- Plugin ∗ last ()

5.63.1 Constructor & Destructor Documentation

5.63.1.1 PluginManager::PluginManager (**Simulator** ∗ *simulator*)

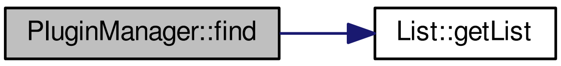
5.63.1.2 PluginManager::PluginManager (const **PluginManager** & *orig*)

5.63.1.3 PluginManager::∼PluginManager ( ) `[virtual]`

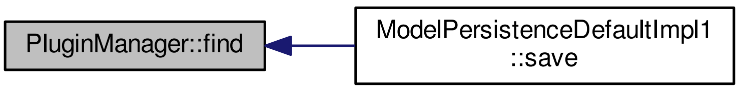
5.63.2 Member Function Documentation

5.63.2.1 **Plugin** ∗ PluginManager::find (std::string *pluginTypeName*)

Here is the call graph for this function:

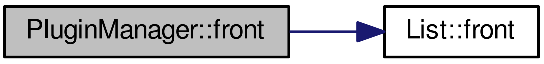


Here is the caller graph for this function:



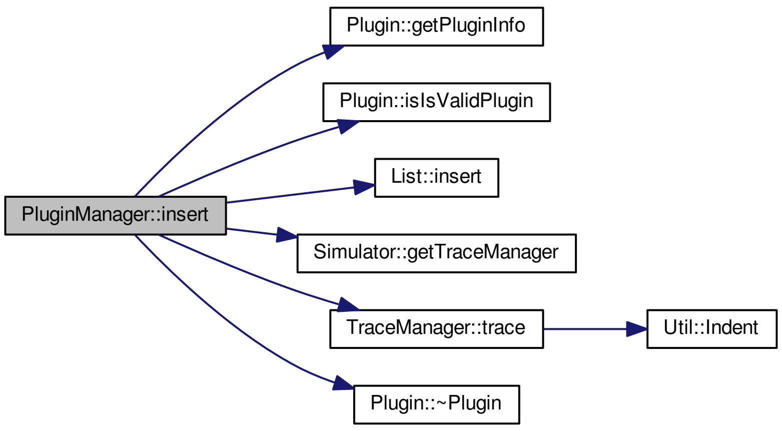
5.63.2.2 Plugin ∗ PluginManager::front ( )

Here is the call graph for this function:

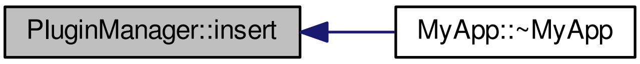


5.63.2.3 bool PluginManager::insert (Plugin ∗ *plugin*)

Here is the call graph for this function:

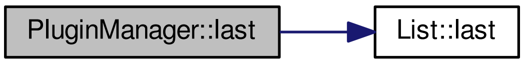


Here is the caller graph for this function:



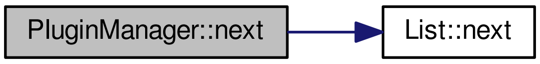
5.63.2.4 Plugin ∗ PluginManager::last ( )

Here is the call graph for this function:



5.63.2.5 Plugin ∗ PluginManager::next ( )

Here is the call graph for this function:



5.63.2.6 void PluginManager::remove (Plugin ∗ *plugin*)

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

- PluginManager.h
- PluginManager.cpp

5.64 ProbDistrib Class Reference

```
#include <ProbDistrib.h>
```

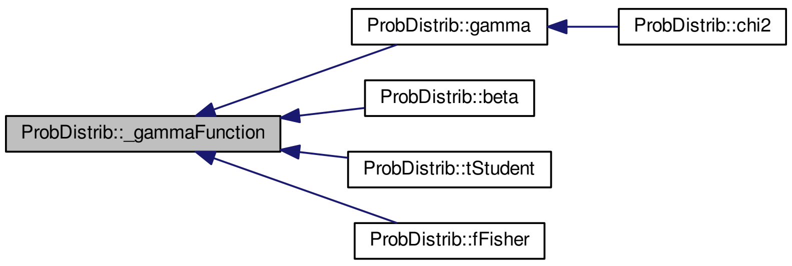
Static Public Member Functions

- static double uniform (double x, double min, double max)
- static double exponential (double x, double mean)
- static double erlang (double x, double mean, double M)
- static double normal (double x, double mean, double stddev)
- static double gamma (double x, unsigned int alpha, double beta)
- static double beta (double x, double alpha, double beta)
- static double weibull (double x, double alpha, double scale)
- static double logNormal (double x, double mean, double stddev)
- static double triangular (double x, double min, double mode, double max)
- static double tStudent (double x, double mean, double stddev, unsigned int degreeFreedom)
- static double fFisher (double x, double k, double m)
- static double chi2 (double x, double m)
- static double inverseNormal (double cumulativeProbability, double mean, double stddev)
- static double inverseTStudent (double cumulativeProbability, double mean, double stddev, double degree↩ Freedom)
- static double inverseFSnedecor (double cumulativeProbability, double u, double v)
- static double inverseChi2 (double cumulativeProbability, double m)
- static long double _gammaFunction (double z)

5.64.1 Member Function Documentation

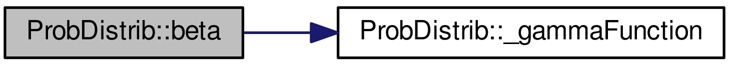
5.64.1.1 long double ProbDistrib::_gammaFunction (double *z*) [static]

Here is the caller graph for this function:



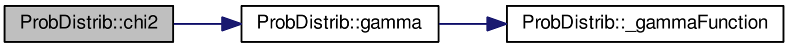
5.64.1.2 double ProbDistrib::beta (double *x*, double *alpha*, double *beta*) [static]

Here is the call graph for this function:



5.64.1.3 double ProbDistrib::chi2 (double *x*, double *m*) [static]

Here is the call graph for this function:

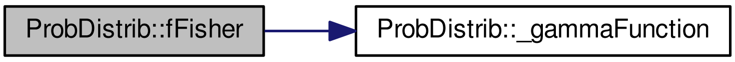


5.64.1.4 double ProbDistrib::erlang (double *x*, double *mean*, double *M*) [static]

5.64.1.5 double ProbDistrib::exponential (double *x*, double *mean*) [static]

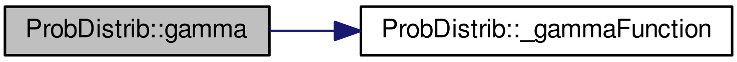
5.64.1.6 double ProbDistrib::fFisher (double *x*, double *k*, double *m*) [static]

Here is the call graph for this function:

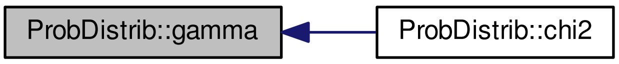


5.64.1.7 double ProbDistrib::gamma (double *x*, unsigned int *alpha*, double *beta*) [static]

Here is the call graph for this function:



Here is the caller graph for this function:



5.64.1.8 double ProbDistrib::inverseChi2 (double *cumulativeProbability,* double *m*) [static]

5.64.1.9 double ProbDistrib::inverseFSnedecor (double *cumulativeProbability,* double *u,* double *v*) [static]

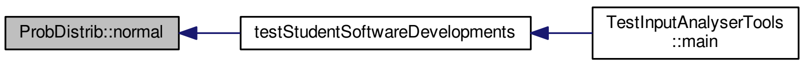
5.64.1.10 double ProbDistrib::inverseNormal (double *cumulativeProbability,* double *mean,* double *stddev*) [static]

5.64.1.11 double ProbDistrib::inverseTStudent (double *cumulativeProbability,* double *mean,* double *stddev,* double *degreeFreedom*) [static]

5.64.1.12 double ProbDistrib::logNormal (double *x,* double *mean,* double *stddev*) [static]

5.64.1.13 double ProbDistrib::normal (double *x,* double *mean,* double *stddev*) [static]

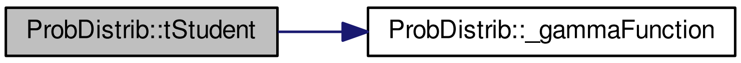
Here is the caller graph for this function:


ProbDistrib::normal
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.64.1.14 double ProbDistrib::triangular (double *x,* double *min,* double *mode,* double *max*) [static]

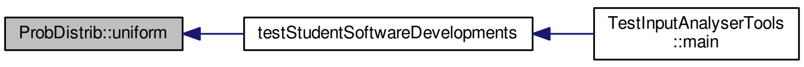
5.64.1.15 double ProbDistrib::tStudent (double *x,* double *mean,* double *stddev,* unsigned int *degreeFreedom*) [static]

Here is the call graph for this function:



5.64.1.16 double ProbDistrib::uniform (double *x,* double *min,* double *max*) [static]

Here is the caller graph for this function:


ProbDistrib::uniform
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.64.1.17 double ProbDistrib::weibull (double *x*, double *alpha*, double *scale*) [static]

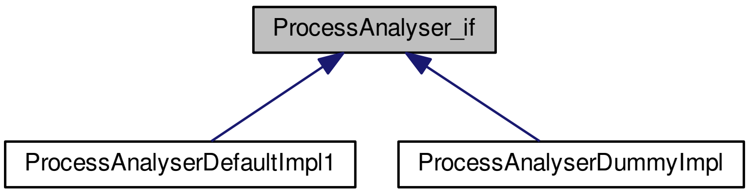
The documentation for this class was generated from the following files:

- ProbDistrib.h
- ProbDistrib.cpp

5.65 ProcessAnalyser_if Class Reference

```
#include <ProcessAnalyser_if.h>
```

Inheritance diagram for ProcessAnalyser_if:



Public Member Functions

- virtual List< SimulationScenario ∗ > ∗ getScenarios () const =0
- virtual List< SimulationControl ∗ > ∗ getControls () const =0
- virtual List< SimulationResponse ∗ > ∗ getResponses () const =0
- virtual List< SimulationControl ∗ > ∗ extractControlsFromModel (std::string modelFilename) const =0
- virtual List< SimulationResponse ∗ > ∗ extractResponsesFromModel (std::string modelFilename) const =0
- virtual void startSimulationOfScenario (SimulationScenario ∗scenario)=0
- virtual void startSimulation ()=0
- virtual void stopSimulation ()=0
- virtual void addTraceSimulationHandler (traceSimulationProcessListener traceSimulationProcessListener)=0

5.65.1 Detailed Description

The process analyser allows to extract controls and responses from a model, incluse some of then as controls and responses for a set of scenarios to be simulated

5.65.2 Member Function Documentation

5.65.2.1 virtual void ProcessAnalyser_if::addTraceSimulationHandler (traceSimulationProcessListener *traceSimulationProcessListener*) [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.2 virtual List<SimulationControl∗>∗ ProcessAnalyser_if::extractControlsFromModel (std::string *modelFilename*) const [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.3 virtual List<SimulationResponse∗>∗ ProcessAnalyser_if::extractResponsesFromModel (std::string *modelFilename*) const [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.4 virtual List<SimulationControl∗>∗ ProcessAnalyser_if::getControls ( ) const [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.5 virtual List<SimulationResponse∗>∗ ProcessAnalyser_if::getResponses ( ) const [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.6 virtual List<SimulationScenario∗>∗ ProcessAnalyser_if::getScenarios ( ) const [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.7 virtual void ProcessAnalyser_if::startSimulation ( ) [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.8 virtual void ProcessAnalyser_if::startSimulationOfScenario (SimulationScenario ∗ *scenario*) [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

5.65.2.9 virtual void ProcessAnalyser_if::stopSimulation ( ) [pure virtual]

Implemented in ProcessAnalyserDefaultImpl1, and ProcessAnalyserDummyImpl.

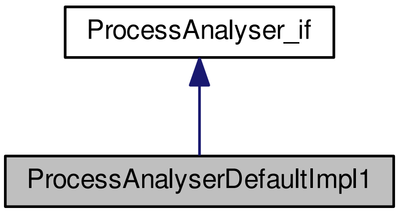
The documentation for this class was generated from the following file:

- ProcessAnalyser_if.h

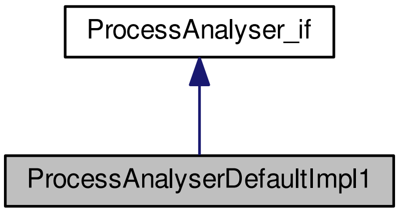
5.66 ProcessAnalyserDefaultImpl1 Class Reference

```
#include <ProcessAnalyserDefaultImpl1.h>
```

Inheritance diagram for ProcessAnalyserDefaultImpl1:



Collaboration diagram for ProcessAnalyserDefaultImpl1:



Public Member Functions

- ProcessAnalyserDefaultImpl1 ()
- ProcessAnalyserDefaultImpl1 (const ProcessAnalyserDefaultImpl1 &orig)
- virtual ∼ProcessAnalyserDefaultImpl1 ()
- virtual List< SimulationScenario ∗ > ∗ getScenarios () const
- virtual List< SimulationControl ∗ > ∗ getControls () const
- virtual List< SimulationResponse ∗ > ∗ getResponses () const
- virtual List< SimulationControl ∗ > ∗ extractControlsFromModel (std::string modelFilename) const
- virtual List< SimulationResponse ∗ > ∗ extractResponsesFromModel (std::string modelFilename) const
- virtual void startSimulationOfScenario (SimulationScenario ∗scenario)
- virtual void startSimulation ()
- virtual void stopSimulation ()
- virtual void addTraceSimulationHandler (traceSimulationProcessListener traceSimulationProcessListener)

5.66.1 Constructor & Destructor Documentation

5.66.1.1 ProcessAnalyserDefaultImpl1::ProcessAnalyserDefaultImpl1 ( )

5.66.1.2 ProcessAnalyserDefaultImpl1::ProcessAnalyserDefaultImpl1 (const ProcessAnalyserDefaultImpl1 & *orig*)

5.66.1.3 ProcessAnalyserDefaultImpl1::∼ProcessAnalyserDefaultImpl1 ( ) [virtual]

5.66.2 Member Function Documentation

5.66.2.1 void ProcessAnalyserDefaultImpl1::addTraceSimulationHandler (traceSimulationProcessListener *traceSimulationProcessListener*) [virtual]

Implements ProcessAnalyser_if.

5.66.2.2 List< SimulationControl ∗ > ∗ ProcessAnalyserDefaultImpl1::extractControlsFromModel (std::string *modelFilename*) const [virtual]

Implements ProcessAnalyser_if.

5.66.2.3 List< SimulationResponse ∗ > ∗ ProcessAnalyserDefaultImpl1::extractResponsesFromModel (std::string *modelFilename*) const [virtual]

Implements ProcessAnalyser_if.

5.66.2.4 List< SimulationControl ∗ > ∗ ProcessAnalyserDefaultImpl1::getControls ( ) const [virtual]

Implements ProcessAnalyser_if.

5.66.2.5 List< SimulationResponse ∗ > ∗ ProcessAnalyserDefaultImpl1::getResponses ( ) const [virtual]

Implements ProcessAnalyser_if.

5.66.2.6 List< SimulationScenario ∗ > ∗ ProcessAnalyserDefaultImpl1::getScenarios ( ) const [virtual]

Implements ProcessAnalyser_if.

5.66.2.7 void ProcessAnalyserDefaultImpl1::startSimulation ( ) [virtual]

Implements ProcessAnalyser_if.

5.66.2.8 void ProcessAnalyserDefaultImpl1::startSimulationOfScenario (SimulationScenario * *scenario*) [virtual]

Implements ProcessAnalyser_if.

5.66.2.9 void ProcessAnalyserDefaultImpl1::stopSimulation ( ) [virtual]

Implements ProcessAnalyser_if.

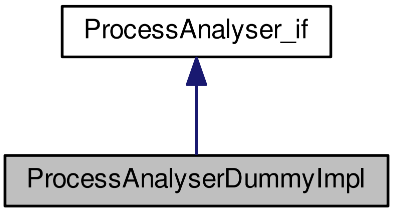
The documentation for this class was generated from the following files:

- ProcessAnalyserDefaultImpl1.h
- ProcessAnalyserDefaultImpl1.cpp

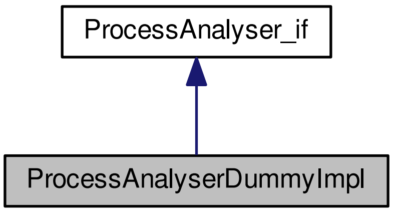
5.67 ProcessAnalyserDummyImpl Class Reference

```
#include <ProcessAnalyserDummyImpl.h>
```

Inheritance diagram for ProcessAnalyserDummyImpl:



Collaboration diagram for ProcessAnalyserDummyImpl:



Public Member Functions

- ProcessAnalyserDummyImpl ()
- ProcessAnalyserDummyImpl (const ProcessAnalyserDummyImpl &orig)
- ∼ProcessAnalyserDummyImpl ()
- virtual List< SimulationScenario ∗ > ∗ getScenarios () const
- virtual List< SimulationControl ∗ > ∗ getControls () const
- virtual List< SimulationResponse ∗ > ∗ getResponses () const
- virtual List< SimulationControl ∗ > ∗ extractControlsFromModel (std::string modelFilename) const
- virtual List< SimulationResponse ∗ > ∗ extractResponsesFromModel (std::string modelFilename) const
- virtual void startSimulationOfScenario (SimulationScenario ∗scenario)
- virtual void startSimulation ()
- virtual void stopSimulation ()
- virtual void addTraceSimulationHandler (traceSimulationProcessListener traceSimulationProcessListener)

5.67.1 Constructor & Destructor Documentation

5.67.1.1 ProcessAnalyserDummyImpl::ProcessAnalyserDummyImpl ( )

5.67.1.2 ProcessAnalyserDummyImpl::ProcessAnalyserDummyImpl (const **ProcessAnalyserDummyImpl** & *orig*)

5.67.1.3 ProcessAnalyserDummyImpl::∼ProcessAnalyserDummyImpl ( )

5.67.2 Member Function Documentation

5.67.2.1 void ProcessAnalyserDummyImpl::addTraceSimulationHandler (**traceSimulationProcessListener** *traceSimulationProcessListener*) [virtual]

Implements ProcessAnalyser_if.

5.67.2.2 **List**< **SimulationControl** ∗ > ∗ ProcessAnalyserDummyImpl::extractControlsFromModel (std::string *modelFilename*) const [virtual]

Implements ProcessAnalyser_if.

5.67.2.3 **List**< **SimulationResponse** ∗ > ∗ ProcessAnalyserDummyImpl::extractResponsesFromModel (std::string *modelFilename*) const [virtual]

Implements ProcessAnalyser_if.

5.67.2.4 **List**< **SimulationControl** ∗ > ∗ ProcessAnalyserDummyImpl::getControls ( ) const [virtual]

Implements ProcessAnalyser_if.

5.67.2.5 **List**< **SimulationResponse** ∗ > ∗ ProcessAnalyserDummyImpl::getResponses ( ) const [virtual]

Implements ProcessAnalyser_if.

5.67.2.6 List< SimulationScenario * > * ProcessAnalyserDummyImpl::getScenarios ( ) const [virtual]

Implements ProcessAnalyser_if.

5.67.2.7 void ProcessAnalyserDummyImpl::startSimulation ( ) [virtual]

Implements ProcessAnalyser_if.

5.67.2.8 void ProcessAnalyserDummyImpl::startSimulationOfScenario (SimulationScenario * *scenario*) [virtual]

Implements ProcessAnalyser_if.

5.67.2.9 void ProcessAnalyserDummyImpl::stopSimulation ( ) [virtual]

Implements ProcessAnalyser_if.

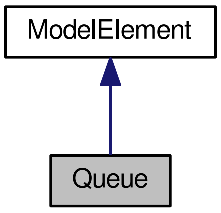
The documentation for this class was generated from the following files:

- ProcessAnalyserDummyImpl.h
- ProcessAnalyserDummyImpl.cpp

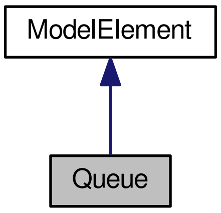
5.68 Queue Class Reference

`#include <Queue.h>`

Inheritance diagram for Queue:



Collaboration diagram for Queue:



Public Types

- enum OrderRule : int { OrderRule::FIFO = 1, OrderRule::LIFO = 2, OrderRule::HIGHESTVALUE = 3, Order↩ Rule::SMALLESTVALUE = 4 }

Public Member Functions

- Queue (ElementManager ∗elems)
- Queue (ElementManager ∗elems, std::string name)
- Queue (const Queue &orig)
- virtual ∼Queue ()
- virtual std::string show ()
- void insertElement (Waiting ∗element)
- void removeElement (Waiting ∗element, double tnow)
- void initBetweenReplication ()
- unsigned int size ()
- Waiting ∗ first ()
- void setAttributeName (std::string _attributeName)
- std::string getAttributeName () const
- void setOrderRule (OrderRule _orderRule)
- Queue::OrderRule getOrderRule () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.68.1 Member Enumeration Documentation

5.68.1.1 enum **Queue::OrderRule** : int [strong]

Enumerator

FIFO

LIFO

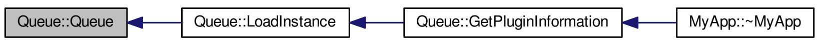
HIGHESTVALUE

SMALLESTVALUE

5.68.2 Constructor & Destructor Documentation

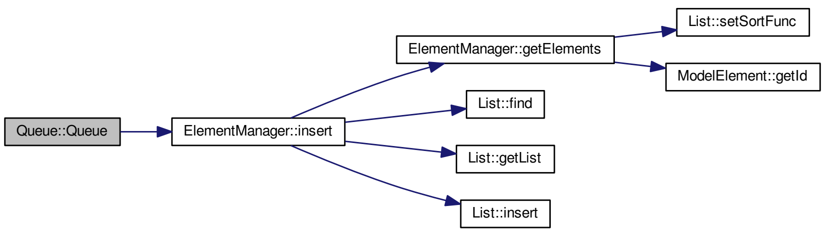
5.68.2.1 Queue::Queue (**ElementManager** ∗ *elems*)

Here is the caller graph for this function:



5.68.2.2 Queue::Queue (**ElementManager** ∗ *elems,* std::string *name*)

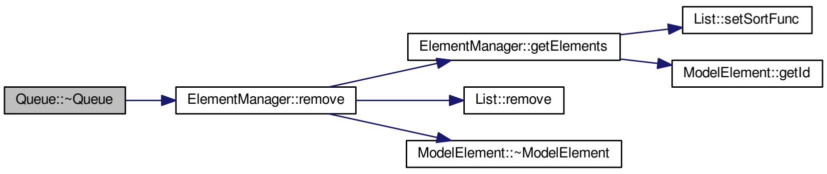
Here is the call graph for this function:



5.68.2.3 Queue::Queue (const Queue & *orig*)

5.68.2.4 Queue::∼Queue ( ) [virtual]

Here is the call graph for this function:

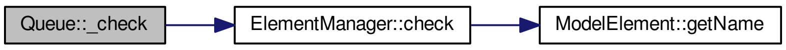


5.68.3 Member Function Documentation

5.68.3.1 bool Queue::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

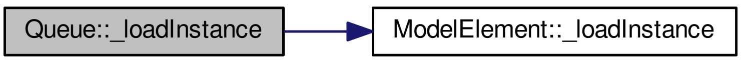
Here is the call graph for this function:



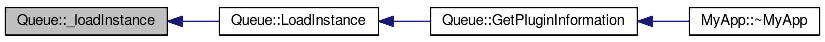
5.68.3.2 bool Queue::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



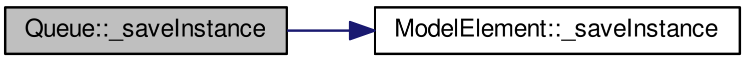
Here is the caller graph for this function:



5.68.3.3 std::map< std::string, std::string > * Queue::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:

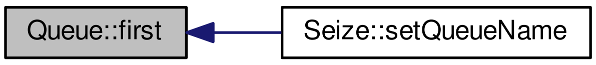


5.68.3.4 Waiting * Queue::first ( )

Here is the call graph for this function:

Queue::first
List::front

Here is the caller graph for this function:

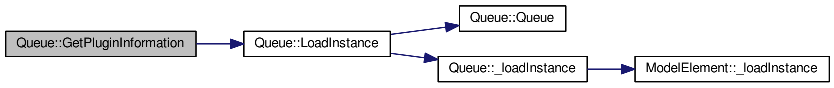


5.68.3.5 std::string Queue::getAttributeName ( ) const

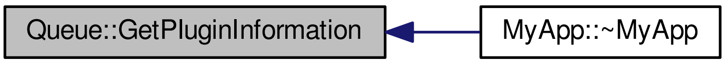
5.68.3.6 Queue::OrderRule Queue::getOrderRule ( ) const

5.68.3.7 PluginInformation ∗ Queue::GetPluginInformation ( ) [static]

Here is the call graph for this function:

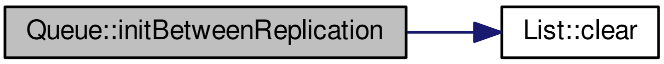


Here is the caller graph for this function:

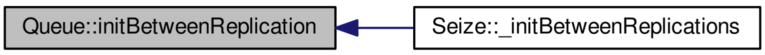


5.68.3.8 void Queue::initBetweenReplication ( )

Here is the call graph for this function:

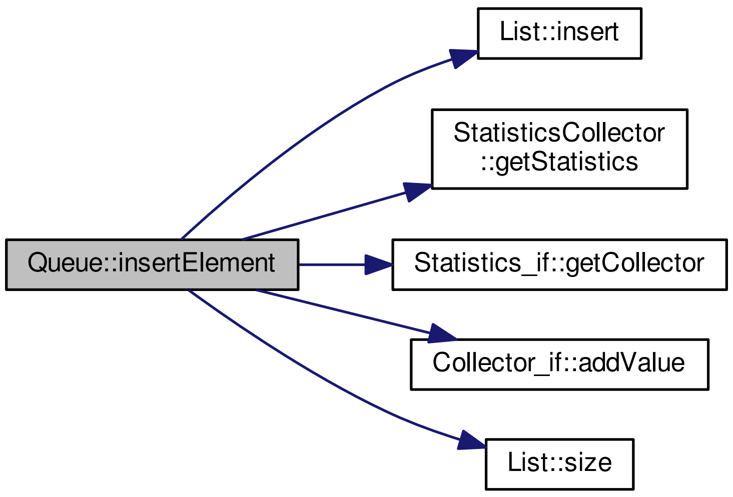


Here is the caller graph for this function:

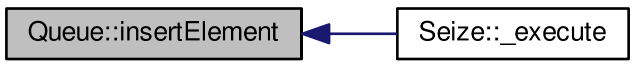

Queue::initBetweenReplication
Seize::_initBetweenReplications

5.68.3.9 void Queue::insertElement (**Waiting** $*$ *element*)

Here is the call graph for this function:

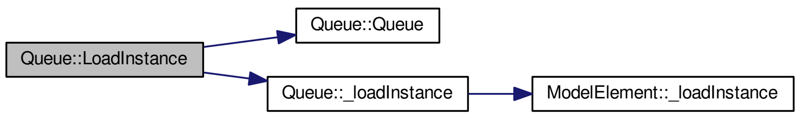


Here is the caller graph for this function:

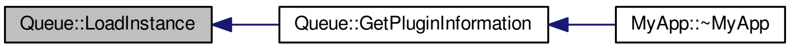


5.68.3.10 **ModelElement** $*$ Queue::LoadInstance (**ElementManager** $*$ *elems,* std::map$<$ std::string, std::string $>$ $*$ *fields*) [static]

Here is the call graph for this function:

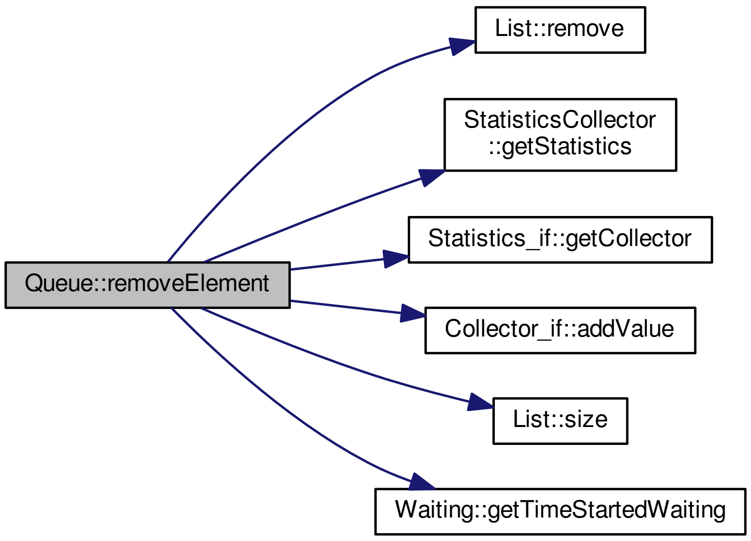


Here is the caller graph for this function:

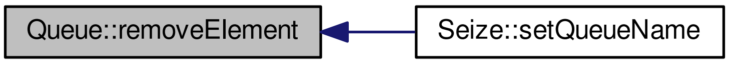


5.68.3.11 void Queue::removeElement (Waiting ∗ *element,* double *tnow*)

Here is the call graph for this function:


Queue::removeElement
List::remove
StatisticsCollector ::getStatistics
Statistics_if::getCollector
Collector_if::addValue
List::size
Waiting::getTimeStartedWaiting

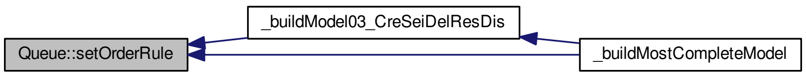
Here is the caller graph for this function:



5.68.3.12 void Queue::setAttributeName (std::string *_attributeName*)

5.68.3.13 void Queue::setOrderRule (OrderRule *_orderRule*)

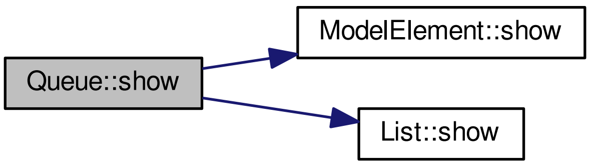
Here is the caller graph for this function:



5.68.3.14 std::string Queue::show ( ) [virtual]

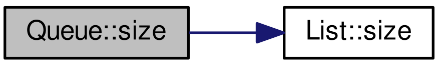
Reimplemented from ModelElement.

Here is the call graph for this function:

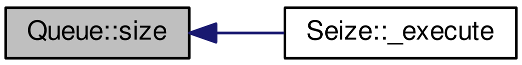


5.68.3.15 unsigned int Queue::size ( )

Here is the call graph for this function:



Here is the caller graph for this function:



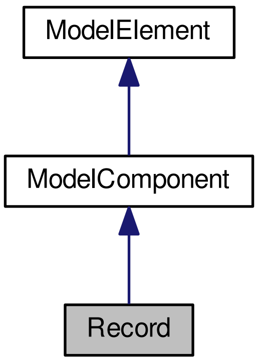
The documentation for this class was generated from the following files:

- Queue.h
- Queue.cpp

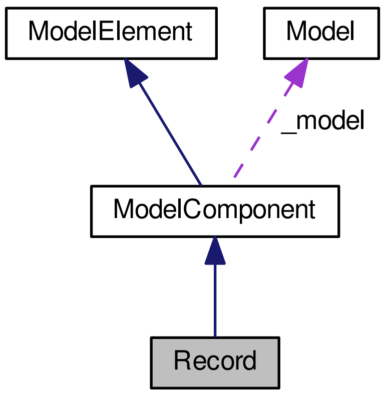
5.69 Record Class Reference

```
#include <Record.h>
```

Inheritance diagram for Record:



Collaboration diagram for Record:



Public Member Functions

- Record (Model ∗model)
- Record (const Record &orig)
- virtual ∼Record ()
- void setFilename (std::string filename)
- std::string getFilename () const
- void setExpression (std::string expression)
- std::string getExpression () const
- void setExpressionName (std::string expressionName)
- std::string getExpressionName () const
- StatisticsCollector ∗ getCstatExpression () const
- virtual std::string show ()

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

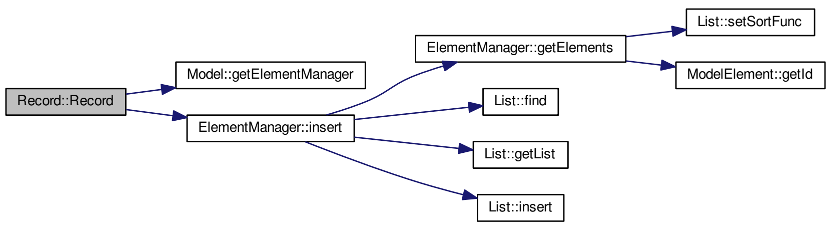
- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

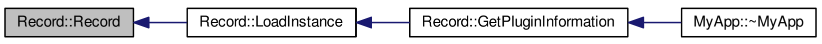
5.69.1 Constructor & Destructor Documentation

5.69.1.1 Record::Record (Model ∗ *model*)

Here is the call graph for this function:



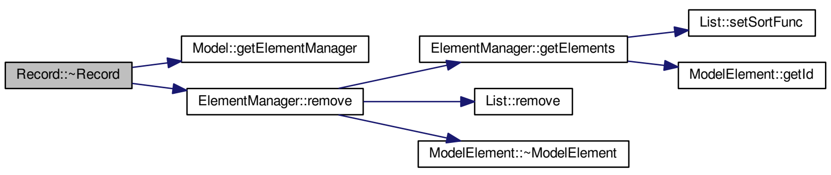
Here is the caller graph for this function:



5.69.1.2 Record::Record (const Record & *orig*)

5.69.1.3 Record::∼Record ( ) [virtual]

Here is the call graph for this function:

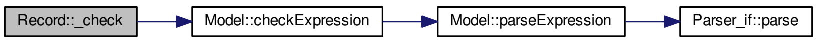


5.69.2 Member Function Documentation

5.69.2.1 bool Record::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

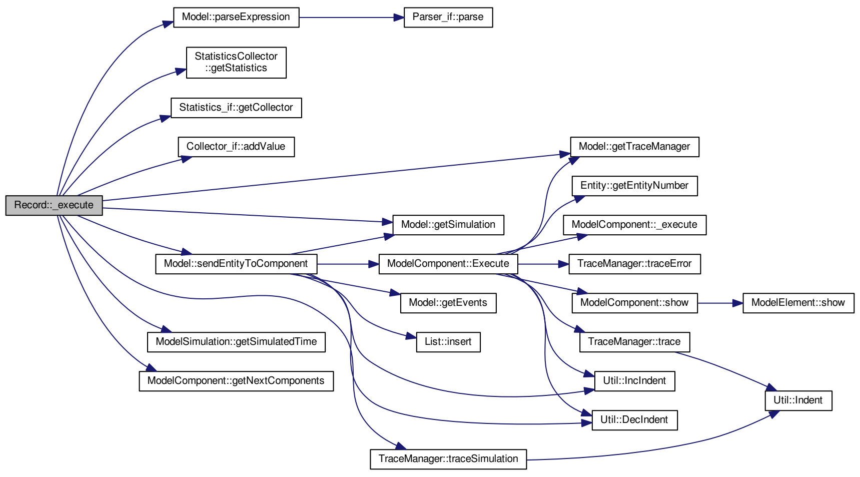
Here is the call graph for this function:



5.69.2.2 void Record::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

Here is the call graph for this function:



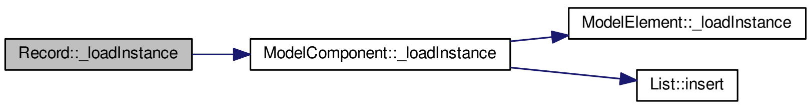
5.69.2.3 void Record::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

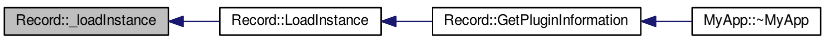
5.69.2.4 bool Record::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



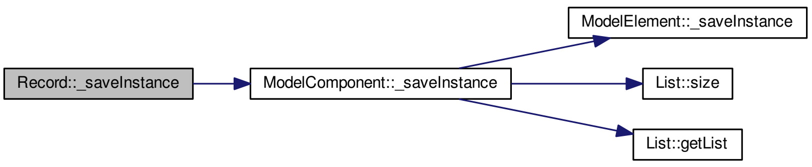
Here is the caller graph for this function:



5.69.2.5 std::map< std::string, std::string > * Record::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



5.69.2.6 StatisticsCollector * Record::getCstatExpression ( ) const

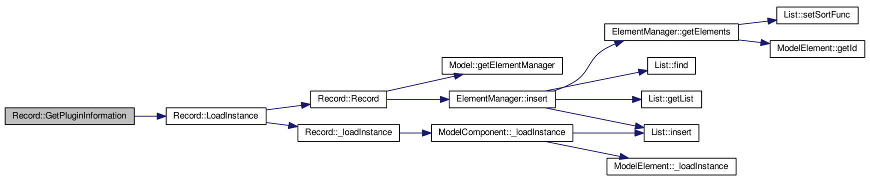
5.69.2.7 std::string Record::getExpression ( ) const

5.69.2.8 std::string Record::getExpressionName ( ) const

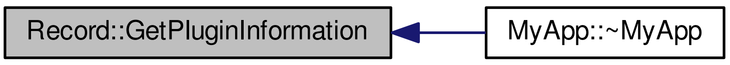
5.69.2.9 std::string Record::getFilename ( ) const

5.69.2.10 PluginInformation * Record::GetPluginInformation ( ) [static]

Here is the call graph for this function:

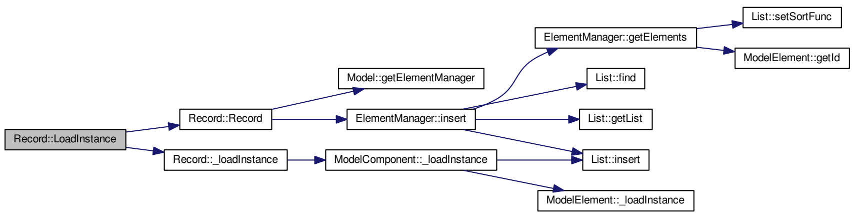


Here is the caller graph for this function:

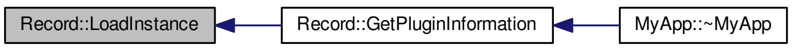


5.69.2.11 ModelComponent * Record::LoadInstance (Model * *model*, std::map< std::string, std::string > * *fields*) [static]

Here is the call graph for this function:

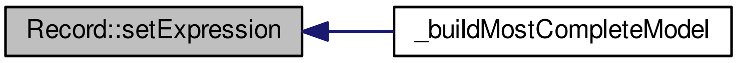


Here is the caller graph for this function:



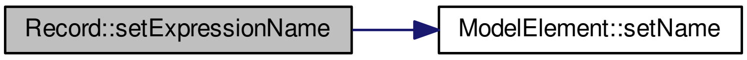
5.69.2.12 void Record::setExpression (std::string *expression*)

Here is the caller graph for this function:

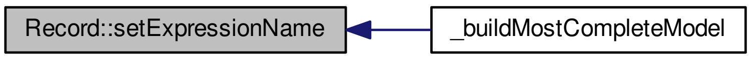


5.69.2.13 void Record::setExpressionName (std::string *expressionName*)

Here is the call graph for this function:

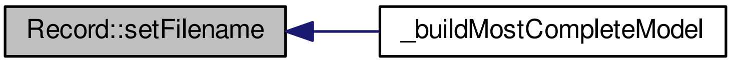


Here is the caller graph for this function:



5.69.2.14 void Record::setFilename (std::string *filename*)

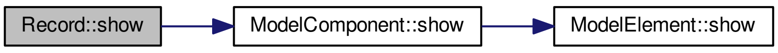
Here is the caller graph for this function:



5.69.2.15 std::string Record::show ( ) [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



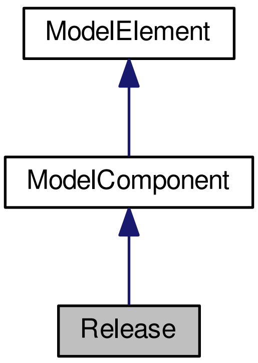
The documentation for this class was generated from the following files:

- Record.h
- Record.cpp

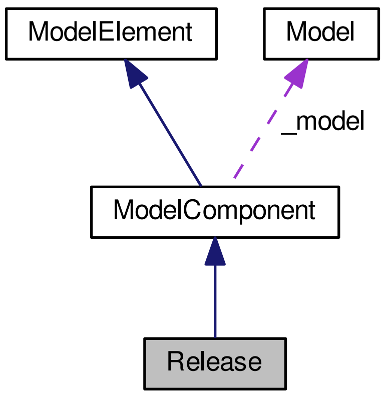
5.70 Release Class Reference

```
#include <Release.h>
```

Inheritance diagram for Release:



Collaboration diagram for Release:



Public Member Functions

- Release (Model ∗model)
- Release (const Release &orig)
- virtual ∼Release ()
- virtual std::string show ()
- void setPriority (unsigned short _priority)
- unsigned short getPriority () const
- void setResourceType (Resource::ResourceType _resourceType)
- Resource::ResourceType getResourceType () const
- void setQuantity (std::string _quantity)
- std::string getQuantity () const
- void setRule (Resource::ResourceRule _rule)
- Resource::ResourceRule getRule () const
- void setSaveAttribute (std::string _saveAttribute)
- std::string getSaveAttribute () const
- void setResource (Resource ∗_resource)
- Resource ∗ getResource () const
- void setResourceName (std::string resourceName) throw ()
- std::string getResourceName () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

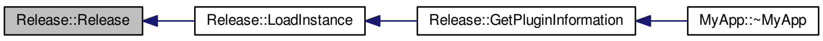
- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.70.1 Constructor & Destructor Documentation

5.70.1.1 Release::Release (Model ∗ *model*)

Here is the caller graph for this function:



5.70.1.2 Release::Release (const Release & *orig*)

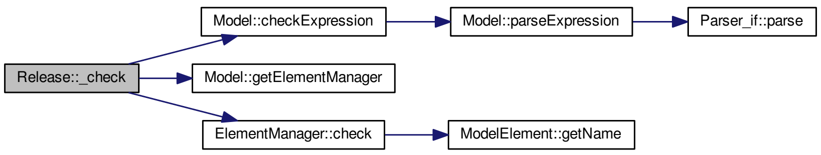
5.70.1.3 Release::∼Release ( ) [virtual]

5.70.2 Member Function Documentation

5.70.2.1 bool Release::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

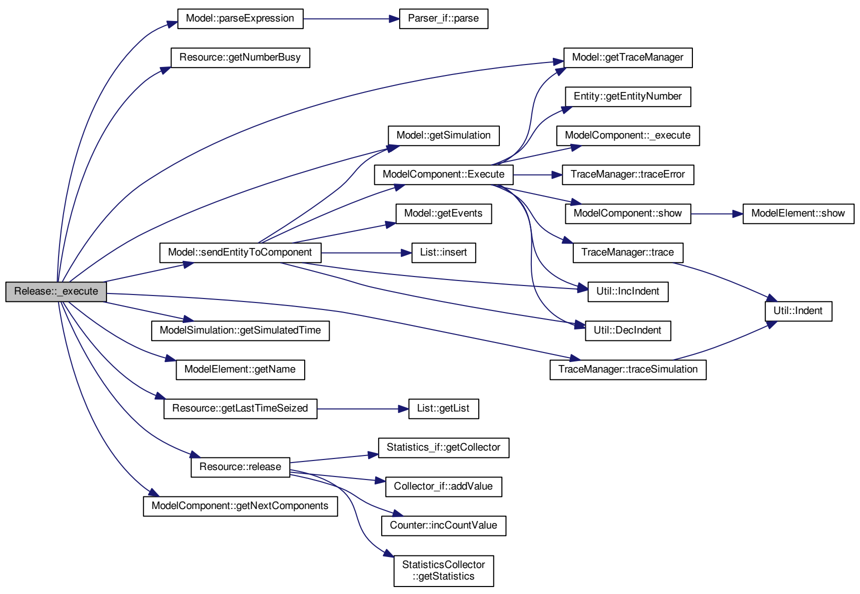
Here is the call graph for this function:



5.70.2.2 void Release::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

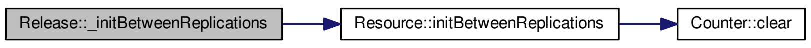
Here is the call graph for this function:



5.70.2.3 void Release::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

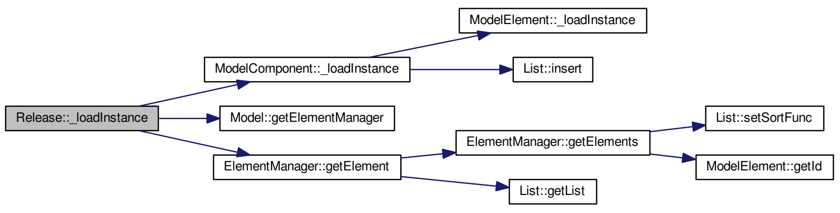
Here is the call graph for this function:


Release::_initBetweenReplications
Resource::initBetweenReplications
Counter::clear

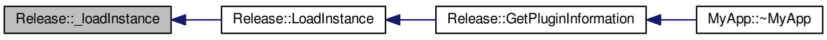
5.70.2.4 bool Release::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:


ModelElement::_loadInstance
ModelComponent::_loadInstance
List::insert
Release::_loadInstance
Model::getElementManager
List::setSortFunc
ElementManager::getElements
ElementManager::getElement
ModelElement::getId
List::getList

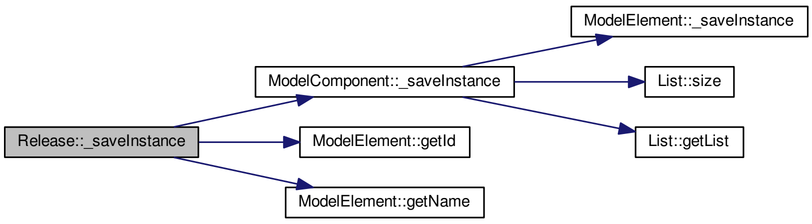
Here is the caller graph for this function:



5.70.2.5 std::map< std::string, std::string > * Release::_saveInstance ( ) [protected], [virtual]

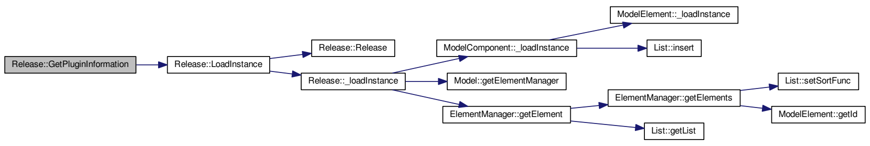
Reimplemented from ModelComponent.

Here is the call graph for this function:

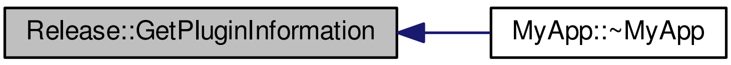


5.70.2.6 PluginInformation * Release::GetPluginInformation ( ) [static]

Here is the call graph for this function:



Here is the caller graph for this function:



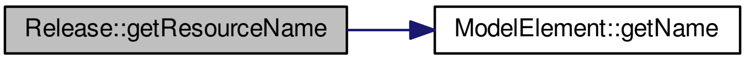
5.70.2.7 unsigned short Release::getPriority ( ) const

5.70.2.8 std::string Release::getQuantity ( ) const

5.70.2.9 **Resource** ∗ Release::getResource ( ) const

5.70.2.10 std::string Release::getResourceName ( ) const

Here is the call graph for this function:



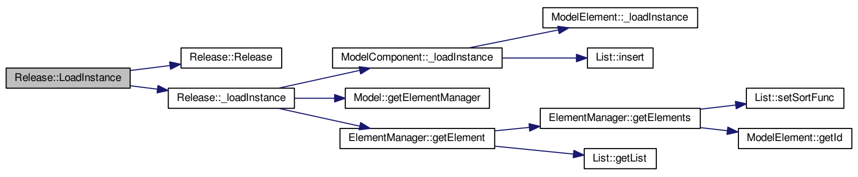
5.70.2.11 **Resource::ResourceType** Release::getResourceType ( ) const

5.70.2.12 **Resource::ResourceRule** Release::getRule ( ) const

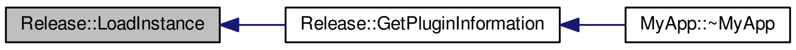
5.70.2.13 std::string Release::getSaveAttribute ( ) const

5.70.2.14 **ModelComponent** ∗ Release::LoadInstance (**Model** ∗ *model,* std::map< std::string, std::string > ∗ *fields*) `[static]`

Here is the call graph for this function:



Here is the caller graph for this function:

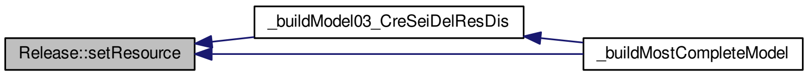


5.70.2.15 void Release::setPriority (unsigned short *_priority*)

5.70.2.16 void Release::setQuantity (std::string *_quantity*)

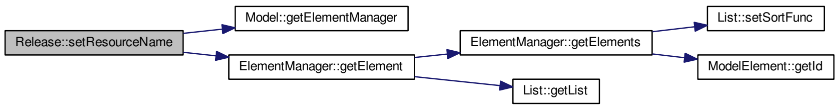
5.70.2.17 void Release::setResource (Resource ∗ *_resource*)

Here is the caller graph for this function:



5.70.2.18 void Release::setResourceName (std::string *resourceName*) throw)

Here is the call graph for this function:



5.70.2.19 void Release::setResourceType (Resource::ResourceType *_resourceType*)

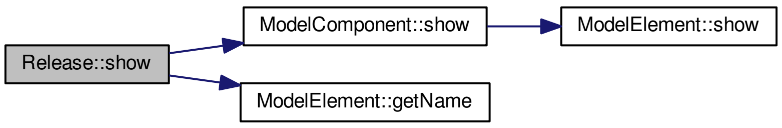
5.70.2.20 void Release::setRule (Resource::ResourceRule *_rule*)

5.70.2.21 void Release::setSaveAttribute (std::string *_saveAttribute*)

5.70.2.22 std::string Release::show ( ) [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



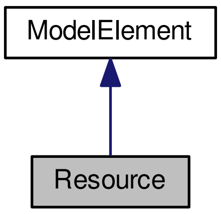
The documentation for this class was generated from the following files:

- Release.h
- Release.cpp

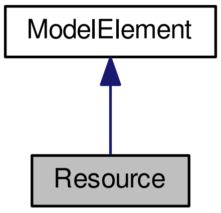
5.71 Resource Class Reference

```
#include <Resource.h>
```

Inheritance diagram for Resource:



Collaboration diagram for Resource:



Public Types

- enum ResourceType : int { ResourceType::SET = 1, ResourceType::RESOURCE = 2 }
- enum ResourceRule : int {
  ResourceRule::RANDOM = 1, ResourceRule::CICLICAL = 2, ResourceRule::ESPECIFIC = 3, Resource↩
  Rule::SMALLESTBUSY = 4,
  ResourceRule::LARGESTREMAININGCAPACITY = 5 }
- enum ResourceState : int {
  ResourceState::IDLE = 1, ResourceState::BUSY = 2, ResourceState::FAILED = 3, ResourceState::INACT↩
  IVE = 4,
  ResourceState::OTHER = 5 }
- typedef std::function< void(Resource ∗) > ResourceEventHandler

Public Member Functions

- Resource (ElementManager ∗elems)
- Resource (ElementManager ∗elems, std::string name)
- Resource (const Resource &orig)
- virtual ∼Resource ()
- virtual std::string show ()
- void seize (unsigned int quantity, double tnow)
- void release (unsigned int quantity, double tnow)
- void initBetweenReplications ()
- void setResourceState (ResourceState _resourceState)
- Resource::ResourceState getResourceState () const
- void setCapacity (unsigned int _capacity)
- unsigned int getCapacity () const
- void setCostBusyHour (double _costBusyHour)
- double getCostBusyHour () const
- void setCostIdleHour (double _costIdleHour)
- double getCostIdleHour () const
- void setCostPerUse (double _costPerUse)
- double getCostPerUse () const
- unsigned int getNumberBusy () const
- void addResourceEventHandler (ResourceEventHandler eventHandler)
- double getLastTimeSeized () const

Static Public Member Functions

- template<typename Class >
  static ResourceEventHandler SetResourceEventHandler (void(Class::∗function)(Resource ∗), Class ∗object)
- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.71.1 Detailed Description

Resource represents a facility that...

5.71.2 Member Typedef Documentation

5.71.2.1 typedef std::function<void(Resource∗) > Resource::ResourceEventHandler

5.71.3 Member Enumeration Documentation

5.71.3.1 enum Resource::ResourceRule : int [strong]

Enumerator

RANDOM

CICLICAL

ESPECIFIC

SMALLESTBUSY

LARGESTREMAININGCAPACITY

5.71.3.2 enum Resource::ResourceState : int [strong]

Enumerator

IDLE

BUSY

FAILED

INACTIVE

OTHER

5.71.3.3 enum Resource::ResourceType : int [strong]

Enumerator

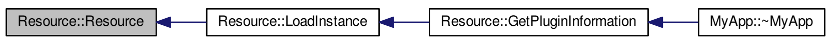
SET

RESOURCE

5.71.4 Constructor & Destructor Documentation

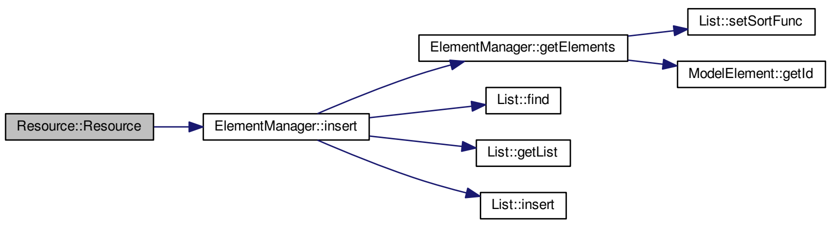
5.71.4.1 Resource::Resource (ElementManager ∗ *elems*)

Here is the caller graph for this function:



5.71.4.2 Resource::Resource (ElementManager ∗ *elems,* std::string *name*)

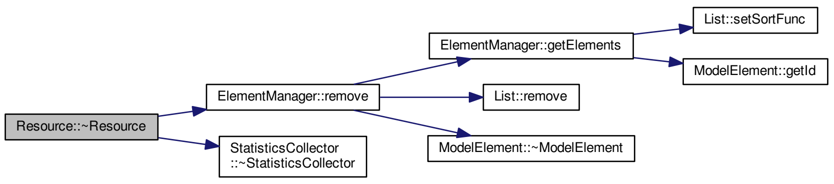
Here is the call graph for this function:



5.71.4.3 Resource::Resource (const Resource & *orig*)

5.71.4.4 Resource::∼Resource ( ) [virtual]

Here is the call graph for this function:


Resource::~Resource
ElementManager::remove
StatisticsCollector ::~StatisticsCollector
ElementManager::getElements
List::setSortFunc
ModelElement::getId
List::remove
ModelElement::~ModelElement

5.71.5 Member Function Documentation

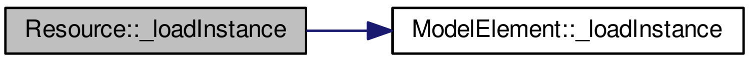
5.71.5.1 bool Resource::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

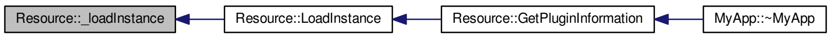
5.71.5.2 bool Resource::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



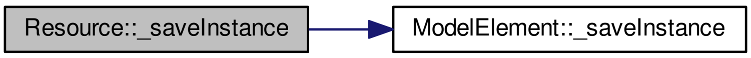
Here is the caller graph for this function:



5.71.5.3 std::map< std::string, std::string > ∗ Resource::_saveInstance ( ) [protected], [virtual]

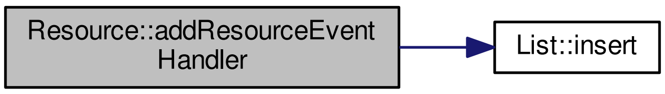
Reimplemented from ModelElement.

Here is the call graph for this function:

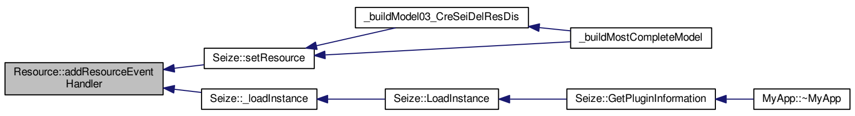


5.71.5.4 void Resource::addResourceEventHandler (ResourceEventHandler *eventHandler*)

Here is the call graph for this function:

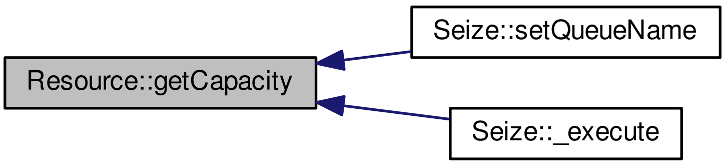


Here is the caller graph for this function:



5.71.5.5 unsigned int Resource::getCapacity ( ) const

Here is the caller graph for this function:



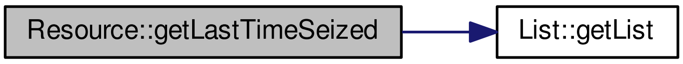
5.71.5.6 double Resource::getCostBusyHour ( ) const

5.71.5.7 double Resource::getCostIdleHour ( ) const

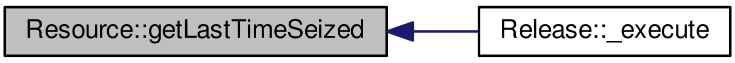
5.71.5.8 double Resource::getCostPerUse ( ) const

5.71.5.9 double Resource::getLastTimeSeized ( ) const

Here is the call graph for this function:

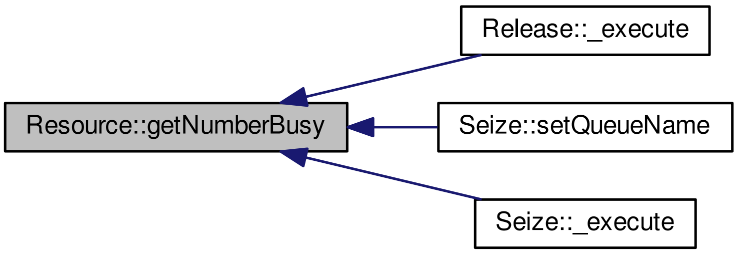


Here is the caller graph for this function:



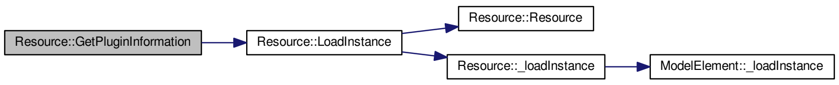
5.71.5.10 unsigned int Resource::getNumberBusy ( ) const

Here is the caller graph for this function:

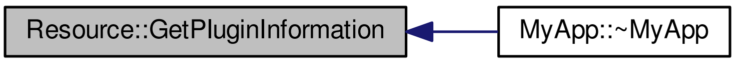


5.71.5.11 PluginInformation ∗ Resource::GetPluginInformation ( ) [static]

Here is the call graph for this function:



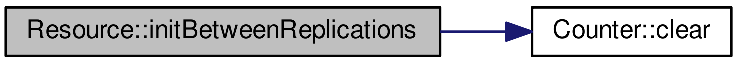
Here is the caller graph for this function:



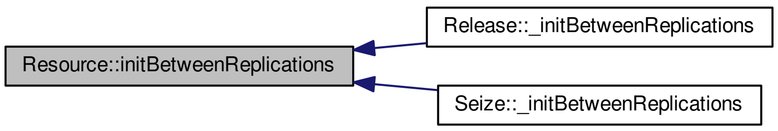
5.71.5.12 Resource::ResourceState Resource::getResourceState ( ) const

5.71.5.13 void Resource::initBetweenReplications ( )

Here is the call graph for this function:

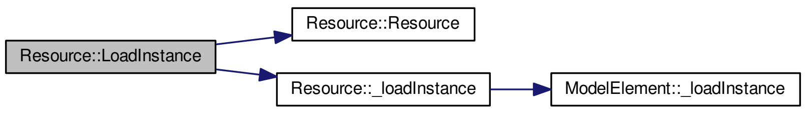


Here is the caller graph for this function:

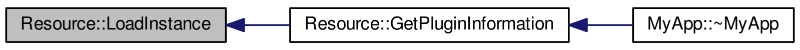

Release::_initBetweenReplications
Resource::initBetweenReplications
Seize::_initBetweenReplications

5.71.5.14 **ModelElement** ∗ Resource::LoadInstance (**ElementManager** ∗ *elems*, std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:

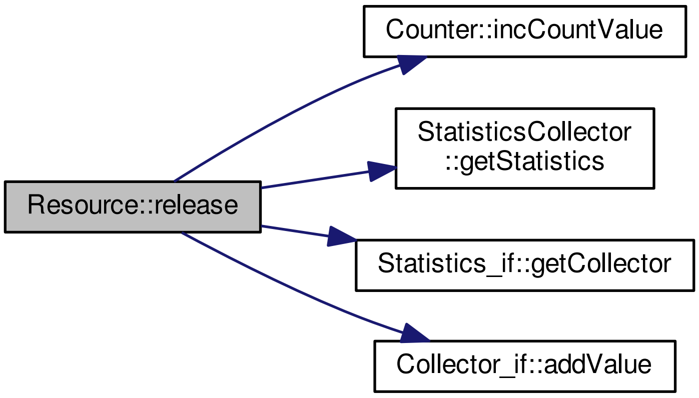


Here is the caller graph for this function:

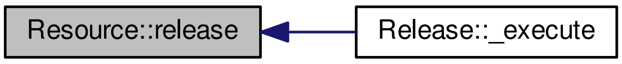


5.71.5.15 void Resource::release (unsigned int *quantity*, double *tnow*)

Here is the call graph for this function:

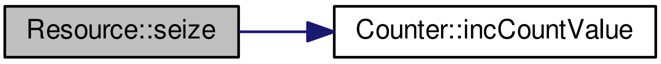

Counter::incCountValue
StatisticsCollector ::getStatistics
Resource::release
Statistics_if::getCollector
Collector_if::addValue

Here is the caller graph for this function:

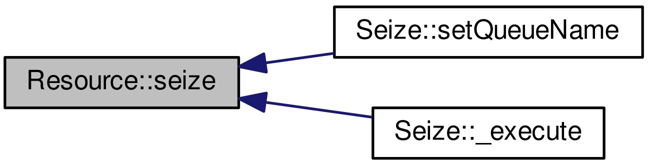


5.71.5.16 void Resource::seize (unsigned int *quantity*, double *tnow*)

Here is the call graph for this function:

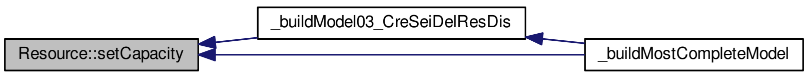


Here is the caller graph for this function:



5.71.5.17 void Resource::setCapacity (unsigned int *_capacity*)

Here is the caller graph for this function:



5.71.5.18 void Resource::setCostBusyHour (double *_costBusyHour*)

5.71.5.19 void Resource::setCostIdleHour (double *_costIdleHour*)

5.71.5.20 void Resource::setCostPerUse (double *_costPerUse*)

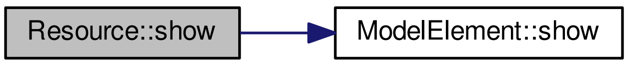
5.71.5.21 template<typename Class > static ResourceEventHandler Resource::SetResourceEventHandler (void(Class::∗)(Resource ∗) *function,* Class ∗ *object*) [inline], [static]

5.71.5.22 void Resource::setResourceState (ResourceState *_resourceState*)

5.71.5.23 std::string Resource::show ( ) [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



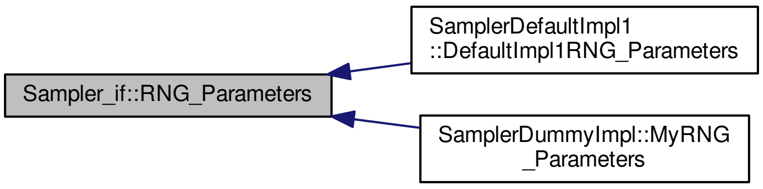
The documentation for this class was generated from the following files:

- Resource.h
- Resource.cpp

5.72 Sampler_if::RNG_Parameters Struct Reference

```
#include <Sampler_if.h>
```

Inheritance diagram for Sampler_if::RNG_Parameters:


SamplerDefaultImpl1 ::DefaultImpl1RNG_Parameters
Sampler_if::RNG_Parameters
SamplerDummyImpl::MyRNG _Parameters

Public Member Functions

- virtual ∼RNG_Parameters ()=default

5.72.1 Detailed Description

class that encapsulates attributes required to generate random numbers, which depends on the generation method used.

5.72.2 Constructor & Destructor Documentation

5.72.2.1 virtual Sampler_if::RNG_Parameters::∼RNG_Parameters ( ) `[virtual], [default]`

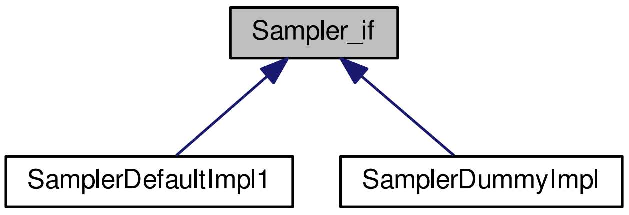
The documentation for this struct was generated from the following file:

- Sampler_if.h

5.73 Sampler_if Class Reference

```
#include <Sampler_if.h>
```

Inheritance diagram for Sampler_if:



Classes

- struct RNG_Parameters

Public Member Functions

- virtual double random ()=0
- virtual double sampleUniform (double min, double max)=0
- virtual double sampleExponential (double mean)=0
- virtual double sampleErlang (double mean, int M)=0
- virtual double sampleNormal (double mean, double stddev)=0
- virtual double sampleGamma (double mean, double alpha)=0
- virtual double sampleBeta (double alpha, double beta, double infLimit, double supLimit)=0
- virtual double sampleWeibull (double alpha, double scale)=0
- virtual double sampleLogNormal (double mean, double stddev)=0
- virtual double sampleTriangular (double min, double mode, double max)=0
- virtual double sampleDiscrete (double value, double acumProb,...)=0
- virtual void setRNGparameters (RNG_Parameters ∗param)=0
- virtual RNG_Parameters ∗ getRNGparameters () const =0

5.73.1 Detailed Description

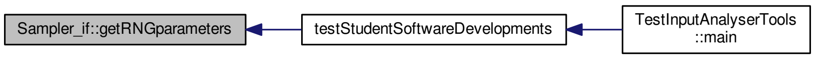
Interface that describes the methods to be implemented by classes that generate random values that follow a specific probability distribution.

5.73.2 Member Function Documentation

5.73.2.1 virtual RNG_Parameters∗ Sampler_if::getRNGparameters ( ) const [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

Here is the caller graph for this function:


Sampler_if::getRNGparameters
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

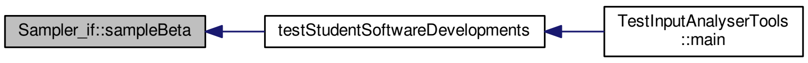
5.73.2.2 virtual double Sampler_if::random ( ) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

5.73.2.3 virtual double Sampler_if::sampleBeta (double *alpha,* double *beta,* double *infLimit,* double *supLimit*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

Here is the caller graph for this function:


Sampler_if::sampleBeta
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

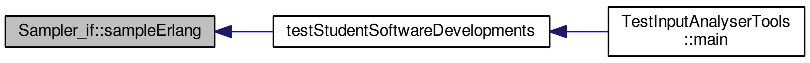
5.73.2.4 virtual double Sampler_if::sampleDiscrete (double *value,* double *acumProb,* ...) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

5.73.2.5 virtual double Sampler_if::sampleErlang (double *mean,* int *M*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

Here is the caller graph for this function:


Sampler_if::sampleErlang
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

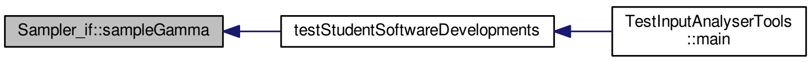
5.73.2.6 virtual double Sampler_if::sampleExponential (double *mean*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

5.73.2.7 virtual double Sampler_if::sampleGamma (double *mean*, double *alpha*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

Here is the caller graph for this function:


Sampler_if::sampleGamma
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

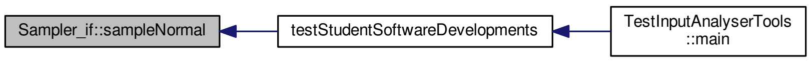
5.73.2.8 virtual double Sampler_if::sampleLogNormal (double *mean*, double *stddev*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

5.73.2.9 virtual double Sampler_if::sampleNormal (double *mean*, double *stddev*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

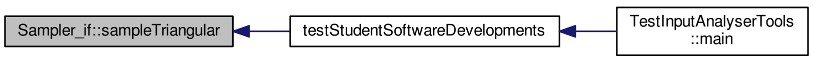
Here is the caller graph for this function:


Sampler_if::sampleNormal
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.73.2.10 virtual double Sampler_if::sampleTriangular (double *min*, double *mode*, double *max*) [pure virtual]

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

Here is the caller graph for this function:


Sampler_if::sampleTriangular
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

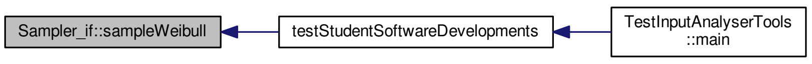
5.73.2.11 virtual double Sampler_if::sampleUniform (double *min*, double *max*) `[pure virtual]`

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

5.73.2.12 virtual double Sampler_if::sampleWeibull (double *alpha*, double *scale*) `[pure virtual]`

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

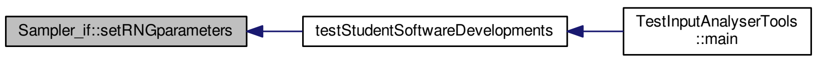
Here is the caller graph for this function:


Sampler_if::sampleWeibull
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.73.2.13 virtual void Sampler_if::setRNGparameters (**RNG_Parameters** ∗ *param*) `[pure virtual]`

Implemented in SamplerDefaultImpl1, and SamplerDummyImpl.

Here is the caller graph for this function:


Sampler_if::setRNGparameters
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

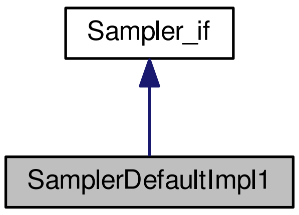
The documentation for this class was generated from the following file:

- Sampler_if.h

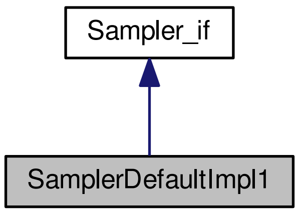
5.74 SamplerDefaultImpl1 Class Reference

`#include <SamplerDefaultImpl1.h>`

Inheritance diagram for SamplerDefaultImpl1:



Collaboration diagram for SamplerDefaultImpl1:



Classes

- struct DefaultImpl1RNG_Parameters

Public Member Functions

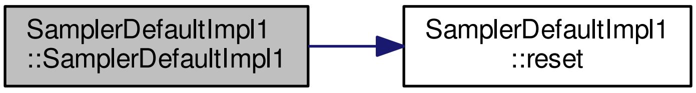
- SamplerDefaultImpl1 ()
- SamplerDefaultImpl1 (const SamplerDefaultImpl1 &orig)
- virtual ∼SamplerDefaultImpl1 ()
- virtual double random ()
- virtual double sampleUniform (double min, double max)
- virtual double sampleExponential (double mean)
- virtual double sampleErlang (double mean, int M)
- virtual double sampleNormal (double mean, double stddev)
- virtual double sampleGamma (double mean, double alpha)
- virtual double sampleBeta (double alpha, double beta, double infLimit, double supLimit)
- virtual double sampleWeibull (double alpha, double scale)
- virtual double sampleLogNormal (double mean, double stddev)
- virtual double sampleTriangular (double min, double mode, double max)
- virtual double sampleDiscrete (double value, double acumProb,...)
- void reset ()

    reinitialize seed and other parameters so (pseudo) random number sequence will be generated again.
- virtual void setRNGparameters (RNG_Parameters ∗param)
- virtual RNG_Parameters ∗ getRNGparameters () const

5.74.1 Constructor & Destructor Documentation

5.74.1.1 SamplerDefaultImpl1::SamplerDefaultImpl1 ( )

Here is the call graph for this function:



5.74.1.2 SamplerDefaultImpl1::SamplerDefaultImpl1 (const **SamplerDefaultImpl1** & *orig*)

5.74.1.3 SamplerDefaultImpl1::∼SamplerDefaultImpl1 ( ) `[virtual]`

5.74.2 Member Function Documentation

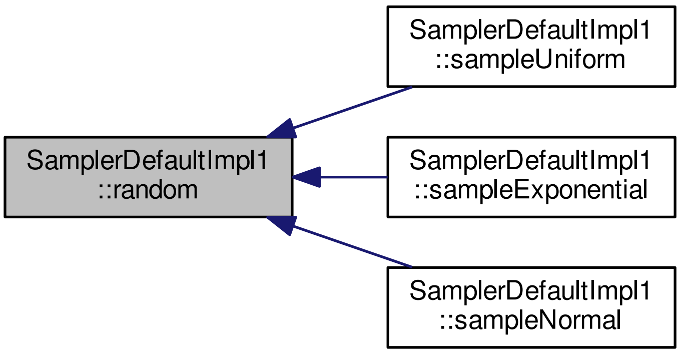
5.74.2.1 **Sampler_if::RNG_Parameters** ∗ SamplerDefaultImpl1::getRNGparameters ( ) const `[virtual]`

Implements Sampler_if.

5.74.2.2 double SamplerDefaultImpl1::random ( ) `[virtual]`

Implements Sampler_if.

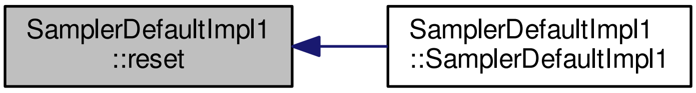
Here is the caller graph for this function:



5.74.2.3 void SamplerDefaultImpl1::reset ( )

reinitialize seed and other parameters so (pseudo) random number sequence will be generated again.

Here is the caller graph for this function:


SamplerDefaultImpl1 ::reset
SamplerDefaultImpl1 ::SamplerDefaultImpl1

5.74.2.4 double SamplerDefaultImpl1::sampleBeta (double *alpha*, double *beta*, double *infLimit*, double *supLimit*) [virtual]

Implements Sampler_if.

5.74.2.5 double SamplerDefaultImpl1::sampleDiscrete (double *value*, double *acumProb*, ...) [virtual]

Implements Sampler_if.

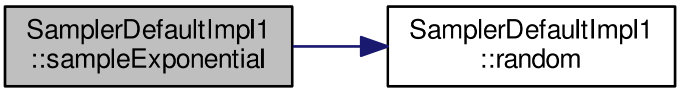
5.74.2.6 double SamplerDefaultImpl1::sampleErlang (double *mean*, int *M*) [virtual]

Implements Sampler_if.

5.74.2.7 double SamplerDefaultImpl1::sampleExponential (double *mean*) [virtual]

Implements Sampler_if.

Here is the call graph for this function:


SamplerDefaultImpl1 ::sampleExponential
SamplerDefaultImpl1 ::random

5.74.2.8 double SamplerDefaultImpl1::sampleGamma (double *mean*, double *alpha*) [virtual]

Implements Sampler_if.

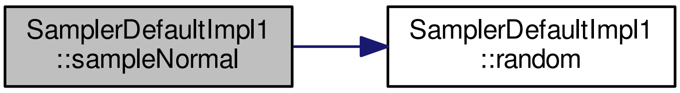
5.74.2.9 double SamplerDefaultImpl1::sampleLogNormal (double *mean*, double *stddev*) [virtual]

Implements Sampler_if.

5.74.2.10 double SamplerDefaultImpl1::sampleNormal (double *mean*, double *stddev*) [virtual]

Implements Sampler_if.

Here is the call graph for this function:


SamplerDefaultImpl1 ::sampleNormal
SamplerDefaultImpl1 ::random

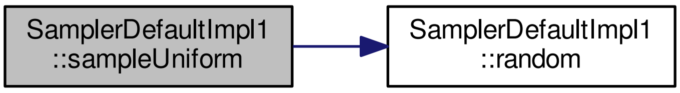
5.74.2.11 double SamplerDefaultImpl1::sampleTriangular (double *min*, double *mode*, double *max*) [virtual]

Implements Sampler_if.

5.74.2.12 double SamplerDefaultImpl1::sampleUniform (double *min*, double *max*) [virtual]

Implements Sampler_if.

Here is the call graph for this function:


SamplerDefaultImpl1 ::sampleUniform
SamplerDefaultImpl1 ::random

5.74.2.13 double SamplerDefaultImpl1::sampleWeibull (double *alpha*, double *scale*) [virtual]

Implements Sampler_if.

5.74.2.14 void SamplerDefaultImpl1::setRNGparameters (Sampler_if::RNG_Parameters ∗ *param*) [virtual]

Implements Sampler_if.

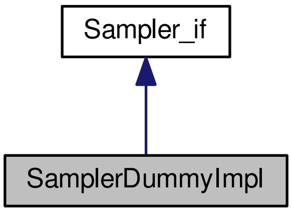
The documentation for this class was generated from the following files:

- SamplerDefaultImpl1.h
- SamplerDefaultImpl1.cpp

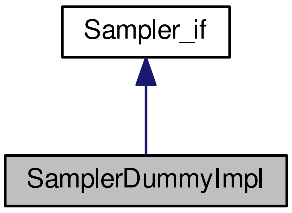
5.75 SamplerDummyImpl Class Reference

```
#include <SamplerDummyImpl.h>
```

Inheritance diagram for SamplerDummyImpl:



Collaboration diagram for SamplerDummyImpl:



Classes

- struct MyRNG_Parameters

Public Member Functions

- SamplerDummyImpl ()
- SamplerDummyImpl (const SamplerDummyImpl &orig)
- ∼SamplerDummyImpl ()
- double random ()
- double sampleUniform (double min, double max)
- double sampleExponential (double mean)
- double sampleErlang (double mean, int M)
- double sampleNormal (double mean, double stddev)
- double sampleGamma (double mean, double alpha)
- double sampleBeta (double alpha, double beta, double infLimit, double supLimit)
- double sampleWeibull (double alpha, double scale)
- double sampleLogNormal (double mean, double stddev)
- double sampleTriangular (double min, double mode, double max)
- double sampleDiscrete (double value, double acumProb,...)
- void setRNGparameters (RNG_Parameters ∗param)
- RNG_Parameters ∗ getRNGparameters () const

5.75.1 Constructor & Destructor Documentation

5.75.1.1 SamplerDummyImpl::SamplerDummyImpl ( )

5.75.1.2 SamplerDummyImpl::SamplerDummyImpl (const SamplerDummyImpl & *orig*)

5.75.1.3 SamplerDummyImpl::∼SamplerDummyImpl ( )

5.75.2 Member Function Documentation

5.75.2.1 Sampler_if::RNG_Parameters ∗ SamplerDummyImpl::getRNGparameters ( ) const [virtual]

Implements Sampler_if.

5.75.2.2 double SamplerDummyImpl::random ( ) [virtual]

Implements Sampler_if.

5.75.2.3 double SamplerDummyImpl::sampleBeta (double *alpha,* double *beta,* double *infLimit,* double *supLimit*) [virtual]

Implements Sampler_if.

5.75.2.4 double SamplerDummyImpl::sampleDiscrete (double *value*, double *acumProb*, ...) [virtual]

Implements Sampler_if.

5.75.2.5 double SamplerDummyImpl::sampleErlang (double *mean*, int *M*) [virtual]

Implements Sampler_if.

5.75.2.6 double SamplerDummyImpl::sampleExponential (double *mean*) [virtual]

Implements Sampler_if.

5.75.2.7 double SamplerDummyImpl::sampleGamma (double *mean*, double *alpha*) [virtual]

Implements Sampler_if.

5.75.2.8 double SamplerDummyImpl::sampleLogNormal (double *mean*, double *stddev*) [virtual]

Implements Sampler_if.

5.75.2.9 double SamplerDummyImpl::sampleNormal (double *mean*, double *stddev*) [virtual]

Implements Sampler_if.

5.75.2.10 double SamplerDummyImpl::sampleTriangular (double *min*, double *mode*, double *max*) [virtual]

Implements Sampler_if.

5.75.2.11 double SamplerDummyImpl::sampleUniform (double *min*, double *max*) [virtual]

Implements Sampler_if.

5.75.2.12 double SamplerDummyImpl::sampleWeibull (double *alpha*, double *scale*) [virtual]

Implements Sampler_if.

5.75.2.13 void SamplerDummyImpl::setRNGparameters (Sampler_if::RNG_Parameters ∗ *param*) [virtual]

Implements Sampler_if.

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

- SamplerDummyImpl.h
- SamplerDummyImpl.cpp

5.76 ScenarioExperiment_if Class Reference

```
#include <ScenarioExperiment_if.h>
```

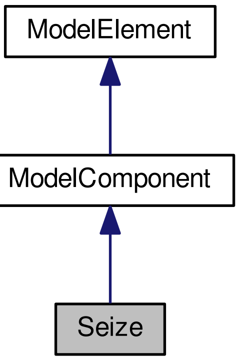
The documentation for this class was generated from the following file:

- ScenarioExperiment_if.h

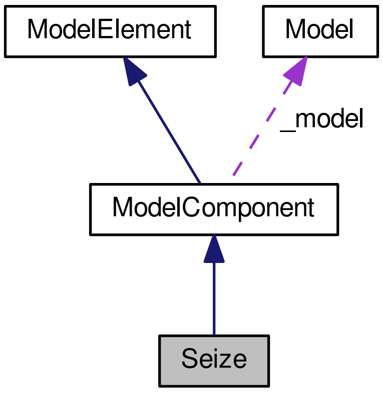
5.77 Seize Class Reference

```
#include <Seize.h>
```

Inheritance diagram for Seize:



Collaboration diagram for Seize:



Public Member Functions

- Seize (Model ∗model)
- Seize (const Seize &orig)
- virtual ∼Seize ()
- virtual std::string show ()
- void setLastMemberSeized (unsigned int _lastMemberSeized)
- unsigned int getLastMemberSeized () const
- void setSaveAttribute (std::string _saveAttribute)
- std::string getSaveAttribute () const
- void setRule (Resource::ResourceRule _rule)
- Resource::ResourceRule getRule () const
- void setQuantity (std::string _quantity)
- std::string getQuantity () const
- void setResourceType (Resource::ResourceType _resourceType)
- Resource::ResourceType getResourceType () const
- void setPriority (unsigned short _priority)
- unsigned short getPriority () const
- void setAllocationType (unsigned int _allocationType)
- unsigned int getAllocationType () const
- void setResourceName (std::string _resourceName) throw ()
- std::string getResourceName () const
- void setQueueName (std::string queueName) throw ()
- std::string getQueueName () const
- void setResource (Resource ∗resource)
- Resource ∗ getResource () const
- void setQueue (Queue ∗queue)
- Queue ∗ getQueue () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelComponent ∗ LoadInstance (Model ∗model, std::map< std::string, std::string > ∗fields)

Protected Member Functions

- virtual void _execute (Entity ∗entity)
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

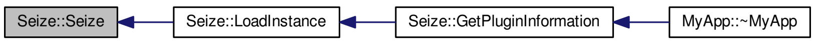
5.77.1 Detailed Description

Seize tries to allocate a certain amount of a resource

5.77.2 Constructor & Destructor Documentation

5.77.2.1 Seize::Seize (Model ∗ *model*)

Here is the caller graph for this function:



5.77.2.2 Seize::Seize (const Seize & *orig*)

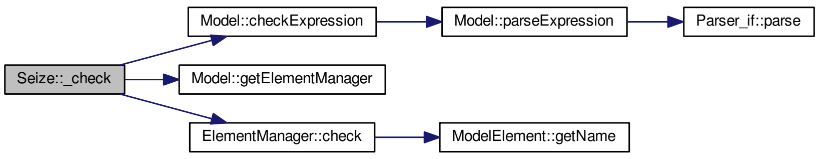
5.77.2.3 Seize::∼Seize ( ) [virtual]

5.77.3 Member Function Documentation

5.77.3.1 bool Seize::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

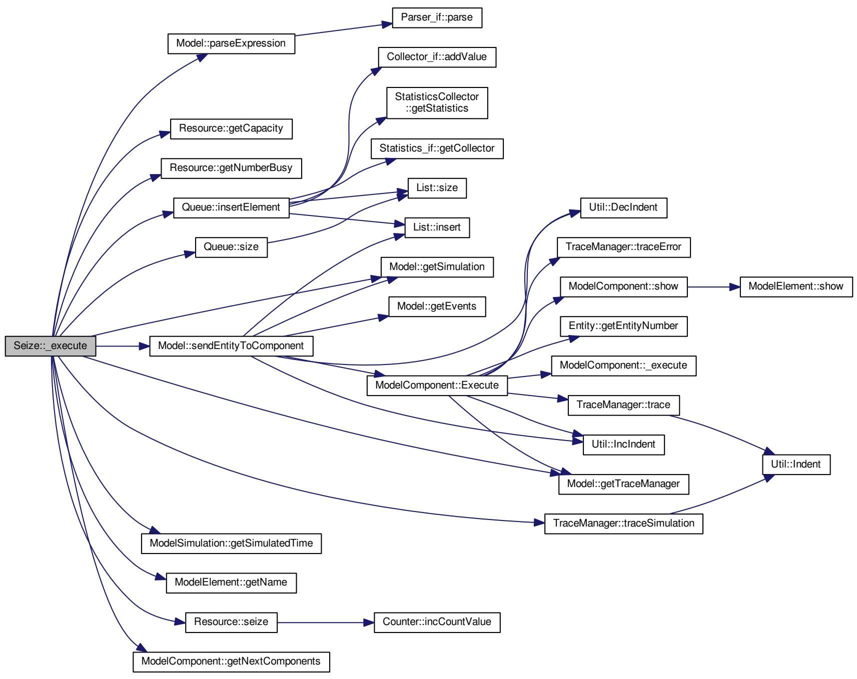
Here is the call graph for this function:



5.77.3.2 void Seize::_execute (Entity ∗ *entity*) [protected], [virtual]

Implements ModelComponent.

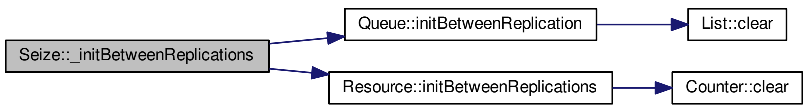
Here is the call graph for this function:



5.77.3.3 void Seize::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

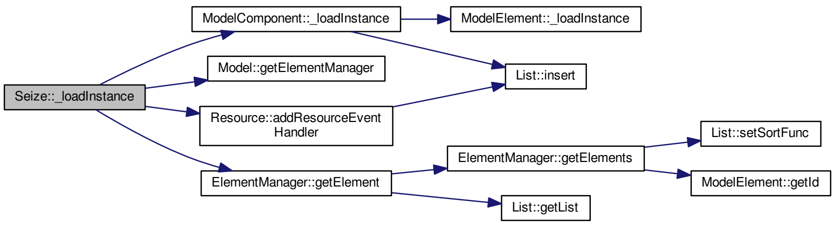
Here is the call graph for this function:


Seize::_initBetweenReplications
Queue::initBetweenReplication
List::clear
Resource::initBetweenReplications
Counter::clear

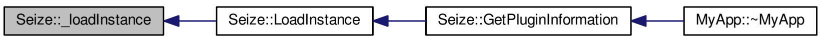
5.77.3.4 bool Seize::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:


ModelComponent::_loadInstance
ModelElement::_loadInstance
Model::getElementManager
List::insert
Seize::_loadInstance
Resource::addResourceEvent Handler
List::setSortFunc
ElementManager::getElements
ElementManager::getElement
ModelElement::getId
List::getList

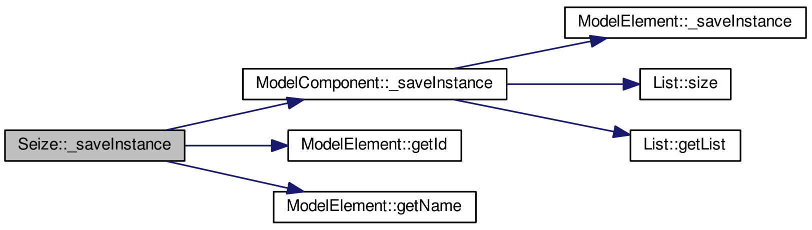
Here is the caller graph for this function:



5.77.3.5 std::map< std::string, std::string > * Seize::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:

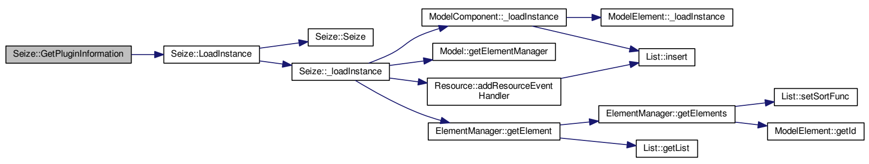


5.77.3.6 unsigned int Seize::getAllocationType ( ) const

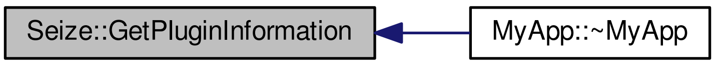
5.77.3.7 unsigned int Seize::getLastMemberSeized ( ) const

5.77.3.8 PluginInformation ∗ Seize::GetPluginInformation ( ) [static]

Here is the call graph for this function:


Seize::GetPluginInformation
Seize::LoadInstance
Seize::Seize
Seize::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
Model::getElementManager
List::insert
Resource::addResourceEvent Handler
ElementManager::getElements
List::setSortFunc
ModelElement::getId
ElementManager::getElement
List::getList

Here is the caller graph for this function:



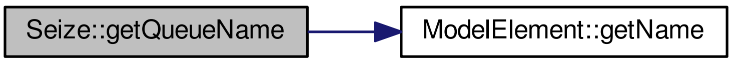
5.77.3.9 unsigned short Seize::getPriority ( ) const

5.77.3.10 std::string Seize::getQuantity ( ) const

5.77.3.11 Queue ∗ Seize::getQueue ( ) const

5.77.3.12 std::string Seize::getQueueName ( ) const

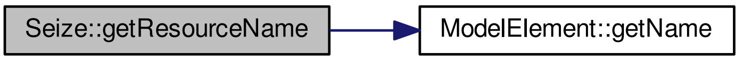
Here is the call graph for this function:



5.77.3.13 Resource ∗ Seize::getResource ( ) const

5.77.3.14 std::string Seize::getResourceName ( ) const

Here is the call graph for this function:



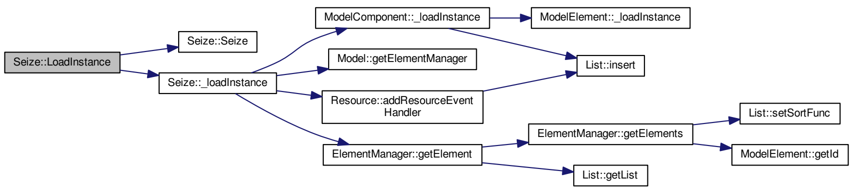
5.77.3.15 Resource::ResourceType Seize::getResourceType ( ) const

5.77.3.16 Resource::ResourceRule Seize::getRule ( ) const

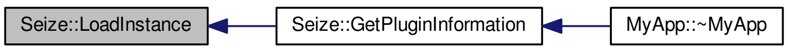
5.77.3.17 std::string Seize::getSaveAttribute ( ) const

5.77.3.18 ModelComponent ∗ Seize::LoadInstance (Model ∗ *model,* std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:


Seize::LoadInstance
Seize::Seize
Seize::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
Model::getElementManager
List::insert
Resource::addResourceEvent Handler
ElementManager::getElements
List::setSortFunc
ModelElement::getId
ElementManager::getElement
List::getList

Here is the caller graph for this function:



5.77.3.19 void Seize::setAllocationType (unsigned int *_allocationType*)

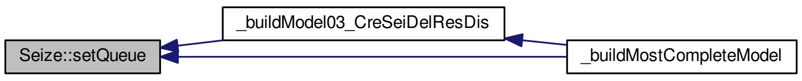
5.77.3.20 void Seize::setLastMemberSeized (unsigned int *_lastMemberSeized*)

5.77.3.21 void Seize::setPriority (unsigned short *_priority*)

5.77.3.22 void Seize::setQuantity (std::string *_quantity*)

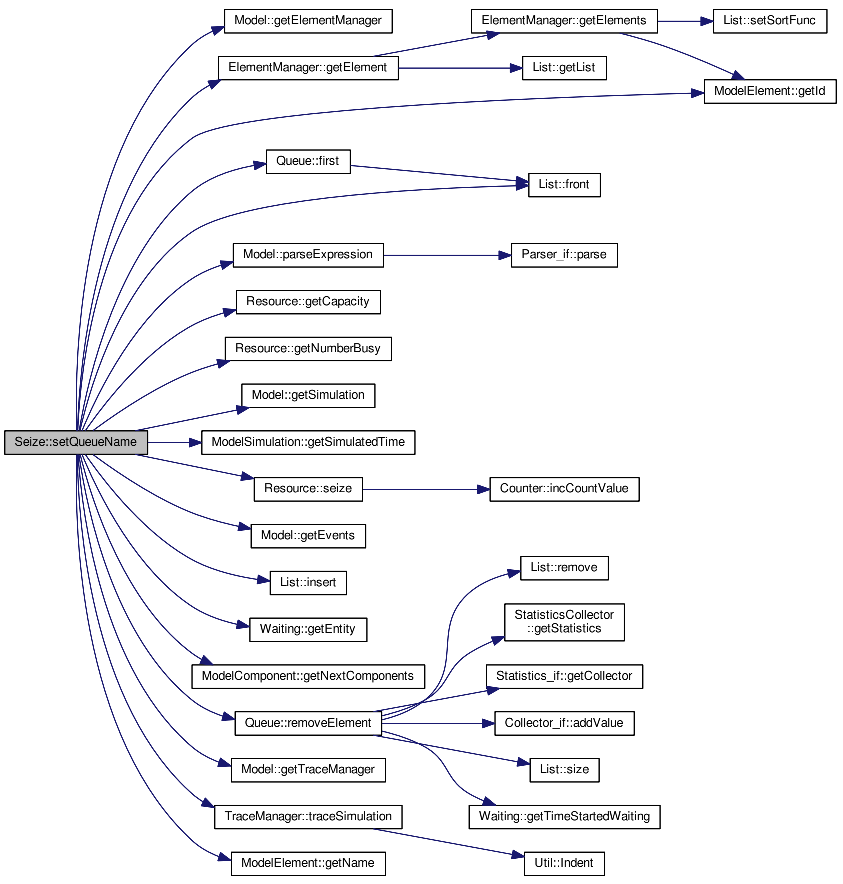
5.77.3.23 void Seize::setQueue (Queue ∗ *queue*)

Here is the caller graph for this function:



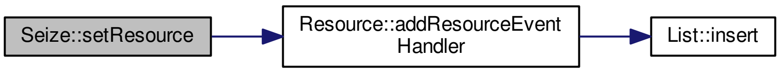
5.77.3.24 void Seize::setQueueName (std::string *queueName*) throw)

Here is the call graph for this function:

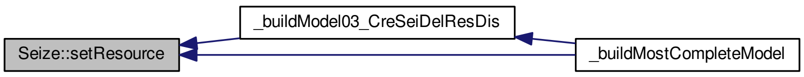


5.77.3.25 void Seize::setResource (Resource ∗ *resource*)

Here is the call graph for this function:

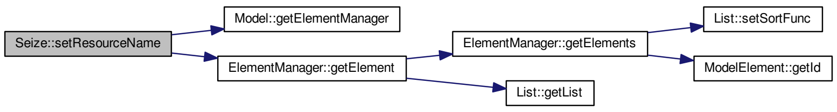


Here is the caller graph for this function:



5.77.3.26 void Seize::setResourceName (std::string *_resourceName*) throw)

Here is the call graph for this function:



5.77.3.27 void Seize::setResourceType (Resource::ResourceType *_resourceType*)

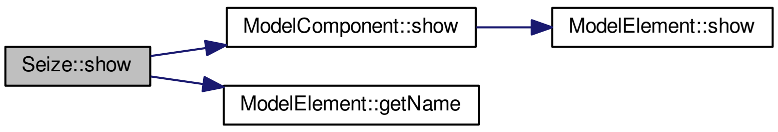
5.77.3.28 void Seize::setRule (Resource::ResourceRule *_rule*)

5.77.3.29 void Seize::setSaveAttribute (std::string *_saveAttribute*)

5.77.3.30 std::string Seize::show ( ) [virtual]

Reimplemented from ModelComponent.

Here is the call graph for this function:



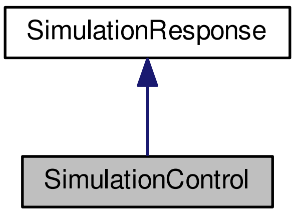
The documentation for this class was generated from the following files:

- Seize.h
- Seize.cpp

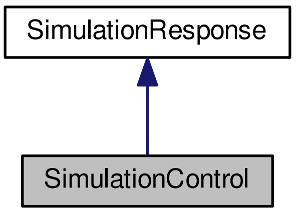
5.78 SimulationControl Class Reference

```
#include <SimulationControl.h>
```

Inheritance diagram for SimulationControl:



Collaboration diagram for SimulationControl:



Public Member Functions

- SimulationControl (std::string type, std::string name, GetterMember getterMember, SetterMember setter↩ Member)
- SimulationControl (const SimulationControl &orig)
- virtual ∼SimulationControl ()
- void setValue (double value)

Additional Inherited Members

5.78.1 Detailed Description

Represents any possible parameter or control for a simulation. Any element or event the model can declare one of its own attribute as a simulation control. It just have to create a SimulationControl object, passing the access to the methods that gets and sets the control value and including this SimulationControl in the corresponding list of the model

5.78.2 Constructor & Destructor Documentation

5.78.2.1 SimulationControl::SimulationControl (std::string *type,* std::string *name,* GetterMember *getterMember,* SetterMember *setterMember*)

5.78.2.2 SimulationControl::SimulationControl (const SimulationControl & *orig*)

5.78.2.3 SimulationControl::∼SimulationControl ( ) `[virtual]`

5.78.3 Member Function Documentation

5.78.3.1 void SimulationControl::setValue (double *value*)

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

- SimulationControl.h
- SimulationControl.cpp

5.79 SimulationEvent Class Reference

```
#include <OnEventManager.h>
```

Public Member Functions

- SimulationEvent (unsigned int replicationNumber, Event ∗event)
- unsigned int getReplicationNumber () const
- Event ∗ getEventProcessed () const

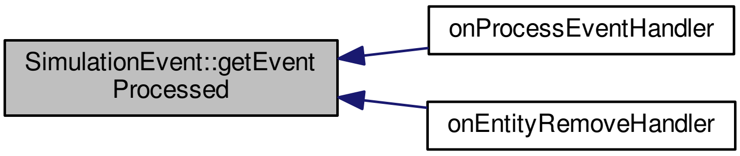
5.79.1 Constructor & Destructor Documentation

5.79.1.1 SimulationEvent::SimulationEvent (unsigned int *replicationNumber*, Event ∗ *event*) [inline]

5.79.2 Member Function Documentation

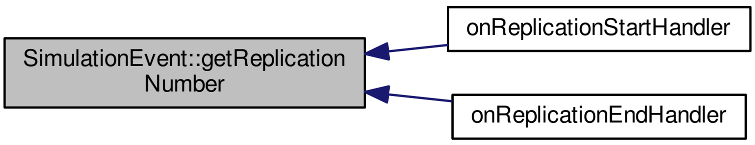
5.79.2.1 Event∗ SimulationEvent::getEventProcessed ( ) const [inline]

Here is the caller graph for this function:



5.79.2.2 unsigned int SimulationEvent::getReplicationNumber ( ) const [inline]

Here is the caller graph for this function:


SimulationEvent::getReplication Number
onReplicationStartHandler
onReplicationEndHandler

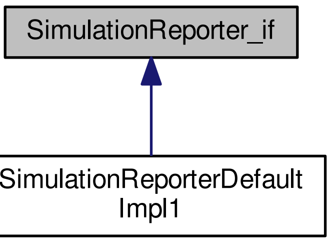
The documentation for this class was generated from the following file:

- OnEventManager.h

5.80 SimulationReporter_if Class Reference

`#include <SimulationReporter_if.h>`

Inheritance diagram for SimulationReporter_if:



Public Member Functions

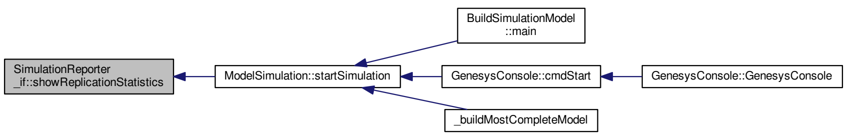
- virtual void showReplicationStatistics ()=0
- virtual void showSimulationStatistics ()=0

5.80.1 Member Function Documentation

5.80.1.1 virtual void SimulationReporter_if::showReplicationStatistics ( ) [pure virtual]

Implemented in SimulationReporterDefaultImpl1.

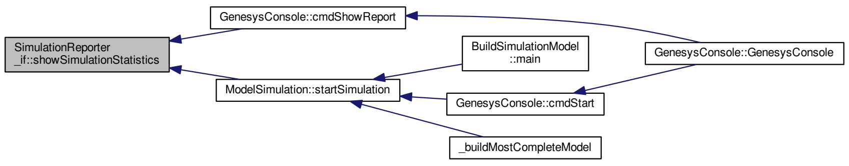
Here is the caller graph for this function:


SimulationReporter
_if::showReplicationStatistics
ModelSimulation::startSimulation
BuildSimulationModel
::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.80.1.2 virtual void SimulationReporter_if::showSimulationStatistics ( ) [pure virtual]

Implemented in SimulationReporterDefaultImpl1.

Here is the caller graph for this function:


SimulationReporter_if::showSimulationStatistics
GenesysConsole::cmdShowReport
ModelSimulation::startSimulation
BuildSimulationModel::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

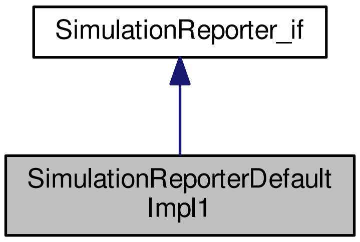
The documentation for this class was generated from the following file:

- SimulationReporter_if.h

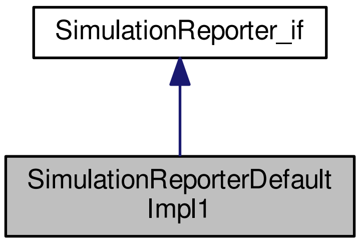
5.81 SimulationReporterDefaultImpl1 Class Reference

```
#include <SimulationReporterDefaultImpl1.h>
```

Inheritance diagram for SimulationReporterDefaultImpl1:



Collaboration diagram for SimulationReporterDefaultImpl1:



Public Member Functions

- SimulationReporterDefaultImpl1 (ModelSimulation ∗simulation, Model ∗model, List< ModelElement ∗ > ∗statsCountersSimulation)
- SimulationReporterDefaultImpl1 (const SimulationReporterDefaultImpl1 &orig)
- virtual ∼SimulationReporterDefaultImpl1 ()
- virtual void showReplicationStatistics ()
- virtual void showSimulationStatistics ()

5.81.1 Detailed Description

Class that implements SimulationReporter_if interface and is responsible for building and showing replication and simulation reports

5.81.2 Constructor & Destructor Documentation

5.81.2.1 SimulationReporterDefaultImpl1::SimulationReporterDefaultImpl1 (**ModelSimulation** ∗ *simulation,* **Model** ∗ *model,* **List**< **ModelElement** ∗ > ∗ *statsCountersSimulation*)

5.81.2.2 SimulationReporterDefaultImpl1::SimulationReporterDefaultImpl1 (const **SimulationReporterDefaultImpl1** & *orig*)

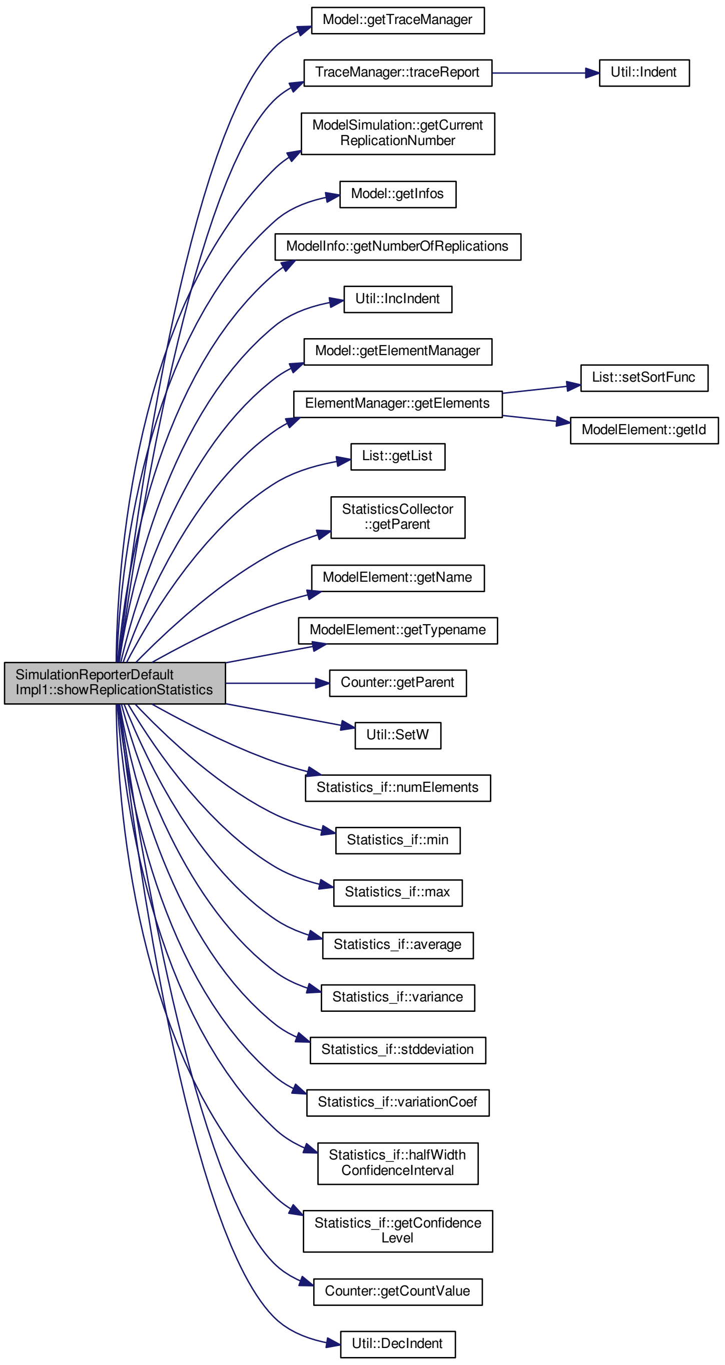
5.81.2.3 SimulationReporterDefaultImpl1::∼SimulationReporterDefaultImpl1 ( ) `[virtual]`

5.81.3 Member Function Documentation

5.81.3.1 void SimulationReporterDefaultImpl1::showReplicationStatistics ( ) `[virtual]`

Implements SimulationReporter_if.

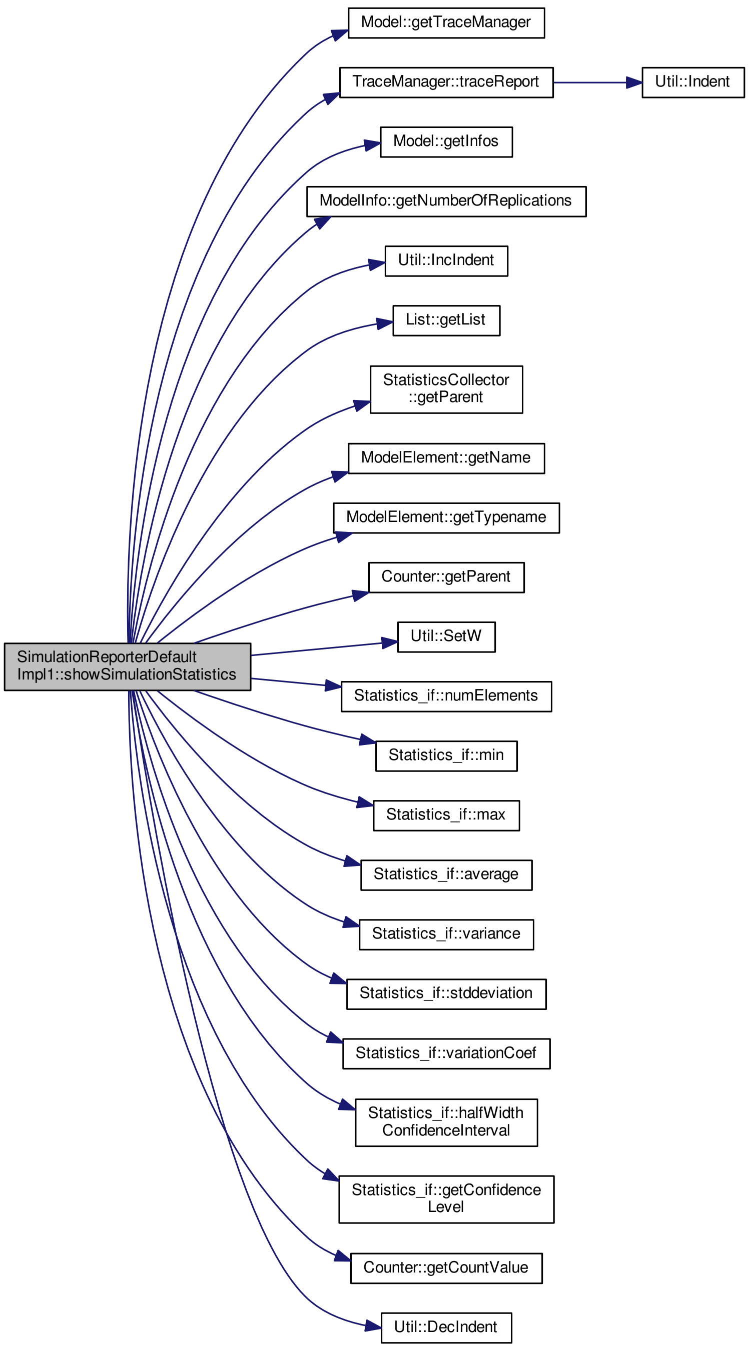
Here is the call graph for this function:


SimulationReporterDefault Impl1::showReplicationStatistics
Model::getTraceManager
TraceManager::traceReport
Util::Indent
ModelSimulation::getCurrent ReplicationNumber
Model::getInfos
ModelInfo::getNumberOfReplications
Util::IncIndent
Model::getElementManager
ElementManager::getElements
List::setSortFunc
ModelElement::getId
List::getList
StatisticsCollector ::getParent
ModelElement::getName
ModelElement::getTypename
Counter::getParent
Util::SetW
Statistics_if::numElements
Statistics_if::min
Statistics_if::max
Statistics_if::average
Statistics_if::variance
Statistics_if::stddeviation
Statistics_if::variationCoef
Statistics_if::halfWidth ConfidenceInterval
Statistics_if::getConfidence Level
Counter::getCountValue
Util::DecIndent

5.81.3.2 void SimulationReporterDefaultImpl1::showSimulationStatistics ( ) [virtual]

Implements SimulationReporter_if.

Here is the call graph for this function:



Model::getTraceManager
TraceManager::traceReport
Util::Indent
Model::getInfos
ModelInfo::getNumberOfReplications
Util::IncIndent
List::getList
StatisticsCollector ::getParent
ModelElement::getName
ModelElement::getTypename
Counter::getParent
Util::SetW
SimulationReporterDefault Impl1::showSimulationStatistics
Statistics_if::numElements
Statistics_if::min
Statistics_if::max
Statistics_if::average
Statistics_if::variance
Statistics_if::stddeviation
Statistics_if::variationCoef
Statistics_if::halfWidth ConfidenceInterval
Statistics_if::getConfidence Level
Counter::getCountValue
Util::DecIndent

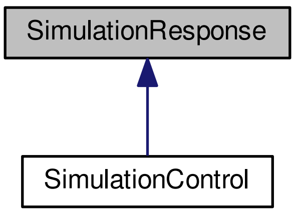
The documentation for this class was generated from the following files:

- SimulationReporterDefaultImpl1.h
- SimulationReporterDefaultImpl1.cpp

5.82 SimulationResponse Class Reference

```
#include <SimulationResponse.h>
```

Inheritance diagram for SimulationResponse:



Public Member Functions

- SimulationResponse (std::string type, std::string name, GetterMember getterMember)
- SimulationResponse (const SimulationResponse &orig)
- virtual ∼SimulationResponse ()
- double getValue ()
- std::string getName () const
- std::string getType () const

Protected Attributes

- std::string _type
- std::string _name
- GetterMember _getterMemberFunction

5.82.1 Detailed Description

Represents any possible response of a simulation. Any element or event the model can declare one of its own attribute as a simulation response. It just have to create a SimulationResponse object, passing the access to the method that gets the response value and including this SimulationResponse in the corresponding list of the model

5.82.2 Constructor & Destructor Documentation

5.82.2.1 SimulationResponse::SimulationResponse (std::string *type*, std::string *name*, GetterMember *getterMember*)

5.82.2.2 SimulationResponse::SimulationResponse (const SimulationResponse & *orig*)

5.82.2.3 SimulationResponse::∼SimulationResponse ( ) `[virtual]`

5.82.3 Member Function Documentation

5.82.3.1 std::string SimulationResponse::getName ( ) const

5.82.3.2 std::string SimulationResponse::getType ( ) const

5.82.3.3 double SimulationResponse::getValue ( )

5.82.4 Member Data Documentation

5.82.4.1 GetterMember SimulationResponse::_getterMemberFunction `[protected]`

5.82.4.2 std::string SimulationResponse::_name `[protected]`

5.82.4.3 std::string SimulationResponse::_type `[protected]`

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

- SimulationResponse.h
- SimulationResponse.cpp

5.83 SimulationScenario Class Reference

`#include <SimulationScenario.h>`

Public Member Functions

- SimulationScenario ()
- SimulationScenario (const SimulationScenario &orig)
- virtual ∼SimulationScenario ()
- void setName (std::string _name)
- std::string getName () const
- std::list< double > $*$ getResponseValues () const
- std::list< double > $*$ getControlValues () const
- void setModelFilename (std::string _modelFilename)
- std::string getModelFilename () const
- double getResponseValue (SimulationResponse $*$value)
- double getControlValue (SimulationControl $*$control)
- void setControlValue (SimulationControl $*$control, double value)

5.83.1 Detailed Description

Represents a scenario where a specific model (defined my ModelFilename) will be simulated. To each scenario will be associated a set of SimulationControl and SimulationResponse, and their values are set to the scenario by the ProcessAnalyser.

5.83.2 Constructor & Destructor Documentation

5.83.2.1 SimulationScenario::SimulationScenario ( )

5.83.2.2 SimulationScenario::SimulationScenario (const **SimulationScenario** & *orig*)

5.83.2.3 SimulationScenario::∼SimulationScenario ( ) `[virtual]`

5.83.3 Member Function Documentation

5.83.3.1 double SimulationScenario::getControlValue (**SimulationControl** ∗ *control*)

5.83.3.2 std::list< double > ∗ SimulationScenario::getControlValues ( ) const

5.83.3.3 std::string SimulationScenario::getModelFilename ( ) const

5.83.3.4 std::string SimulationScenario::getName ( ) const

5.83.3.5 double SimulationScenario::getResponseValue (**SimulationResponse** ∗ *value*)

5.83.3.6 std::list< double > ∗ SimulationScenario::getResponseValues ( ) const

5.83.3.7 void SimulationScenario::setControlValue (**SimulationControl** ∗ *control,* double *value*)

5.83.3.8 void SimulationScenario::setModelFilename (std::string *_modelFilename*)

5.83.3.9 void SimulationScenario::setName (std::string *_name*)

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

- SimulationScenario.h
- SimulationScenario.cpp

5.84 Simulator Class Reference

```
#include <Simulator.h>
```

Public Member Functions

- Simulator ()
- Simulator (const Simulator &orig)
- virtual ∼Simulator ()
- std::string getVersion () const
- std::string getName () const
- LicenceManager ∗ getLicenceManager () const
- PluginManager ∗ getPluginManager () const
- ModelManager ∗ getModelManager () const
- ToolManager ∗ getToolManager () const
- TraceManager ∗ getTraceManager () const

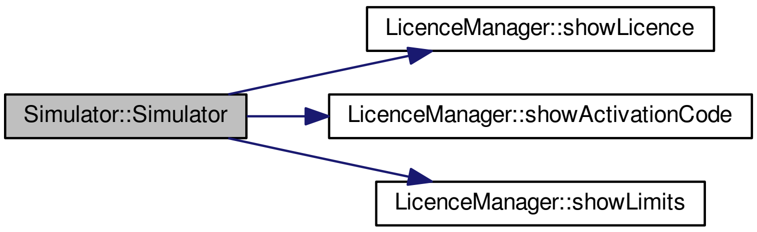
5.84.1 Detailed Description

The main class of the ReGenesys KERNEL simulation. It gives access to simulation models and tools.

5.84.2 Constructor & Destructor Documentation

5.84.2.1 Simulator::Simulator ( )

Here is the call graph for this function:


Simulator::Simulator
LicenceManager::showLicence
LicenceManager::showActivationCode
LicenceManager::showLimits

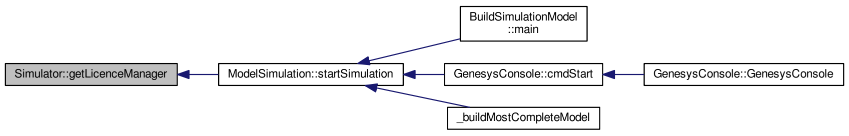
5.84.2.2 Simulator::Simulator (const Simulator & *orig*)

5.84.2.3 Simulator::∼Simulator ( ) `[virtual]`

5.84.3 Member Function Documentation

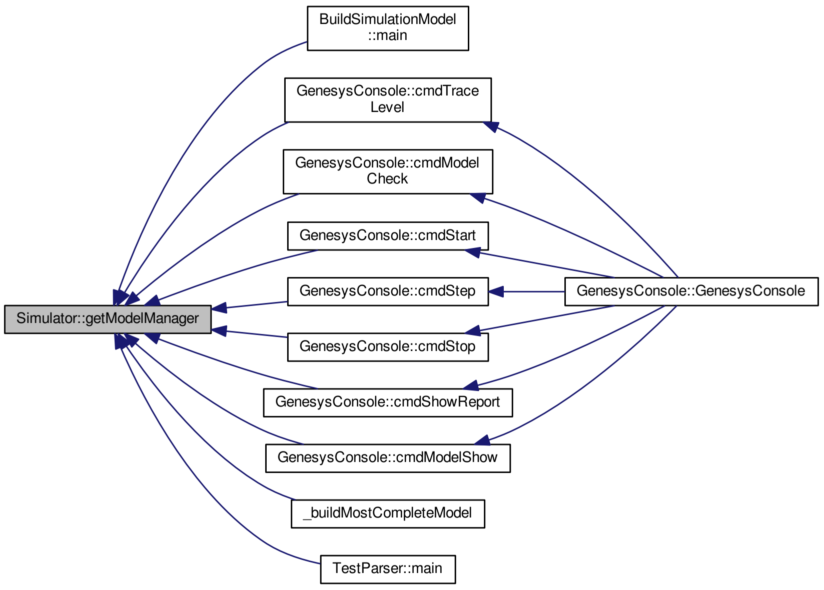
5.84.3.1 LicenceManager ∗ Simulator::getLicenceManager ( ) const

Here is the caller graph for this function:


BuildSimulationModel ::main
Simulator::getLicenceManager
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.84.3.2 ModelManager ∗ Simulator::getModelManager ( ) const

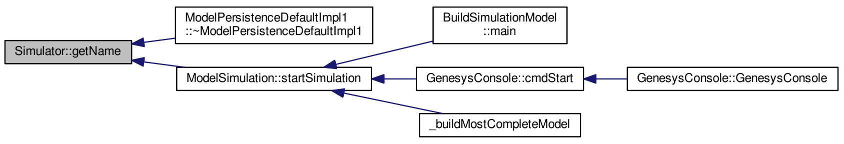
Here is the caller graph for this function:



Simulator::getModelManager
BuildSimulationModel ::main
GenesysConsole::cmdTrace Level
GenesysConsole::cmdModel Check
GenesysConsole::cmdStart
GenesysConsole::cmdStep
GenesysConsole::cmdStop
GenesysConsole::cmdShowReport
GenesysConsole::cmdModelShow
_buildMostCompleteModel
TestParser::main
GenesysConsole::GenesysConsole

5.84.3.3 std::string Simulator::getName ( ) const

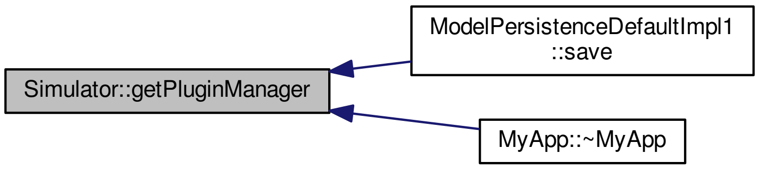
Here is the caller graph for this function:



Simulator::getName
ModelPersistenceDefaultImpl1 ::~ModelPersistenceDefaultImpl1
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

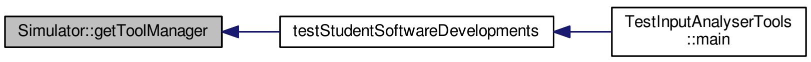
5.84.3.4 PluginManager ∗ Simulator::getPluginManager ( ) const

Here is the caller graph for this function:


Simulator::getPluginManager
ModelPersistenceDefaultImpl1 ::save
MyApp::~MyApp

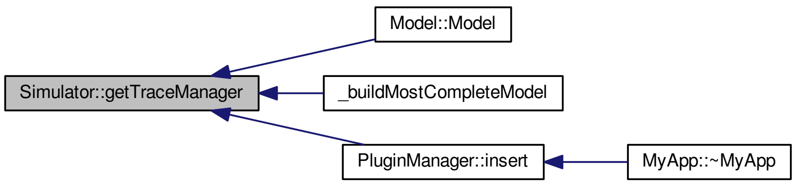
5.84.3.5 ToolManager ∗ Simulator::getToolManager ( ) const

Here is the caller graph for this function:


Simulator::getToolManager
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

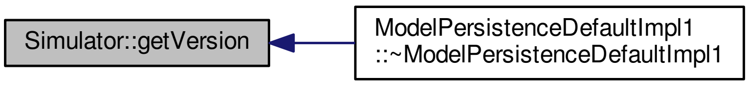
5.84.3.6 TraceManager ∗ Simulator::getTraceManager ( ) const

Here is the caller graph for this function:



5.84.3.7 std::string Simulator::getVersion ( ) const

Here is the caller graph for this function:



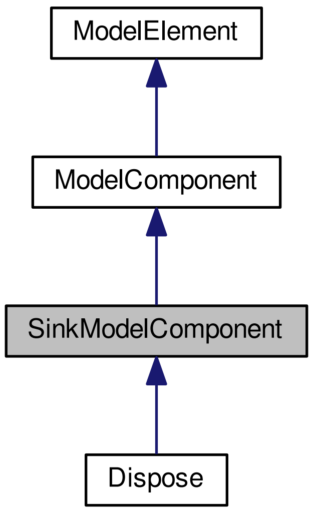
The documentation for this class was generated from the following files:

- Simulator.h
- Simulator.cpp

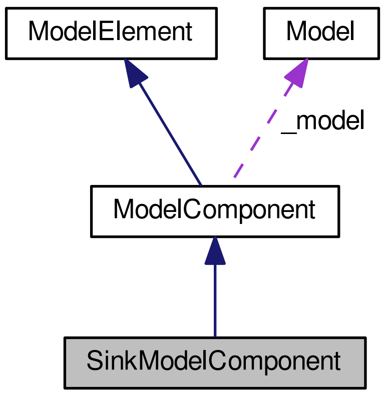
5.85 SinkModelComponent Class Reference

```
#include <SinkModelComponent.h>
```

Inheritance diagram for SinkModelComponent:



Collaboration diagram for SinkModelComponent:



Public Member Functions

- SinkModelComponent (Model ∗model, std::string componentTypename)
- SinkModelComponent (const SinkModelComponent &orig)
- virtual ∼SinkModelComponent ()
- void setCollectStatistics (bool _collectStatistics)
- bool isCollectStatistics () const

Protected Member Functions

- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.85.1 Detailed Description

This class is the basis for any component representing the end of a process flow, such as a Dispose. It can remove entities from the system and collect statistics.

5.85.2 Constructor & Destructor Documentation

5.85.2.1 SinkModelComponent::SinkModelComponent (Model ∗ *model*, std::string *componentTypename*)

5.85.2.2 SinkModelComponent::SinkModelComponent (const SinkModelComponent & *orig*)

5.85.2.3 SinkModelComponent::∼SinkModelComponent ( ) [virtual]

5.85.3 Member Function Documentation

5.85.3.1 bool SinkModelComponent::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

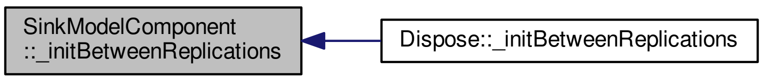
Reimplemented in Dispose.

5.85.3.2 void SinkModelComponent::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

Reimplemented in Dispose.

Here is the caller graph for this function:

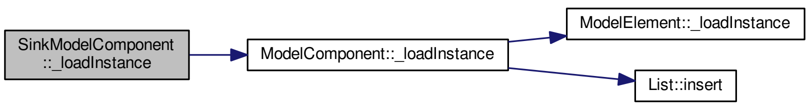

SinkModelComponent ::_initBetweenReplications
Dispose::_initBetweenReplications

5.85.3.3 bool SinkModelComponent::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelComponent.

Reimplemented in Dispose.

Here is the call graph for this function:

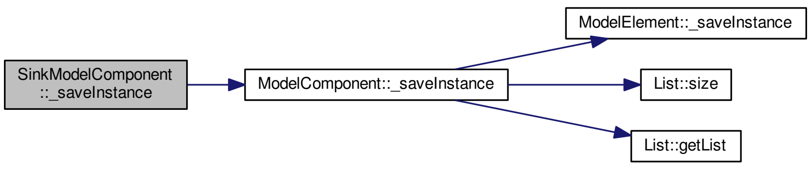

SinkModelComponent ::_loadInstance
ModelComponent::_loadInstance
ModelElement::_loadInstance
List::insert

5.85.3.4 std::map< std::string, std::string > * SinkModelComponent::_saveInstance ( ) [protected], [virtual]

Reimplemented from ModelComponent.

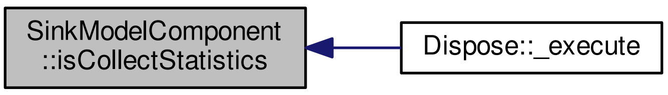
Reimplemented in Dispose.

Here is the call graph for this function:


SinkModelComponent ::_saveInstance
ModelComponent::_saveInstance
ModelElement::_saveInstance
List::size
List::getList

5.85.3.5 bool SinkModelComponent::isCollectStatistics ( ) const

Here is the caller graph for this function:



5.85.3.6 void SinkModelComponent::setCollectStatistics (bool *_collectStatistics*)

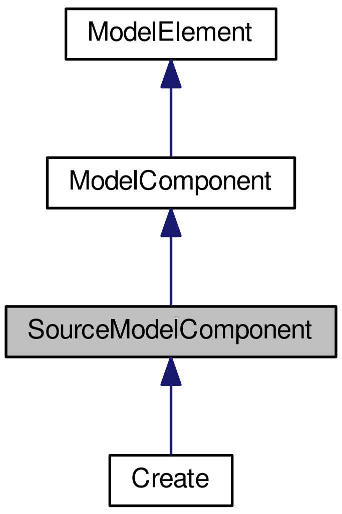
The documentation for this class was generated from the following files:

- SinkModelComponent.h
- SinkModelComponent.cpp

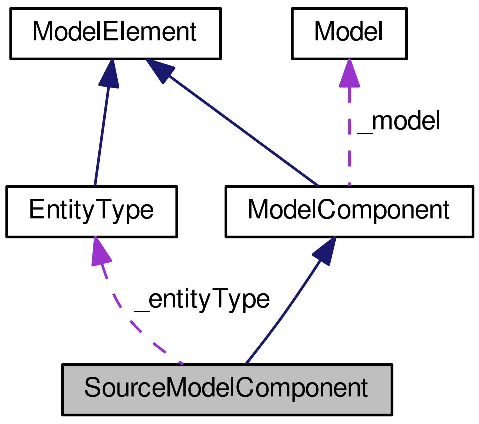
5.86 SourceModelComponent Class Reference

```
#include <SourceModelComponent.h>
```

Inheritance diagram for SourceModelComponent:



Collaboration diagram for SourceModelComponent:



Public Member Functions

- SourceModelComponent (Model ∗model, std::string componentTypename)
- SourceModelComponent (const SourceModelComponent &orig)
- virtual ∼SourceModelComponent ()
- void setFirstCreation (double _firstCreation)
- double getFirstCreation () const
- void setCollectStatistics (bool _collectStatistics)
- bool isCollectStatistics () const
- void setEntityType (EntityType ∗_entityType)
- EntityType ∗ getEntityType () const
- void setTimeUnit (Util::TimeUnit _timeUnit)
- Util::TimeUnit getTimeUnit () const
- void setTimeBetweenCreationsExpression (std::string _timeBetweenCreations)
- std::string getTimeBetweenCreationsExpression () const
- void setMaxCreations (std::string _maxCreationsExpression)
- std::string getMaxCreations () const
- unsigned int getEntitiesCreated () const
- void setEntitiesCreated (unsigned int _entitiesCreated)
- void setEntitiesPerCreation (unsigned int _entitiesPerCreation)
- unsigned int getEntitiesPerCreation () const
- virtual std::string show ()

Protected Member Functions

- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual void _initBetweenReplications ()
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Protected Attributes

- EntityType ∗ _entityType
- double _firstCreation = 0.0
- unsigned int _entitiesPerCreation = 1
- std::string _maxCreationsExpression = std::to_string(std::numeric_limits<unsigned int>::max())
- std::string _timeBetweenCreationsExpression = "EXPO(1)"
- Util::TimeUnit _timeBetweenCreationsTimeUnit = Util::TimeUnit::second
- unsigned int _entitiesCreatedSoFar = 0

Additional Inherited Members

5.86.1 Detailed Description

A source component implements the base for inserting entities into the model when its simulation is initialized. During the initialization, the new and empty future events list is populated by events of creating entities and sending them to the source components existing in the model

5.86.2 Constructor & Destructor Documentation

5.86.2.1 SourceModelComponent::SourceModelComponent (**Model** ∗ *model,* std::string *componentTypename*)

5.86.2.2 SourceModelComponent::SourceModelComponent (const **SourceModelComponent** & *orig*)

5.86.2.3 SourceModelComponent::∼SourceModelComponent ( ) [virtual]

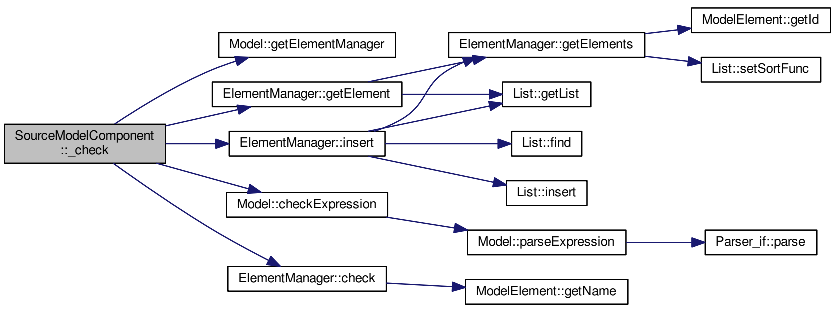
5.86.3 Member Function Documentation

5.86.3.1 bool SourceModelComponent::_check (std::string ∗ *errorMessage*) [protected], [virtual]

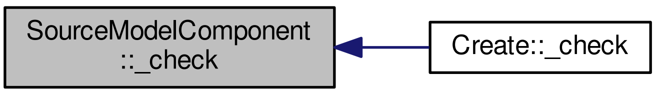
Reimplemented from ModelElement.

Reimplemented in Create.

Here is the call graph for this function:



Here is the caller graph for this function:

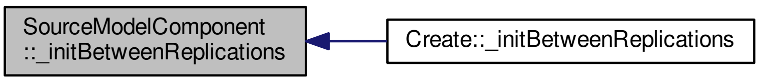


5.86.3.2 void SourceModelComponent::_initBetweenReplications ( ) [protected], [virtual]

Implements ModelComponent.

Reimplemented in Create.

Here is the caller graph for this function:

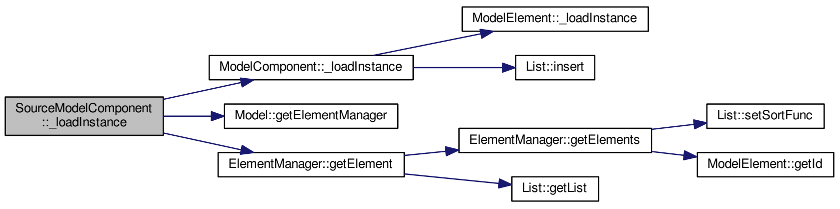

SourceModelComponent ::_initBetweenReplications
Create::_initBetweenReplications

5.86.3.3 bool SourceModelComponent::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

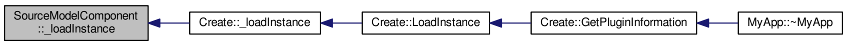
Reimplemented from ModelComponent.

Reimplemented in Create.

Here is the call graph for this function:


SourceModelComponent ::_loadInstance
ModelComponent::_loadInstance
Model::getElementManager
ElementManager::getElement
ModelElement::_loadInstance
List::insert
ElementManager::getElements
List::getList
List::setSortFunc
ModelElement::getId

Here is the caller graph for this function:

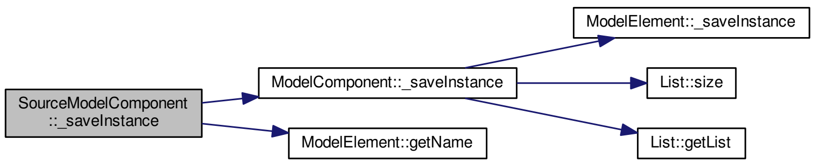

SourceModelComponent ::_loadInstance
Create::_loadInstance
Create::LoadInstance
Create::GetPluginInformation
MyApp::~MyApp

5.86.3.4 std::map< std::string, std::string > ∗ SourceModelComponent::_saveInstance ( ) [protected], [virtual]

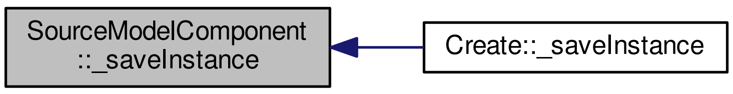
Reimplemented from ModelComponent.

Reimplemented in Create.

Here is the call graph for this function:


ModelElement::_saveInstance
ModelComponent::_saveInstance
List::size
SourceModelComponent ::_saveInstance
ModelElement::getName
List::getList

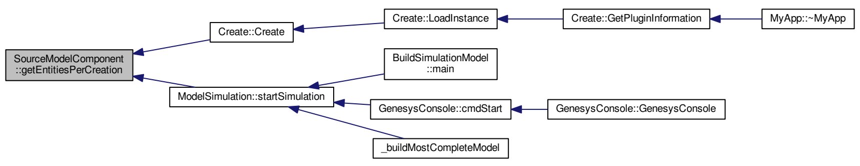
Here is the caller graph for this function:



5.86.3.5 unsigned int SourceModelComponent::getEntitiesCreated ( ) const

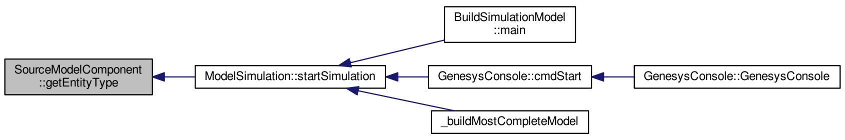
5.86.3.6 unsigned int SourceModelComponent::getEntitiesPerCreation ( ) const

Here is the caller graph for this function:


Create::LoadInstance
Create::GetPluginInformation
MyApp::~MyApp
Create::Create
SourceModelComponent ::getEntitiesPerCreation
BuildSimulationModel ::main
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

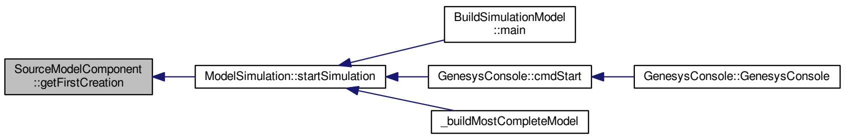
5.86.3.7 EntityType ∗ SourceModelComponent::getEntityType ( ) const

Here is the caller graph for this function:


SourceModelComponent ::getEntityType
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.86.3.8 double SourceModelComponent::getFirstCreation ( ) const

Here is the caller graph for this function:


SourceModelComponent ::getFirstCreation
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.86.3.9 std::string SourceModelComponent::getMaxCreations ( ) const

5.86.3.10 std::string SourceModelComponent::getTimeBetweenCreationsExpression ( ) const

5.86.3.11 Util::TimeUnit SourceModelComponent::getTimeUnit ( ) const

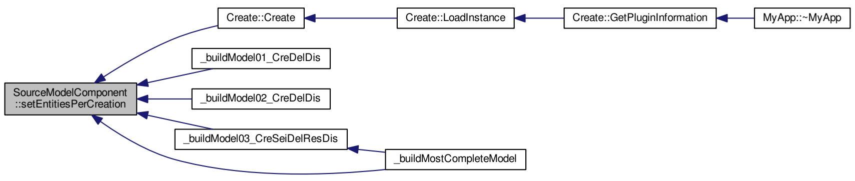
5.86.3.12 bool SourceModelComponent::isCollectStatistics ( ) const

5.86.3.13 void SourceModelComponent::setCollectStatistics (bool *_collectStatistics*)

5.86.3.14 void SourceModelComponent::setEntitiesCreated (unsigned int *_entitiesCreated*)

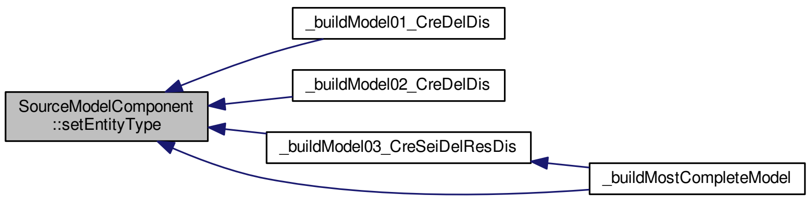
5.86.3.15 void SourceModelComponent::setEntitiesPerCreation (unsigned int *_entitiesPerCreation*)

Here is the caller graph for this function:



5.86.3.16 void SourceModelComponent::setEntityType (EntityType ∗ *_entityType*)

Here is the caller graph for this function:

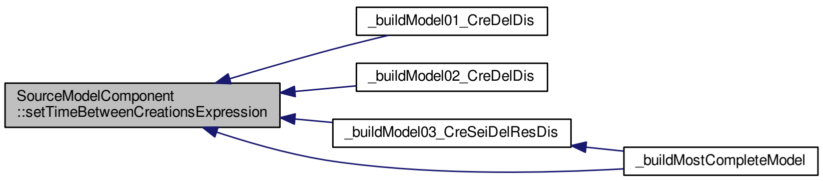


5.86.3.17 void SourceModelComponent::setFirstCreation (double *_firstCreation*)

5.86.3.18 void SourceModelComponent::setMaxCreations (std::string *_maxCreationsExpression*)

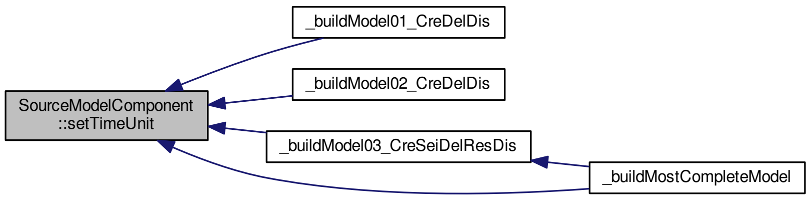
5.86.3.19 void SourceModelComponent::setTimeBetweenCreationsExpression (std::string *_timeBetweenCreations*)

Here is the caller graph for this function:


SourceModelComponent ::setTimeBetweenCreationsExpression
_buildModel01_CreDelDis
_buildModel02_CreDelDis
_buildModel03_CreSeiDelResDis
_buildMostCompleteModel

5.86.3.20 void SourceModelComponent::setTimeUnit (Util::TimeUnit *_timeUnit*)

Here is the caller graph for this function:

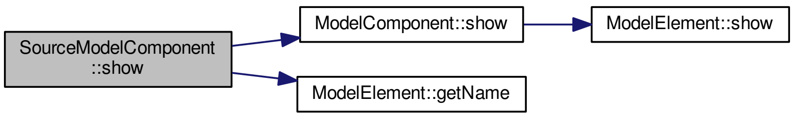


5.86.3.21 std::string SourceModelComponent::show ( ) [virtual]

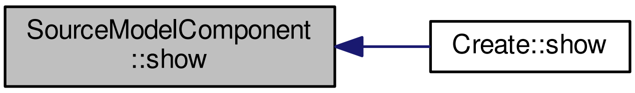
Reimplemented from ModelComponent.

Reimplemented in Create.

Here is the call graph for this function:



Here is the caller graph for this function:



5.86.4 Member Data Documentation

5.86.4.1 unsigned int SourceModelComponent::_entitiesCreatedSoFar = 0 [protected]

5.86.4.2 unsigned int SourceModelComponent::_entitiesPerCreation = 1 [protected]

5.86.4.3 EntityType∗ SourceModelComponent::_entityType [protected]

5.86.4.4 double SourceModelComponent::_firstCreation = 0.0 [protected]

5.86.4.5 std::string SourceModelComponent::_maxCreationsExpression = std::to_string(std::numeric_limits<unsigned int>::max()) [protected]

5.86.4.6 std::string SourceModelComponent::_timeBetweenCreationsExpression = "EXPO(1)" [protected]

5.86.4.7 Util::TimeUnit SourceModelComponent::_timeBetweenCreationsTimeUnit = Util::TimeUnit::second [protected]

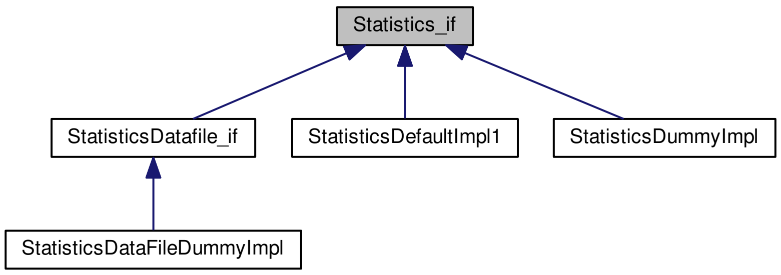
The documentation for this class was generated from the following files:

- SourceModelComponent.h
- SourceModelComponent.cpp

5.87 Statistics_if Class Reference

```
#include <Statistics_if.h>
```

Inheritance diagram for Statistics_if:



Public Member Functions

- virtual Collector_if ∗ getCollector ()=0
- virtual void setCollector (Collector_if ∗collector)=0
- virtual unsigned int numElements ()=0
- virtual double min ()=0
- virtual double max ()=0
- virtual double average ()=0
- virtual double variance ()=0
- virtual double stddeviation ()=0
- virtual double variationCoef ()=0
- virtual double halfWidthConfidenceInterval ()=0
- virtual unsigned int newSampleSize (double halfWidth)=0
- virtual double getConfidenceLevel ()=0
- virtual void setConfidenceLevel (double confidencelevel)=0

5.87.1 Detailed Description

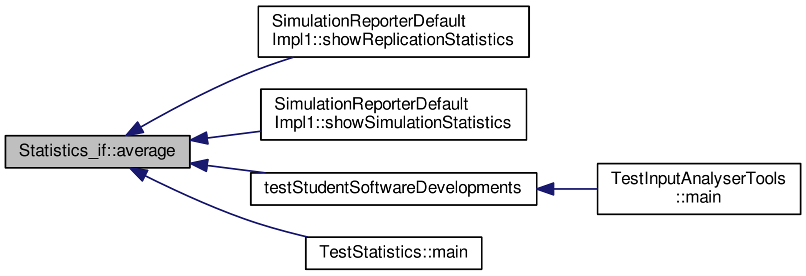
Interface for statisct synthesis of a stochastic variable collected by a Collector_if. The statistics generated may be updated based only on the previous statistics and the single newest added value or they may be updated based on a datafile, depending on the Collector implementation.

5.87.2 Member Function Documentation

5.87.2.1 virtual double Statistics_if::average ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

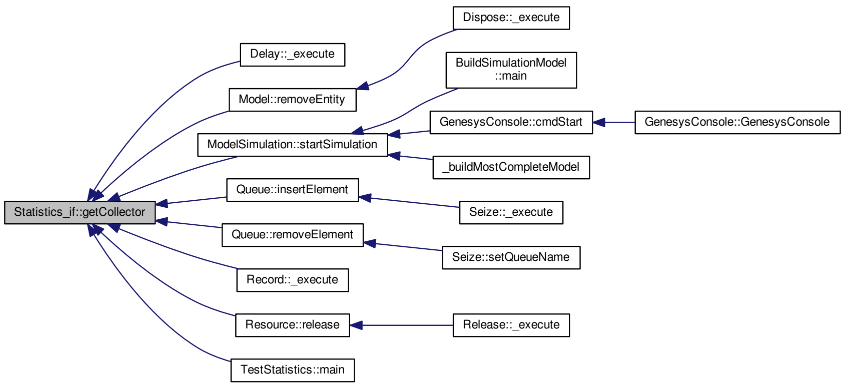
Here is the caller graph for this function:


Statistics_if::average
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main
TestStatistics::main

5.87.2.2 virtual Collector_if∗ Statistics_if::getCollector ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

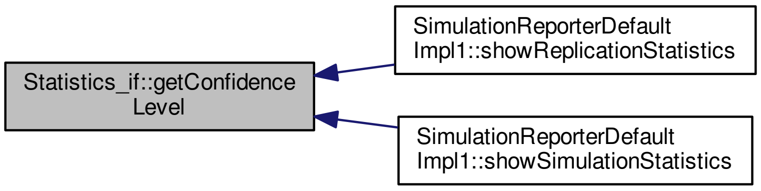
Here is the caller graph for this function:



5.87.2.3 virtual double Statistics_if::getConfidenceLevel ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, and StatisticsDummyImpl.

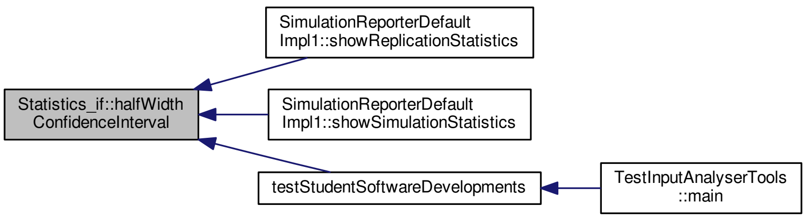
Here is the caller graph for this function:


Statistics_if::getConfidence Level
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics

5.87.2.4 virtual double Statistics_if::halfWidthConfidenceInterval ( ) `[pure virtual]`

Implemented in StatisticsDefaultImpl1, and StatisticsDummyImpl.

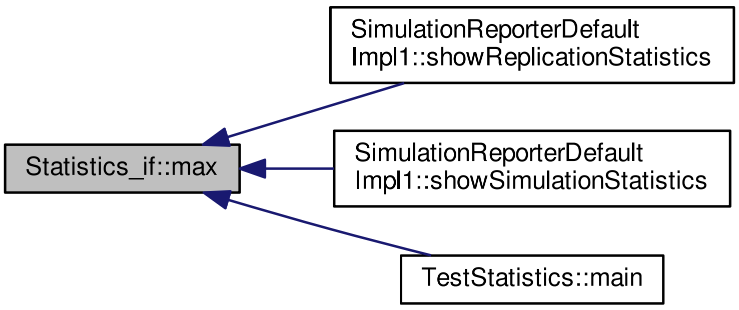
Here is the caller graph for this function:


Statistics_if::halfWidth ConfidenceInterval
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.87.2.5 virtual double Statistics_if::max ( ) `[pure virtual]`

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

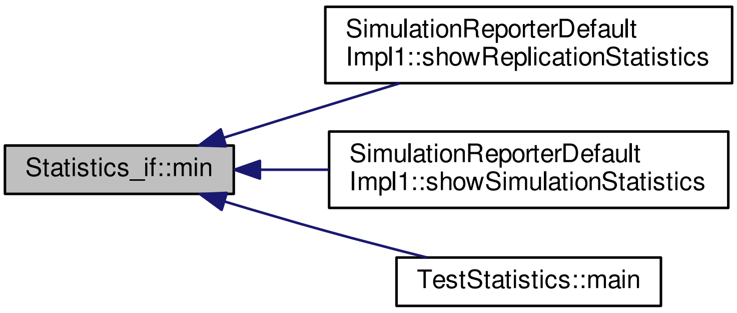
Here is the caller graph for this function:


Statistics_if::max
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
TestStatistics::main

5.87.2.6 virtual double Statistics_if::min ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

Here is the caller graph for this function:


Statistics_if::min
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
TestStatistics::main

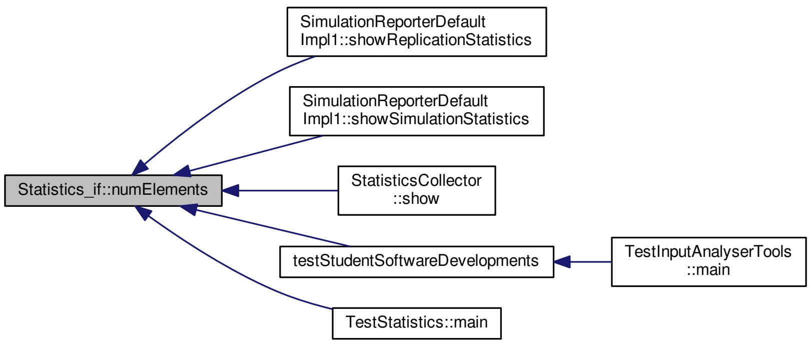
5.87.2.7 virtual unsigned int Statistics_if::newSampleSize (double *halfWidth*) [pure virtual]

Implemented in StatisticsDefaultImpl1, and StatisticsDummyImpl.

5.87.2.8 virtual unsigned int Statistics_if::numElements ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

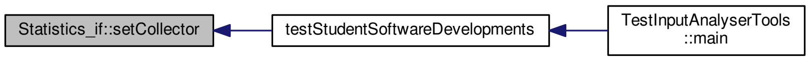
Here is the caller graph for this function:


Statistics_if::numElements
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
StatisticsCollector ::show
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main
TestStatistics::main

5.87.2.9 virtual void Statistics_if::setCollector (Collector_if * *collector*) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

Here is the caller graph for this function:


Statistics_if::setCollector
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

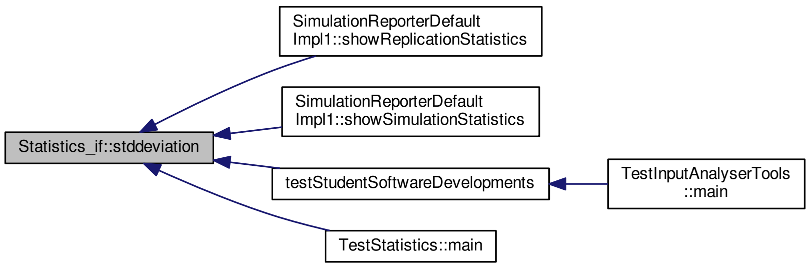
5.87.2.10 virtual void Statistics_if::setConfidenceLevel (double *confidencelevel*) [pure virtual]

Implemented in StatisticsDefaultImpl1, and StatisticsDummyImpl.

5.87.2.11 virtual double Statistics_if::stddeviation ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

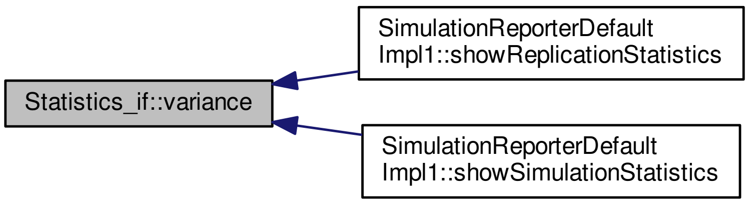
Here is the caller graph for this function:


SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
Statistics_if::stddeviation
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main
TestStatistics::main

5.87.2.12 virtual double Statistics_if::variance ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

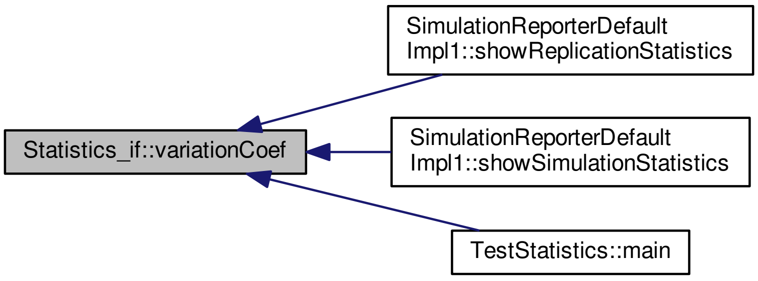
Here is the caller graph for this function:


SimulationReporterDefault Impl1::showReplicationStatistics
Statistics_if::variance
SimulationReporterDefault Impl1::showSimulationStatistics

5.87.2.13 virtual double Statistics_if::variationCoef ( ) [pure virtual]

Implemented in StatisticsDefaultImpl1, StatisticsDummyImpl, and StatisticsDataFileDummyImpl.

Here is the caller graph for this function:


SimulationReporterDefault Impl1::showReplicationStatistics
Statistics_if::variationCoef
SimulationReporterDefault Impl1::showSimulationStatistics
TestStatistics::main

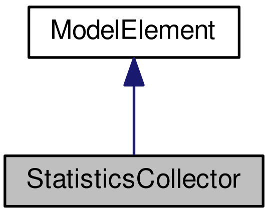
The documentation for this class was generated from the following file:

- Statistics_if.h

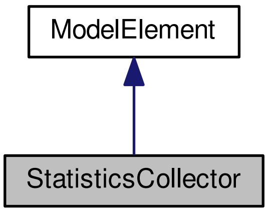
5.88 StatisticsCollector Class Reference

```
#include <StatisticsCollector.h>
```

Inheritance diagram for StatisticsCollector:



Collaboration diagram for StatisticsCollector:



Public Member Functions

- StatisticsCollector ()
- StatisticsCollector (std::string name)
- StatisticsCollector (std::string name, ModelElement ∗parent)
- StatisticsCollector (const StatisticsCollector &orig)
- virtual ∼StatisticsCollector ()
- virtual std::string show ()
- ModelElement ∗ getParent () const
- Statistics_if ∗ getStatistics () const

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

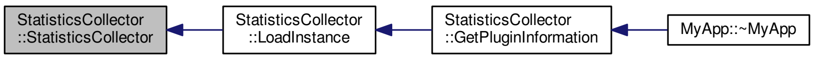
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.88.1 Constructor & Destructor Documentation

5.88.1.1 StatisticsCollector::StatisticsCollector ( )

Here is the caller graph for this function:


StatisticsCollector ::StatisticsCollector
StatisticsCollector ::LoadInstance
StatisticsCollector ::GetPluginInformation
MyApp::~MyApp

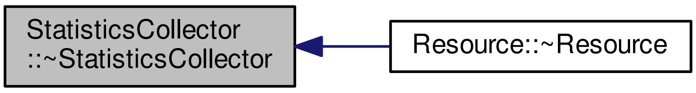
5.88.1.2 StatisticsCollector::StatisticsCollector (std::string *name*)

5.88.1.3 StatisticsCollector::StatisticsCollector (std::string *name,* ModelElement ∗ *parent*)

5.88.1.4 StatisticsCollector::StatisticsCollector (const StatisticsCollector & *orig*)

5.88.1.5 StatisticsCollector::∼StatisticsCollector ( ) [virtual]

Here is the caller graph for this function:



5.88.2 Member Function Documentation

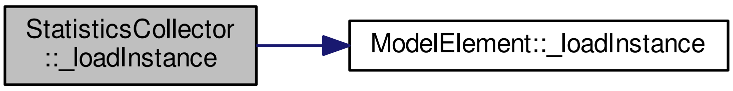
5.88.2.1 bool StatisticsCollector::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

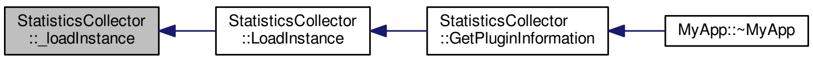
5.88.2.2 bool StatisticsCollector::_loadInstance (std::map< std::string, std::string > ∗ *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:


StatisticsCollector ::_loadInstance
ModelElement::_loadInstance

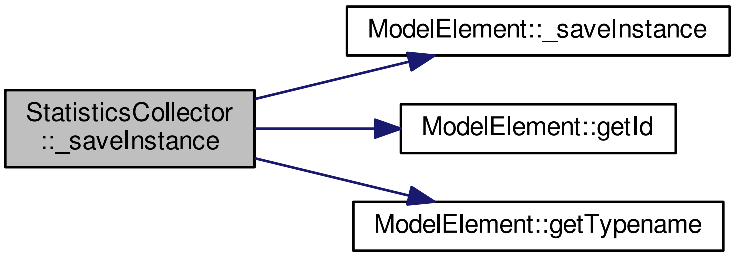
Here is the caller graph for this function:


StatisticsCollector ::_loadInstance
StatisticsCollector ::LoadInstance
StatisticsCollector ::GetPluginInformation
MyApp::~MyApp

5.88.2.3 std::map< std::string, std::string > * StatisticsCollector::_saveInstance ( ) [protected], [virtual]

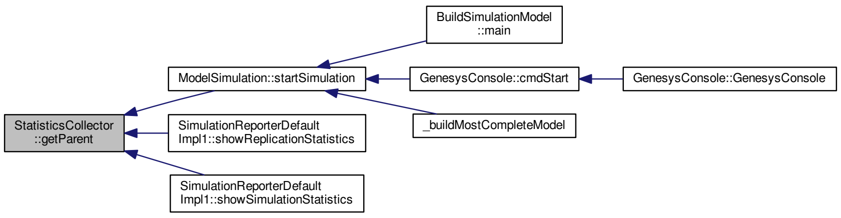
Reimplemented from ModelElement.

Here is the call graph for this function:


StatisticsCollector ::_saveInstance
ModelElement::_saveInstance
ModelElement::getId
ModelElement::getTypename

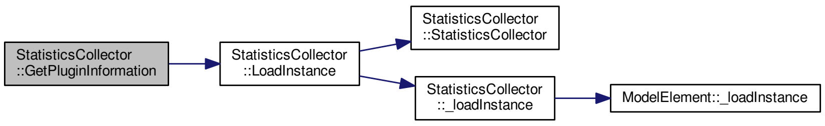
5.88.2.4 ModelElement * StatisticsCollector::getParent ( ) const

Here is the caller graph for this function:

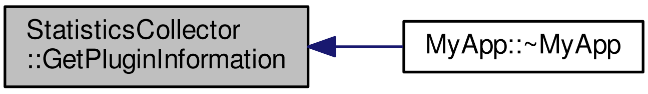

StatisticsCollector ::getParent
ModelSimulation::startSimulation
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

5.88.2.5 PluginInformation * StatisticsCollector::GetPluginInformation ( ) [static]

Here is the call graph for this function:

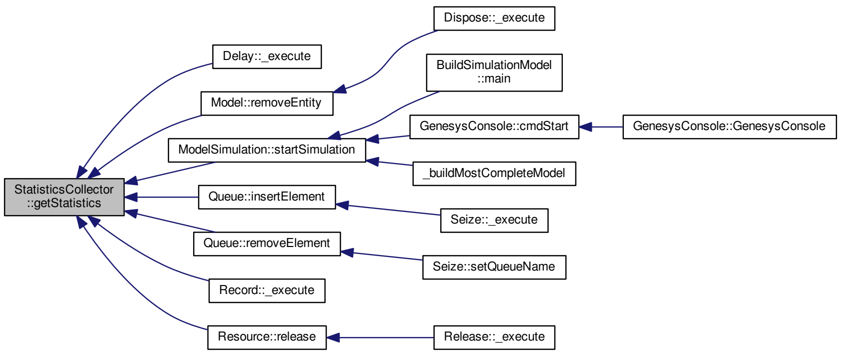

StatisticsCollector ::GetPluginInformation
StatisticsCollector ::LoadInstance
StatisticsCollector ::StatisticsCollector
StatisticsCollector ::_loadInstance
ModelElement::_loadInstance

Here is the caller graph for this function:



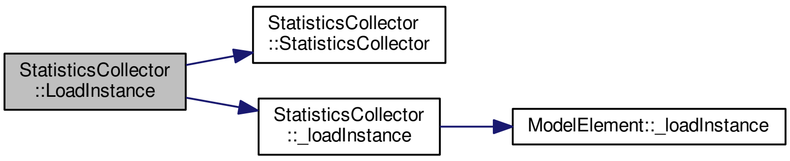
5.88.2.6 Statistics_if ∗ StatisticsCollector::getStatistics ( ) const

Here is the caller graph for this function:

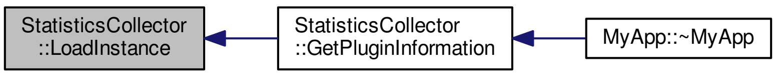


5.88.2.7 ModelElement ∗ StatisticsCollector::LoadInstance (ElementManager ∗ *elems*, std::map< std::string, std::string > ∗ *fields*) [static]

Here is the call graph for this function:


StatisticsCollector ::LoadInstance
StatisticsCollector ::StatisticsCollector
StatisticsCollector ::_loadInstance
ModelElement::_loadInstance

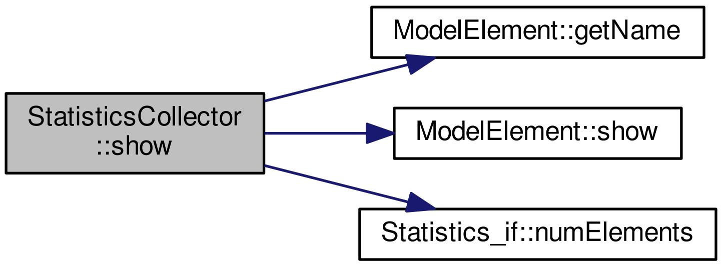
Here is the caller graph for this function:


StatisticsCollector ::LoadInstance
StatisticsCollector ::GetPluginInformation
MyApp::~MyApp

5.88.2.8 std::string StatisticsCollector::show ( ) [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



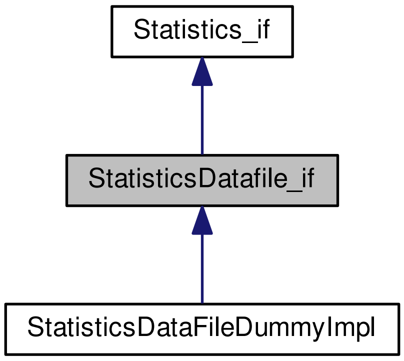
The documentation for this class was generated from the following files:

- StatisticsCollector.h
- StatisticsCollector.cpp

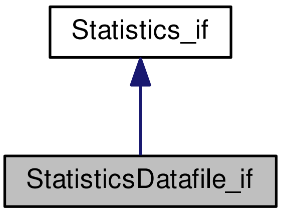
5.89 StatisticsDatafile_if Class Reference

```
#include <StatisticsDataFile_if.h>
```

Inheritance diagram for StatisticsDatafile_if:



Collaboration diagram for StatisticsDatafile_if:



Public Member Functions

- virtual double mode ()=0
- virtual double mediane ()=0
- virtual double quartil (unsigned short num)=0
- virtual double decil (unsigned short num)=0
- virtual double centil (unsigned short num)=0
- virtual void setHistogramNumClasses (unsigned short num)=0
- virtual unsigned short histogramNumClasses ()=0
- virtual double histogramClassLowerLimit (unsigned short classNum)=0
- virtual unsigned int histogramClassFrequency (unsigned short classNum)=0

5.89.1 Member Function Documentation

5.89.1.1 virtual double StatisticsDatafile_if::centil (unsigned short *num*) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.2 virtual double StatisticsDatafile_if::decil (unsigned short *num*) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.3 virtual unsigned int StatisticsDatafile_if::histogramClassFrequency (unsigned short *classNum*) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.4 virtual double StatisticsDatafile_if::histogramClassLowerLimit (unsigned short *classNum*) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.5 virtual unsigned short StatisticsDatafile_if::histogramNumClasses ( ) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.6 virtual double StatisticsDatafile_if::mediane ( ) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.7 virtual double StatisticsDatafile_if::mode ( ) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.8 virtual double StatisticsDatafile_if::quartil (unsigned short *num*) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

5.89.1.9 virtual void StatisticsDatafile_if::setHistogramNumClasses (unsigned short *num*) `[pure virtual]`

Implemented in StatisticsDataFileDummyImpl.

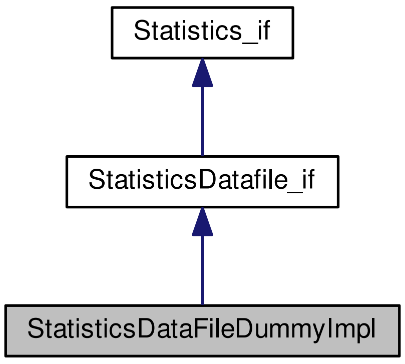
The documentation for this class was generated from the following file:

- StatisticsDataFile_if.h

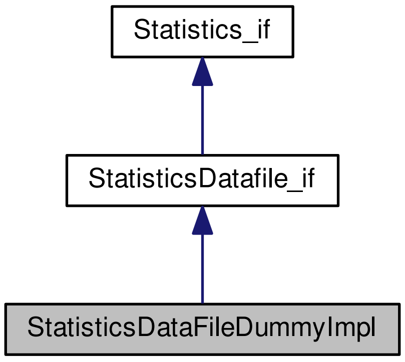
5.90 StatisticsDataFileDummyImpl Class Reference

```
#include <StatisticsDataFileDummyImpl.h>
```

Inheritance diagram for StatisticsDataFileDummyImpl:



Collaboration diagram for StatisticsDataFileDummyImpl:



Public Member Functions

- StatisticsDataFileDummyImpl ()
- StatisticsDataFileDummyImpl (const StatisticsDataFileDummyImpl &orig)
- virtual ∼StatisticsDataFileDummyImpl ()
- virtual Collector_if ∗ getCollector ()
- void setCollector (Collector_if ∗collector)

- virtual unsigned int numElements ()
- virtual double min ()
- virtual double max ()
- virtual double average ()
- virtual double variance ()
- virtual double stddeviation ()
- virtual double variationCoef ()
- virtual double halfWidthConfidenceInterval (double confidencelevel)
- virtual unsigned int newSampleSize (double confidencelevel, double halfWidth)
- virtual double mode ()
- virtual double mediane ()
- virtual double quartil (unsigned short num)
- virtual double decil (unsigned short num)
- virtual double centil (unsigned short num)
- virtual void setHistogramNumClasses (unsigned short num)
- virtual unsigned short histogramNumClasses ()
- virtual double histogramClassLowerLimit (unsigned short classNum)
- virtual unsigned int histogramClassFrequency (unsigned short classNum)

5.90.1 Constructor & Destructor Documentation

5.90.1.1 StatisticsDataFileDummyImpl::StatisticsDataFileDummyImpl ( )

5.90.1.2 StatisticsDataFileDummyImpl::StatisticsDataFileDummyImpl (const StatisticsDataFileDummyImpl & *orig*)

5.90.1.3 StatisticsDataFileDummyImpl::∼StatisticsDataFileDummyImpl ( ) [virtual]

5.90.2 Member Function Documentation

5.90.2.1 double StatisticsDataFileDummyImpl::average ( ) [virtual]

Implements Statistics_if.

5.90.2.2 double StatisticsDataFileDummyImpl::centil (unsigned short *num*) [virtual]

Implements StatisticsDatafile_if.

5.90.2.3 double StatisticsDataFileDummyImpl::decil (unsigned short *num*) [virtual]

Implements StatisticsDatafile_if.

5.90.2.4 Collector_if ∗ StatisticsDataFileDummyImpl::getCollector ( ) [virtual]

Implements Statistics_if.

5.90.2.5 double StatisticsDataFileDummyImpl::halfWidthConfidenceInterval (double *confidencelevel*) [virtual]

5.90.2.6 unsigned int StatisticsDataFileDummyImpl::histogramClassFrequency (unsigned short *classNum*) [virtual]

Implements StatisticsDatafile_if.

5.90.2.7 double StatisticsDataFileDummyImpl::histogramClassLowerLimit (unsigned short *classNum*) [virtual]

Implements StatisticsDatafile_if.

5.90.2.8 unsigned short StatisticsDataFileDummyImpl::histogramNumClasses ( ) [virtual]

Implements StatisticsDatafile_if.

5.90.2.9 double StatisticsDataFileDummyImpl::max ( ) [virtual]

Implements Statistics_if.

5.90.2.10 double StatisticsDataFileDummyImpl::mediane ( ) [virtual]

Implements StatisticsDatafile_if.

5.90.2.11 double StatisticsDataFileDummyImpl::min ( ) [virtual]

Implements Statistics_if.

5.90.2.12 double StatisticsDataFileDummyImpl::mode ( ) [virtual]

Implements StatisticsDatafile_if.

5.90.2.13 unsigned int StatisticsDataFileDummyImpl::newSampleSize (double *confidencelevel,* double *halfWidth*) [virtual]

5.90.2.14 unsigned int StatisticsDataFileDummyImpl::numElements ( ) [virtual]

Implements Statistics_if.

5.90.2.15 double StatisticsDataFileDummyImpl::quartil (unsigned short *num*) [virtual]

Implements StatisticsDatafile_if.

5.90.2.16 void StatisticsDataFileDummyImpl::setCollector (Collector_if ∗ *collector*) [virtual]

Implements Statistics_if.

5.90.2.17 void StatisticsDataFileDummyImpl::setHistogramNumClasses (unsigned short *num*) [virtual]

Implements StatisticsDatafile_if.

5.90.2.18 double StatisticsDataFileDummyImpl::stddeviation ( ) [virtual]

Implements Statistics_if.

5.90.2.19 double StatisticsDataFileDummyImpl::variance ( ) [virtual]

Implements Statistics_if.

5.90.2.20 double StatisticsDataFileDummyImpl::variationCoef ( ) [virtual]

Implements Statistics_if.

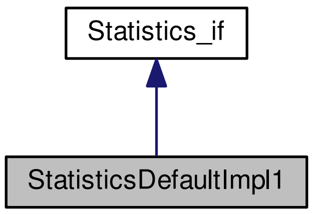
The documentation for this class was generated from the following files:

- StatisticsDataFileDummyImpl.h
- StatisticsDataFileDummyImpl.cpp

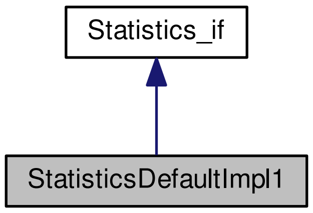
5.91 StatisticsDefaultImpl1 Class Reference

```
#include <StatisticsDefaultImpl1.h>
```

Inheritance diagram for StatisticsDefaultImpl1:



Collaboration diagram for StatisticsDefaultImpl1:



Public Member Functions

- StatisticsDefaultImpl1 ()

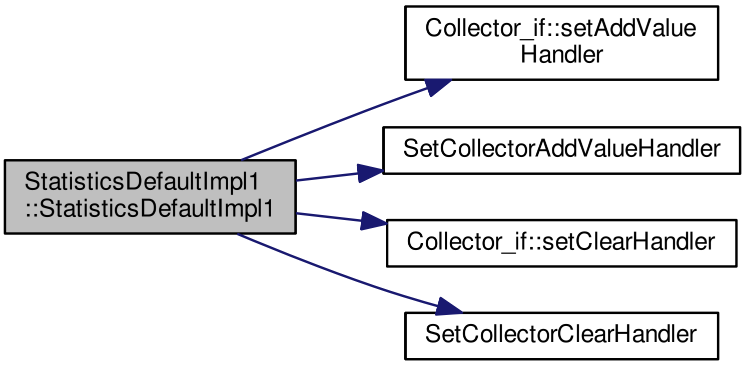
    When constructor is invoked without a Collector, it is taken from Traits<Statistics_if>::CollectorImplementation configuration.
- StatisticsDefaultImpl1 (Collector_if ∗collector)
- StatisticsDefaultImpl1 (const StatisticsDefaultImpl1 &orig)
- virtual ∼StatisticsDefaultImpl1 ()
- virtual Collector_if ∗ getCollector ()
- virtual void setCollector (Collector_if ∗collector)
- virtual unsigned int numElements ()
- virtual double min ()
- virtual double max ()
- virtual double average ()
- virtual double variance ()
- virtual double stddeviation ()
- virtual double variationCoef ()
- virtual double halfWidthConfidenceInterval ()
- virtual unsigned int newSampleSize (double halfWidth)
- virtual double getConfidenceLevel ()
- virtual void setConfidenceLevel (double confidencelevel)

5.91.1 Constructor & Destructor Documentation

5.91.1.1 StatisticsDefaultImpl1::StatisticsDefaultImpl1 ( )

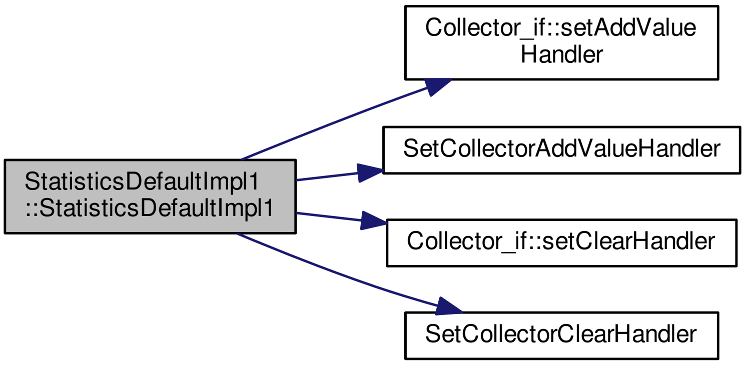
When constructor is invoked without a Collector, it is taken from Traits<Statistics_if>::CollectorImplementation configuration.

Here is the call graph for this function:


Collector_if::setAddValue Handler
SetCollectorAddValueHandler
StatisticsDefaultImpl1 ::StatisticsDefaultImpl1
Collector_if::setClearHandler
SetCollectorClearHandler

5.91.1.2 StatisticsDefaultImpl1::StatisticsDefaultImpl1 (**Collector_if** ∗ *collector*)

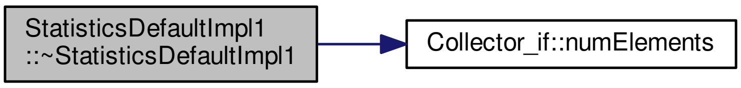
Here is the call graph for this function:


Collector_if::setAddValue Handler
SetCollectorAddValueHandler
StatisticsDefaultImpl1 ::StatisticsDefaultImpl1
Collector_if::setClearHandler
SetCollectorClearHandler

5.91.1.3 StatisticsDefaultImpl1::StatisticsDefaultImpl1 (const **StatisticsDefaultImpl1** & *orig*)

5.91.1.4 StatisticsDefaultImpl1::∼StatisticsDefaultImpl1 ( ) `[virtual]`

Here is the call graph for this function:



5.91.2 Member Function Documentation

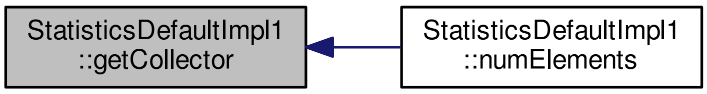
5.91.2.1 double StatisticsDefaultImpl1::average ( ) `[virtual]`

Implements Statistics_if.

5.91.2.2 Collector_if ∗ StatisticsDefaultImpl1::getCollector ( ) `[virtual]`

Implements Statistics_if.

Here is the caller graph for this function:



5.91.2.3 double StatisticsDefaultImpl1::getConfidenceLevel ( ) `[virtual]`

Implements Statistics_if.

5.91.2.4 double StatisticsDefaultImpl1::halfWidthConfidenceInterval ( ) `[virtual]`

Implements Statistics_if.

5.91.2.5 double StatisticsDefaultImpl1::max ( ) [virtual]

Implements Statistics_if.

5.91.2.6 double StatisticsDefaultImpl1::min ( ) [virtual]

Implements Statistics_if.

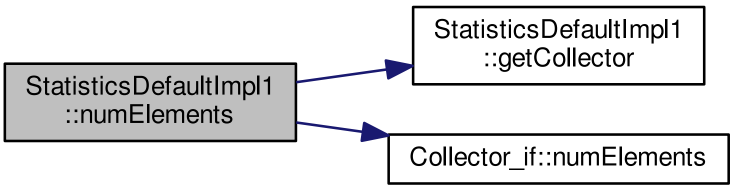
5.91.2.7 unsigned int StatisticsDefaultImpl1::newSampleSize (double *halfWidth*) [virtual]

Implements Statistics_if.

5.91.2.8 unsigned int StatisticsDefaultImpl1::numElements ( ) [virtual]

Implements Statistics_if.

Here is the call graph for this function:



5.91.2.9 void StatisticsDefaultImpl1::setCollector (Collector_if ∗ *collector*) [virtual]

Implements Statistics_if.

5.91.2.10 void StatisticsDefaultImpl1::setConfidenceLevel (double *confidencelevel*) [virtual]

Implements Statistics_if.

5.91.2.11 double StatisticsDefaultImpl1::stddeviation ( ) [virtual]

Implements Statistics_if.

5.91.2.12 double StatisticsDefaultImpl1::variance ( ) `[virtual]`

Implements Statistics_if.

5.91.2.13 double StatisticsDefaultImpl1::variationCoef ( ) `[virtual]`

Implements Statistics_if.

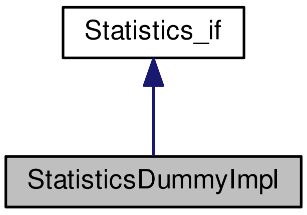
The documentation for this class was generated from the following files:

- StatisticsDefaultImpl1.h
- StatisticsDefaultImpl1.cpp

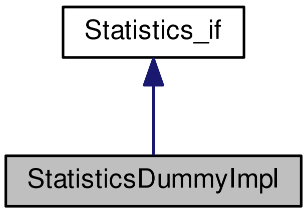
5.92 StatisticsDummyImpl Class Reference

```
#include <StatisticsDummyImpl.h>
```

Inheritance diagram for StatisticsDummyImpl:



Collaboration diagram for StatisticsDummyImpl:



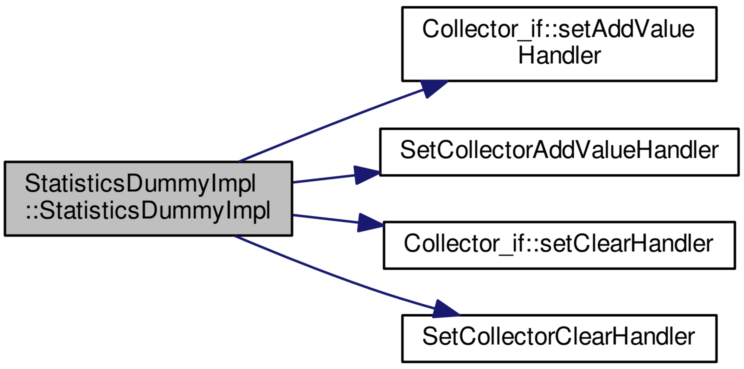
Public Member Functions

- StatisticsDummyImpl ()
- StatisticsDummyImpl (const StatisticsDummyImpl &orig)
- virtual ∼StatisticsDummyImpl ()
- virtual Collector_if ∗ getCollector ()
- void setCollector (Collector_if ∗collector)
- virtual unsigned int numElements ()
- virtual double min ()
- virtual double max ()
- virtual double average ()
- virtual double variance ()
- virtual double stddeviation ()
- virtual double variationCoef ()
- virtual double halfWidthConfidenceInterval ()
- virtual unsigned int newSampleSize (double halfWidth)
- virtual double getConfidenceLevel ()
- virtual void setConfidenceLevel (double confidencelevel)

5.92.1 Constructor & Destructor Documentation

5.92.1.1 StatisticsDummyImpl::StatisticsDummyImpl ( )

Here is the call graph for this function:


Collector_if::setAddValue Handler
SetCollectorAddValueHandler
StatisticsDummyImpl ::StatisticsDummyImpl
Collector_if::setClearHandler
SetCollectorClearHandler

5.92.1.2 StatisticsDummyImpl::StatisticsDummyImpl (const StatisticsDummyImpl & *orig*)

5.92.1.3 StatisticsDummyImpl::∼StatisticsDummyImpl ( ) [virtual]

5.92.2 Member Function Documentation

5.92.2.1 double StatisticsDummyImpl::average ( ) [virtual]

Implements Statistics_if.

5.92.2.2 Collector_if ∗ StatisticsDummyImpl::getCollector ( ) [virtual]

Implements Statistics_if.

5.92.2.3 double StatisticsDummyImpl::getConfidenceLevel ( ) [virtual]

Implements Statistics_if.

5.92.2.4 double StatisticsDummyImpl::halfWidthConfidenceInterval ( ) [virtual]

Implements Statistics_if.

5.92.2.5 double StatisticsDummyImpl::max ( ) [virtual]

Implements Statistics_if.

5.92.2.6 double StatisticsDummyImpl::min ( ) [virtual]

Implements Statistics_if.

5.92.2.7 unsigned int StatisticsDummyImpl::newSampleSize (double *halfWidth*) [virtual]

Implements Statistics_if.

5.92.2.8 unsigned int StatisticsDummyImpl::numElements ( ) [virtual]

Implements Statistics_if.

5.92.2.9 void StatisticsDummyImpl::setCollector (Collector_if ∗ *collector*) [virtual]

Implements Statistics_if.

5.92.2.10 void StatisticsDummyImpl::setConfidenceLevel (double *confidencelevel*) [virtual]

Implements Statistics_if.

5.92.2.11 double StatisticsDummyImpl::stddeviation ( ) [virtual]

Implements Statistics_if.

5.92.2.12 double StatisticsDummyImpl::variance ( ) [virtual]

Implements Statistics_if.

5.92.2.13 double StatisticsDummyImpl::variationCoef ( ) [virtual]

Implements Statistics_if.

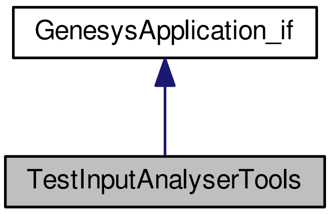
The documentation for this class was generated from the following files:

- StatisticsDummyImpl.h
- StatisticsDummyImpl.cpp

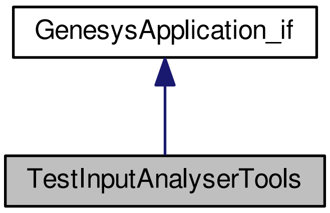
5.93 TestInputAnalyserTools Class Reference

`#include <TestInputAnalyserTools.h>`

Inheritance diagram for TestInputAnalyserTools:



Collaboration diagram for TestInputAnalyserTools:



Public Member Functions

- TestInputAnalyserTools ()
- int main (int argc, char ∗∗argv)

5.93.1 Constructor & Destructor Documentation

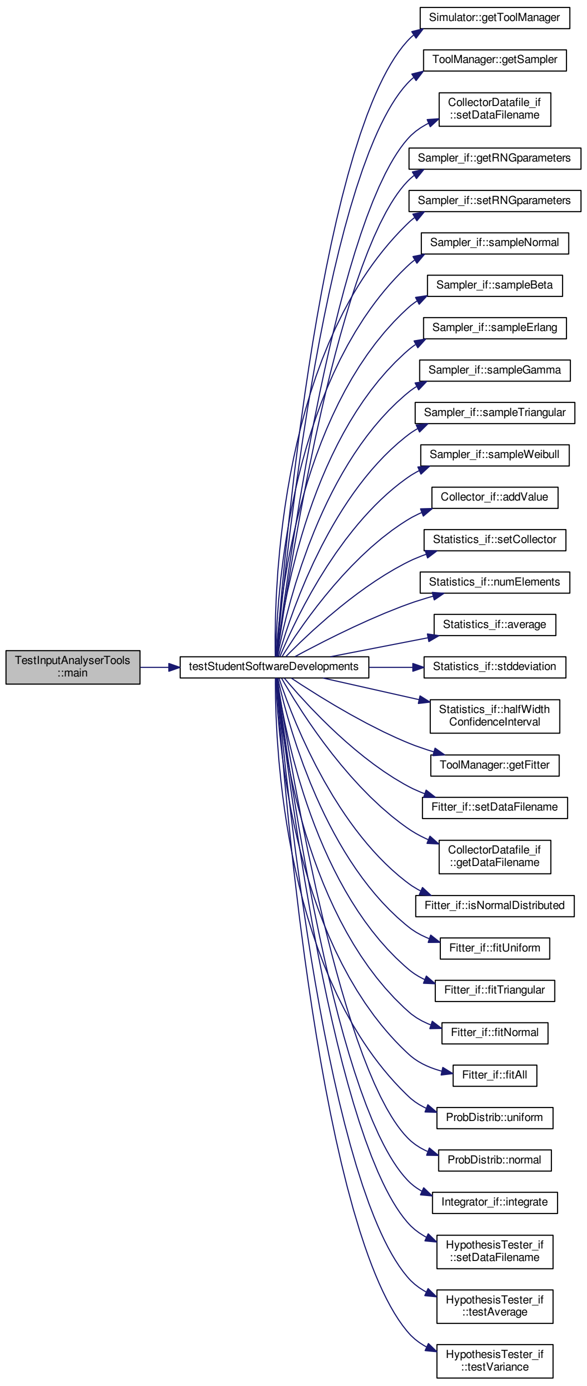
5.93.1.1 TestInputAnalyserTools::TestInputAnalyserTools ( )

5.93.2 Member Function Documentation

5.93.2.1 int TestInputAnalyserTools::main (int *argc,* char ∗∗ *argv*) `[virtual]`

Implements GenesysApplication_if.

Here is the call graph for this function:


TestInputAnalyserTools ::main
testStudentSoftwareDevelopments
Simulator::getToolManager
ToolManager::getSampler
CollectorDatafile_if ::setDataFilename
Sampler_if::getRNGparameters
Sampler_if::setRNGparameters
Sampler_if::sampleNormal
Sampler_if::sampleBeta
Sampler_if::sampleErlang
Sampler_if::sampleGamma
Sampler_if::sampleTriangular
Sampler_if::sampleWeibull
Collector_if::addValue
Statistics_if::setCollector
Statistics_if::numElements
Statistics_if::average
Statistics_if::stddeviation
Statistics_if::halfWidth ConfidenceInterval
ToolManager::getFitter
Fitter_if::setDataFilename
CollectorDatafile_if ::getDataFilename
Fitter_if::isNormalDistributed
Fitter_if::fitUniform
Fitter_if::fitTriangular
Fitter_if::fitNormal
Fitter_if::fitAll
ProbDistrib::uniform
ProbDistrib::normal
Integrator_if::integrate
HypothesisTester_if ::setDataFilename
HypothesisTester_if ::testAverage
HypothesisTester_if ::testVariance

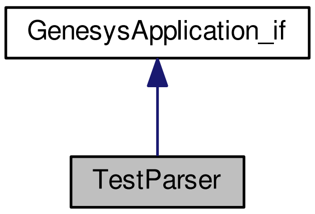
The documentation for this class was generated from the following files:

- TestInputAnalyserTools.h
- TestInputAnalyserTools.cpp

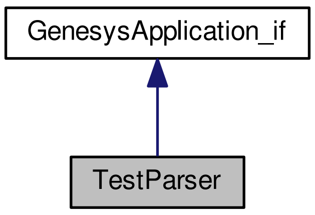
5.94 TestParser Class Reference

`#include <TestParser.h>`

Inheritance diagram for TestParser:



Collaboration diagram for TestParser:



Public Member Functions

- TestParser ()
- TestParser (const TestParser &orig)
- virtual ∼TestParser ()
- virtual int main (int argc, char ∗∗argv)

5.94.1 Constructor & Destructor Documentation

5.94.1.1 TestParser::TestParser ( )

5.94.1.2 TestParser::TestParser (const TestParser & *orig*)

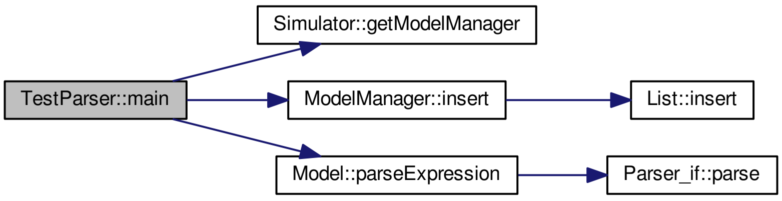
5.94.1.3 TestParser::∼TestParser ( ) `[virtual]`

5.94.2 Member Function Documentation

5.94.2.1 int TestParser::main (int *argc*, char ∗∗ *argv*) `[virtual]`

Implements GenesysApplication_if.

Here is the call graph for this function:



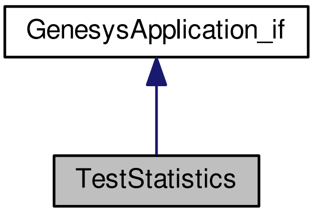
The documentation for this class was generated from the following files:

- TestParser.h
- TestParser.cpp

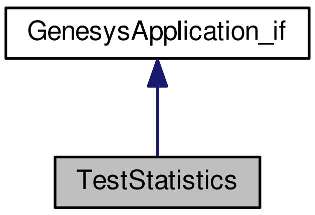
5.95 TestStatistics Class Reference

```
#include <TestStatistics.h>
```

Inheritance diagram for TestStatistics:



Collaboration diagram for TestStatistics:



Public Member Functions

- TestStatistics ()
- int main (int argc, char ∗∗argv)

5.95.1 Constructor & Destructor Documentation

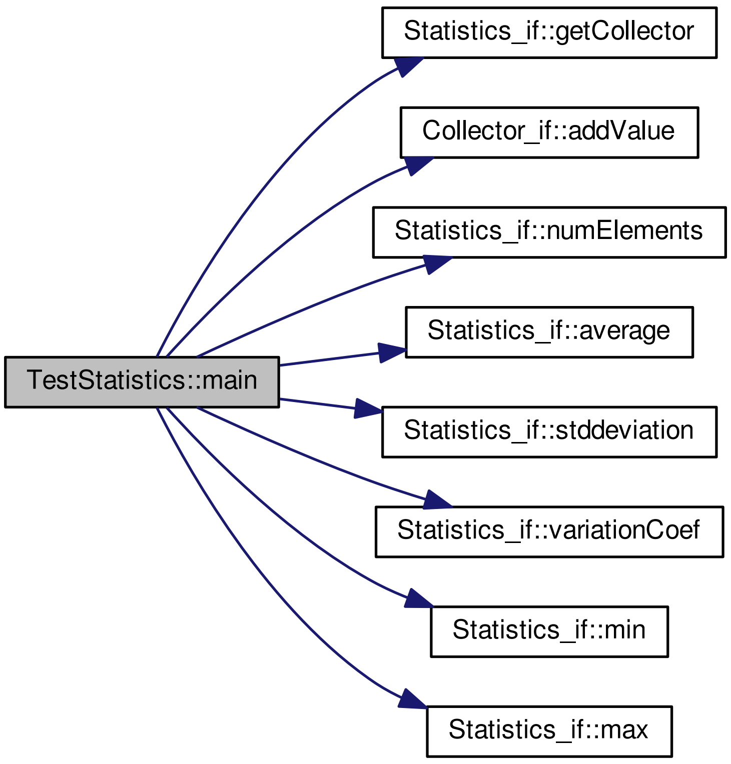
5.95.1.1 TestStatistics::TestStatistics ( )

5.95.2 Member Function Documentation

5.95.2.1 int TestStatistics::main (int *argc,* char ∗∗ *argv*) `[virtual]`

Implements GenesysApplication_if.

Here is the call graph for this function:


TestStatistics::main
Statistics_if::getCollector
Collector_if::addValue
Statistics_if::numElements
Statistics_if::average
Statistics_if::stddeviation
Statistics_if::variationCoef
Statistics_if::min
Statistics_if::max

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

- TestStatistics.h
- TestStatistics.cpp

5.96 ToolManager Class Reference

```
#include <ToolManager.h>
```

Public Member Functions

- ToolManager (Simulator ∗_simulator)
- ToolManager (const ToolManager &orig)
- virtual ∼ToolManager ()
- Sampler_if ∗ getSampler () const

  Returns the Sampler, used to generate samples accordingly to a probability distribution.
- Fitter_if ∗ getFitter () const

  Returns the fitter, responsible for carrying out tests of adherence of theoretical distributions of probability with sampled data.

5.96.1 Constructor & Destructor Documentation

5.96.1.1 ToolManager::ToolManager (Simulator $*$ *_simulator*)

5.96.1.2 ToolManager::ToolManager (const ToolManager & *orig*)

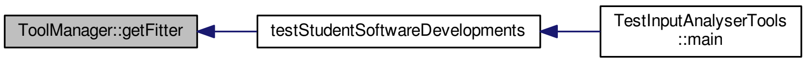
5.96.1.3 ToolManager::∼ToolManager ( ) [virtual]

5.96.2 Member Function Documentation

5.96.2.1 Fitter_if $*$ ToolManager::getFitter ( ) const

Returns the fitter, responsible for carrying out tests of adherence of theoretical distributions of probability with sampled data.

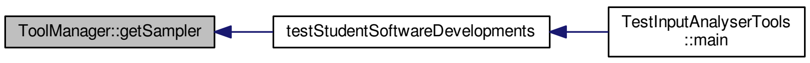
Here is the caller graph for this function:


ToolManager::getFitter
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

5.96.2.2 Sampler_if $*$ ToolManager::getSampler ( ) const

Returns the Sampler, used to generate samples accordingly to a probability distribution.

Here is the caller graph for this function:


ToolManager::getSampler
testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

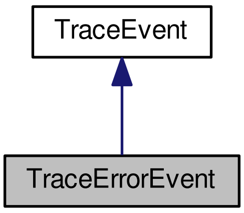
The documentation for this class was generated from the following files:

- ToolManager.h
- ToolManager.cpp

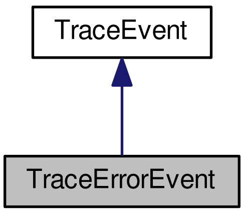
5.97 TraceErrorEvent Class Reference

```
#include <TraceManager.h>
```

Inheritance diagram for TraceErrorEvent:



Collaboration diagram for TraceErrorEvent:



Public Member Functions

- TraceErrorEvent (std::string text, std::exception e)
- std::exception getException () const

5.97.1 Constructor & Destructor Documentation

5.97.1.1 TraceErrorEvent::TraceErrorEvent (std::string *text*, std::exception *e*) `[inline]`

5.97.2 Member Function Documentation

5.97.2.1 std::exception TraceErrorEvent::getException ( ) const `[inline]`

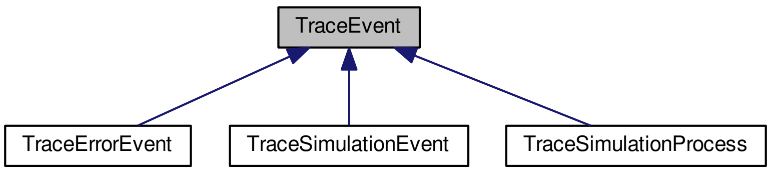
The documentation for this class was generated from the following file:

- TraceManager.h

5.98 TraceEvent Class Reference

```
#include <TraceManager.h>
```

Inheritance diagram for TraceEvent:



Public Member Functions

- TraceEvent (Util::TraceLevel tracelevel, std::string text)
- Util::TraceLevel getTracelevel () const
- std::string getText () const

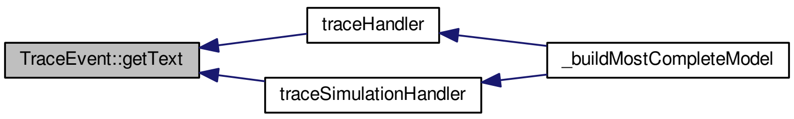
5.98.1 Constructor & Destructor Documentation

5.98.1.1 TraceEvent::TraceEvent (Util::TraceLevel *tracelevel,* std::string *text*) [inline]

5.98.2 Member Function Documentation

5.98.2.1 std::string TraceEvent::getText ( ) const [inline]

Here is the caller graph for this function:



5.98.2.2 Util::TraceLevel TraceEvent::getTracelevel ( ) const [inline]

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

- TraceManager.h

5.99 TraceManager Class Reference

```
#include <TraceManager.h>
```

Public Member Functions

- TraceManager (Simulator ∗simulator)
- TraceManager (const TraceManager &orig)
- virtual ∼TraceManager ()
- void addTraceHandler (traceListener traceListener)
- void addTraceErrorHandler (traceErrorListener traceErrorListener)
- void addTraceReportHandler (traceListener traceReportListener)
- void addTraceSimulationHandler (traceSimulationListener traceSimulationListener)
- void trace (Util::TraceLevel tracelevel, std::string text)
- void traceError (std::exception e, std::string text)
- void traceSimulation (Util::TraceLevel tracelevel, double time, Entity ∗entity, ModelComponent ∗component, std::string text)
- void traceReport (Util::TraceLevel tracelevel, std::string text)
- List< std::string > ∗ getErrorMessages () const
- void setTraceLevel (Util::TraceLevel _traceLevel)
- Util::TraceLevel getTraceLevel () const
- Simulator ∗ getSimulator () const

5.99.1 Detailed Description

The TraceManager is used to trace back model simulation information and track/debug the simulation. It works as the model simulation output (cout) and allows external methods to hook up such output as listeners.

5.99.2 Constructor & Destructor Documentation

5.99.2.1 TraceManager::TraceManager (**Simulator** ∗ *simulator*)

5.99.2.2 TraceManager::TraceManager (const **TraceManager** & *orig*)

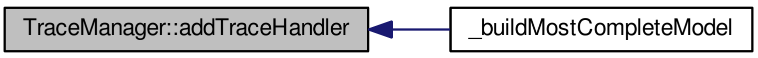
5.99.2.3 TraceManager::∼TraceManager ( ) [virtual]

5.99.3 Member Function Documentation

5.99.3.1 void TraceManager::addTraceErrorHandler (**traceErrorListener** *traceErrorListener*)

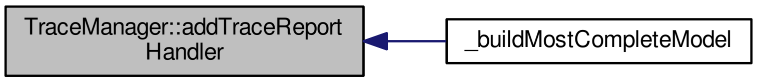
5.99.3.2 void TraceManager::addTraceHandler (**traceListener** *traceListener*)

Here is the caller graph for this function:



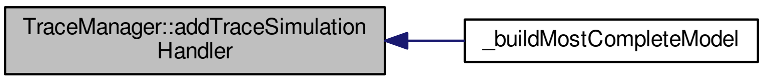
5.99.3.3 void TraceManager::addTraceReportHandler (traceListener *traceReportListener*)

Here is the caller graph for this function:



5.99.3.4 void TraceManager::addTraceSimulationHandler (traceSimulationListener *traceSimulationListener*)

Here is the caller graph for this function:



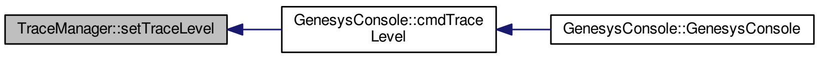
5.99.3.5 List< std::string > * TraceManager::getErrorMessages ( ) const

5.99.3.6 Simulator * TraceManager::getSimulator ( ) const

5.99.3.7 Util::TraceLevel TraceManager::getTraceLevel ( ) const

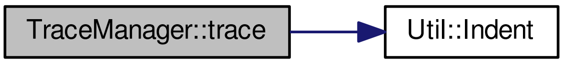
5.99.3.8 void TraceManager::setTraceLevel (Util::TraceLevel *_traceLevel*)

Here is the caller graph for this function:

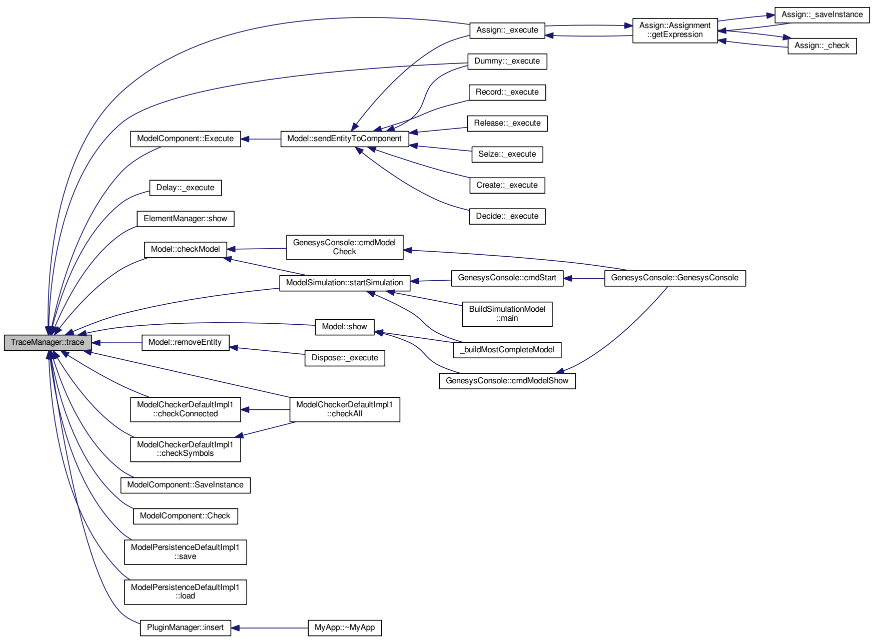

TraceManager::setTraceLevel
GenesysConsole::cmdTrace Level
GenesysConsole::GenesysConsole

5.99.3.9 void TraceManager::trace (Util::TraceLevel *tracelevel*, std::string *text*)

Here is the call graph for this function:

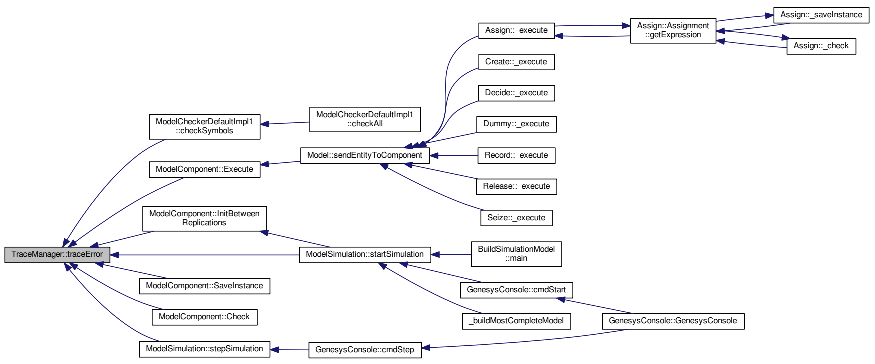


Here is the caller graph for this function:


TraceManager::trace
Assign::_execute
Assign::Assignment ::getExpression
Assign::_saveInstance
Assign::_check
Dummy::_execute
Record::_execute
Release::_execute
Seize::_execute
Create::_execute
Decide::_execute
ModelComponent::Execute
Model::sendEntityToComponent
Delay::_execute
ElementManager::show
Model::checkModel
GenesysConsole::cmdModel Check
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
BuildSimulationModel ::main
Model::show
_buildMostCompleteModel
Model::removeEntity
Dispose::_execute
GenesysConsole::cmdModelShow
ModelCheckerDefaultImpl1 ::checkConnected
ModelCheckerDefaultImpl1 ::checkAll
ModelCheckerDefaultImpl1 ::checkSymbols
ModelComponent::SaveInstance
ModelComponent::Check
ModelPersistenceDefaultImpl1 ::save
ModelPersistenceDefaultImpl1 ::load
PluginManager::insert
MyApp::~MyApp

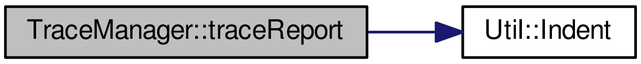
5.99.3.10 void TraceManager::traceError (std::exception *e*, std::string *text*)

Here is the caller graph for this function:

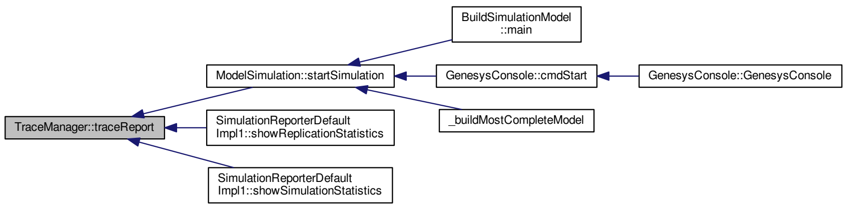

TraceManager::traceError
ModelCheckerDefaultImpl1 ::checkSymbols
ModelComponent::Execute
ModelComponent::InitBetween Replications
ModelSimulation::startSimulation
ModelComponent::SaveInstance
ModelComponent::Check
ModelSimulation::stepSimulation
ModelCheckerDefaultImpl1 ::checkAll
Model::sendEntityToComponent
GenesysConsole::cmdStep
Assign::_execute
Create::_execute
Decide::_execute
Dummy::_execute
Record::_execute
Release::_execute
Seize::_execute
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
Assign::Assignment ::getExpression
GenesysConsole::GenesysConsole
Assign::_saveInstance
Assign::_check

5.99.3.11 void TraceManager::traceReport (Util::TraceLevel *tracelevel*, std::string *text*)

Here is the call graph for this function:

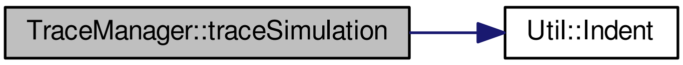


Here is the caller graph for this function:

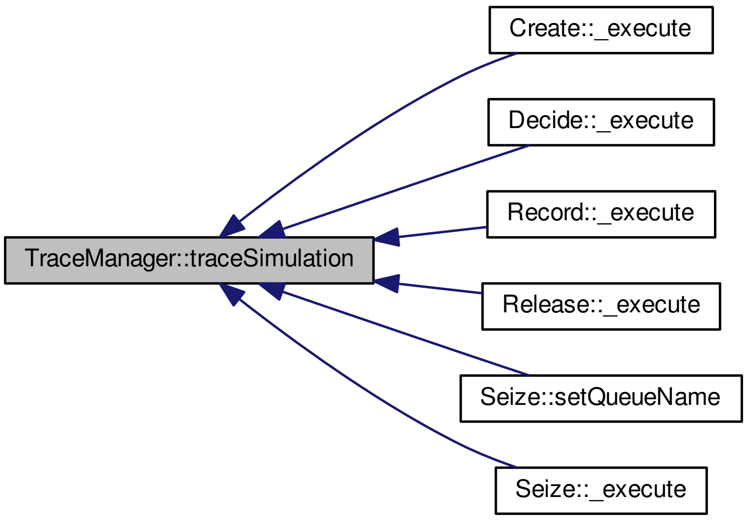

TraceManager::traceReport
ModelSimulation::startSimulation
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics
BuildSimulationModel ::main
GenesysConsole::cmdStart
_buildMostCompleteModel
GenesysConsole::GenesysConsole

5.99.3.12 void TraceManager::traceSimulation (Util::TraceLevel *tracelevel*, double *time*, Entity ∗ *entity*, ModelComponent ∗ *component*, std::string *text*)

Here is the call graph for this function:



Here is the caller graph for this function:



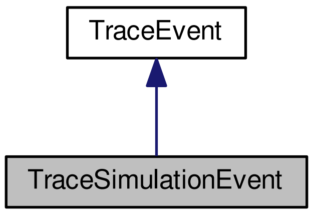
The documentation for this class was generated from the following files:

- TraceManager.h
- TraceManager.cpp

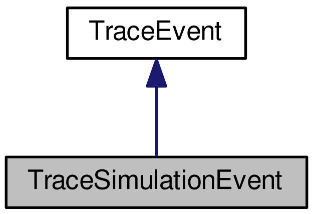
5.100 TraceSimulationEvent Class Reference

```
#include <TraceManager.h>
```

Inheritance diagram for TraceSimulationEvent:



Collaboration diagram for TraceSimulationEvent:



Public Member Functions

- ModelComponent ∗ getComponent () const
- Entity ∗ getEntity () const
- double getTime () const
- TraceSimulationEvent (Util::TraceLevel tracelevel, double time, Entity ∗entity, ModelComponent ∗component, std::string text)

5.100.1 Constructor & Destructor Documentation

5.100.1.1 TraceSimulationEvent::TraceSimulationEvent (**Util::TraceLevel** *tracelevel,* double *time,* **Entity** ∗ *entity,* **ModelComponent** ∗ *component,* std::string *text*) [inline]

5.100.2 Member Function Documentation

5.100.2.1 **ModelComponent**∗ TraceSimulationEvent::getComponent ( ) const [inline]

5.100.2.2 **Entity**∗ TraceSimulationEvent::getEntity ( ) const [inline]

5.100.2.3 double TraceSimulationEvent::getTime ( ) const [inline]

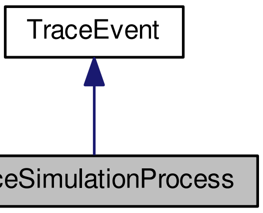
The documentation for this class was generated from the following file:

- TraceManager.h

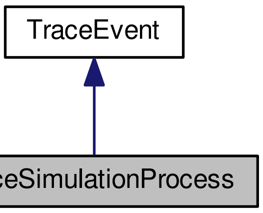
5.101 TraceSimulationProcess Class Reference

`#include <TraceManager.h>`

Inheritance diagram for TraceSimulationProcess:



Collaboration diagram for TraceSimulationProcess:



Public Member Functions

- TraceSimulationProcess (Util::TraceLevel tracelevel, std::string text)

5.101.1 Detailed Description

Events related to simulation "process" (usually process analyser), associated to entire replication or simulation events (begin/end/pause of replication/simulation) TODO: CLASS NOT COMPLETE

5.101.2 Constructor & Destructor Documentation

5.101.2.1 TraceSimulationProcess::TraceSimulationProcess (**Util::TraceLevel** *tracelevel,* std::string *text*) `[inline]`

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

- TraceManager.h

5.102 Traits< T > Struct Template Reference

```
#include <Traits.h>
```

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

- Traits.h

5.103 Traits< Collector_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef CollectorDatafileDefaultImpl1 Implementation

5.103.1 Member Typedef Documentation

5.103.1.1 typedef **CollectorDatafileDefaultImpl1 Traits< Collector_if >::Implementation**

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

- Traits.h

5.104 Traits< ExperimentDesign_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef ExperimentDesignDefaultImpl1 Implementation

5.104.1 Member Typedef Documentation

5.104.1.1 typedef **ExperimentDesignDefaultImpl1 Traits< ExperimentDesign_if >::Implementation**

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

- Traits.h

5.105 Traits< Fitter_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef FitterDefaultImpl1 Implementation

5.105.1 Member Typedef Documentation

5.105.1.1 typedef **FitterDefaultImpl1 Traits< Fitter_if >::Implementation**

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

- Traits.h

5.106 Traits< GenesysApplication_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef MyApp Application

5.106.1 Member Typedef Documentation

5.106.1.1 typedef **MyApp Traits< GenesysApplication_if >::Application**

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

- Traits.h

5.107 Traits< HypothesisTester_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef IntegratorDefaultImpl1 IntegratorImplementation
- typedef HypothesisTesterDefaultImpl1 Implementation

5.107.1 Member Typedef Documentation

5.107.1.1 typedef HypothesisTesterDefaultImpl1 Traits< HypothesisTester_if >::Implementation

5.107.1.2 typedef IntegratorDefaultImpl1 Traits< HypothesisTester_if >::IntegratorImplementation

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

- Traits.h

5.108 Traits< Integrator_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef IntegratorDefaultImpl1 Implementation

Static Public Attributes

- static constexpr unsigned int MaxIterations = 1e3
- static constexpr double Precision = 1e-9

5.108.1 Member Typedef Documentation

5.108.1.1 typedef IntegratorDefaultImpl1 Traits< Integrator_if >::Implementation

5.108.2 Member Data Documentation

5.108.2.1 constexpr unsigned int Traits< Integrator_if >::MaxIterations = 1e3 `[static]`

5.108.2.2 constexpr double Traits< Integrator_if >::Precision = 1e-9 `[static]`

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

- Traits.h

5.109 Traits< Model > Struct Template Reference

```
#include <Traits.h>
```

Static Public Attributes

- static const bool debugged = true
- static const Util::TraceLevel traceLevel = Util::TraceLevel::blockArrival

5.109.1 Member Data Documentation

5.109.1.1 const bool Traits< Model >::debugged = true [static]

5.109.1.2 const Util::TraceLevel Traits< Model >::traceLevel = Util::TraceLevel::blockArrival [static]

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

- Traits.h

5.110 Traits< ModelChecker_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef ModelCheckerDefaultImpl1 Implementation

5.110.1 Member Typedef Documentation

5.110.1.1 typedef ModelCheckerDefaultImpl1 Traits< ModelChecker_if >::Implementation

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

- Traits.h

5.111 Traits< ModelComponent > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef StatisticsDefaultImpl1 StatisticsCollector_StatisticsImplementation
- typedef CollectorDefaultImpl1 StatisticsCollector_CollectorImplementation

5.111.1 Member Typedef Documentation

5.111.1.1 typedef CollectorDefaultImpl1 Traits< ModelComponent >::StatisticsCollector_Collector↩ Implementation

5.111.1.2 typedef StatisticsDefaultImpl1 Traits< ModelComponent >::StatisticsCollector_Statistics↩ Implementation

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

- Traits.h

5.112 Traits< ModelPersistence_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef ModelPersistenceDefaultImpl1 Implementation

5.112.1 Member Typedef Documentation

5.112.1.1 typedef ModelPersistenceDefaultImpl1 Traits< ModelPersistence_if >::Implementation

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

- Traits.h

5.113 Traits< Parser_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef ParserDefaultImpl2 Implementation

5.113.1 Member Typedef Documentation

5.113.1.1 typedef ParserDefaultImpl2 Traits< Parser_if >::Implementation

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

- Traits.h

5.114 Traits< ProcessAnalyser_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef ProcessAnalyserDefaultImpl1 Implementation

5.114.1 Member Typedef Documentation

5.114.1.1 typedef **ProcessAnalyserDefaultImpl1 Traits< ProcessAnalyser_if >::Implementation**

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

- Traits.h

5.115 Traits< Sampler_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef SamplerDefaultImpl1 Implementation
- typedef SamplerDefaultImpl1::DefaultImpl1RNG_Parameters Parameters

5.115.1 Member Typedef Documentation

5.115.1.1 typedef **SamplerDefaultImpl1 Traits< Sampler_if >::Implementation**

5.115.1.2 typedef **SamplerDefaultImpl1::DefaultImpl1RNG_Parameters Traits< Sampler_if >::Parameters**

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

- Traits.h

5.116 Traits< SimulationReporter_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef SimulationReporterDefaultImpl1 Implementation

5.116.1 Member Typedef Documentation

5.116.1.1 typedef SimulationReporterDefaultImpl1 Traits< SimulationReporter_if >::Implementation

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

- Traits.h

5.117 Traits< Statistics_if > Struct Template Reference

```
#include <Traits.h>
```

Public Types

- typedef StatisticsDefaultImpl1 Implementation
- typedef CollectorDefaultImpl1 CollectorImplementation

Static Public Attributes

- static constexpr double SignificanceLevel = 0.05

5.117.1 Member Typedef Documentation

5.117.1.1 typedef CollectorDefaultImpl1 Traits< Statistics_if >::CollectorImplementation

5.117.1.2 typedef StatisticsDefaultImpl1 Traits< Statistics_if >::Implementation

5.117.2 Member Data Documentation

5.117.2.1 constexpr double Traits< Statistics_if >::SignificanceLevel = 0.05 [static]

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

- Traits.h

5.118 Util Class Reference

```
#include <Util.h>
```

Public Types

- enum TimeUnit : int {
  TimeUnit::picosecond = 1, TimeUnit::nanosecond = 2, TimeUnit::microsecond = 3, TimeUnit::milisecond = 4,
  TimeUnit::second = 5, TimeUnit::minute = 6, TimeUnit::hour = 7, TimeUnit::day = 8,
  TimeUnit::week = 9 }
- enum TraceLevel : int {
  TraceLevel::noTraces = 0, TraceLevel::errors = 1, TraceLevel::report = 2, TraceLevel::simulation = 3,
  TraceLevel::transferOnly = 4, TraceLevel::blockArrival = 5, TraceLevel::blockInternal = 6, TraceLevel::most←
  Detailed = 7 }
- typedef unsigned long identitifcation
- typedef unsigned int rank

Static Public Member Functions

- static void SetIndent (const unsigned short indent)
- static void IncIndent ()
- static void DecIndent ()
- static void SepKeyVal (std::string str, std::string ∗key, std::string ∗value)
- static std::string Indent ()
- static std::string SetW (std::string text, unsigned short width)
- static std::string StrTimeUnit (Util::TimeUnit timeUnit)
- static Util::identitifcation GenerateNewId ()
- static Util::identitifcation GenerateNewIdOfType (std::string objtype)
- static Util::identitifcation GetLastIdOfType (std::string objtype)
- static void ResetIdOfType (std::string objtype)
- static double TimeUnitConvert (Util::TimeUnit timeUnit1, Util::TimeUnit timeUnit2)
- template<class T >
  static std::string TypeOf ()
- template<class T >
  static Util::identitifcation GenerateNewIdOfType ()

Static Public Attributes

- static unsigned int _S_indentation

5.118.1 Member Typedef Documentation

5.118.1.1 typedef unsigned long **Util::identitifcation**

5.118.1.2 typedef unsigned int **Util::rank**

5.118.2 Member Enumeration Documentation

5.118.2.1 enum **Util::TimeUnit** : int `[strong]`

Enumerator

picosecond

nanosecond

microsecond

milisecond

second

minute

hour

day

week

5.118.2.2 enum Util::TraceLevel : int [strong]

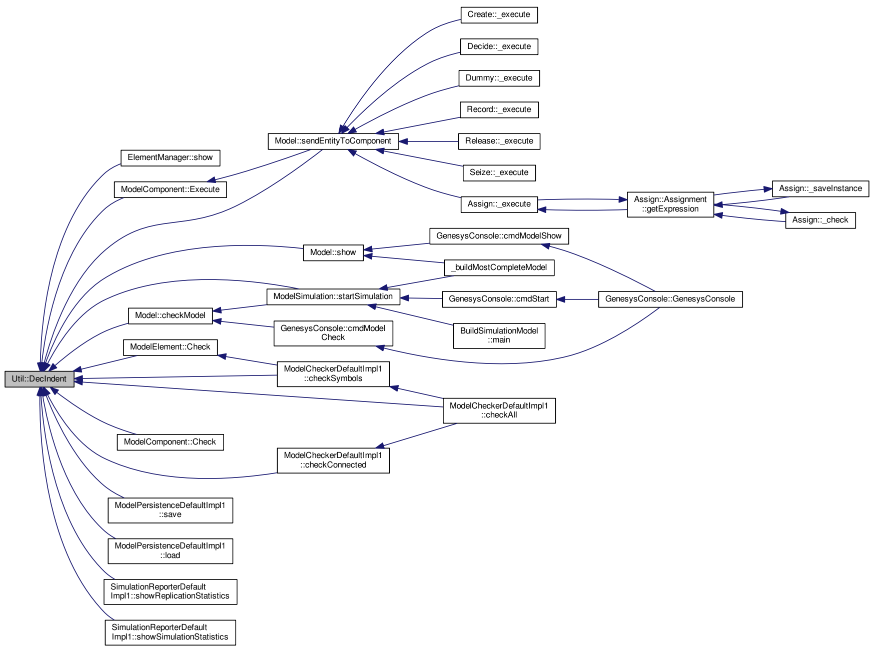
Enumerator

- ***noTraces***
- ***errors***
- ***report***
- ***simulation***
- ***transferOnly***
- ***blockArrival***
- ***blockInternal***
- ***mostDetailed***

5.118.3 Member Function Documentation

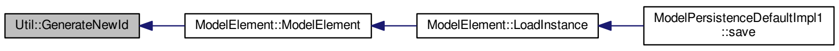
5.118.3.1 void Util::DecIndent ( ) [static]

Here is the caller graph for this function:


Create::_execute
Decide::_execute
Dummy::_execute
Record::_execute
Model::sendEntityToComponent
Release::_execute
ElementManager::show
Seize::_execute
ModelComponent::Execute
Assign::_saveInstance
Assign::_execute
Assign::Assignment ::getExpression
Assign::_check
GenesysConsole::cmdModelShow
Model::show
_buildMostCompleteModel
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
Model::checkModel
GenesysConsole::cmdModel Check
BuildSimulationModel ::main
ModelElement::Check
ModelCheckerDefaultImpl1 ::checkSymbols
Util::DecIndent
ModelCheckerDefaultImpl1 ::checkAll
ModelComponent::Check
ModelCheckerDefaultImpl1 ::checkConnected
ModelPersistenceDefaultImpl1 ::save
ModelPersistenceDefaultImpl1 ::load
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics

5.118.3.2 Util::identitifcation Util::GenerateNewId ( ) [static]

Here is the caller graph for this function:

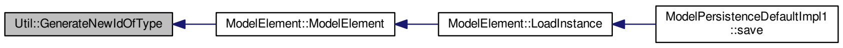

Util::GenerateNewId
ModelElement::ModelElement
ModelElement::LoadInstance
ModelPersistenceDefaultImpl1 ::save

5.118.3.3 Util::identitifcation Util::GenerateNewIdOfType (std::string *objtype*) [static]

5.118.3.4 template<class T > static Util::identitifcation Util::GenerateNewIdOfType ( ) [inline], [static]

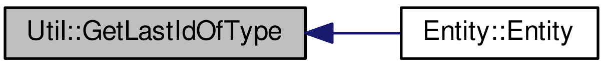
Every component or element has a unique ID for its class, but not unique for other classes. IDs are generated sequentially for each class.

Here is the caller graph for this function:


Util::GenerateNewIdOfType
ModelElement::ModelElement
ModelElement::LoadInstance
ModelPersistenceDefaultImpl1 ::save

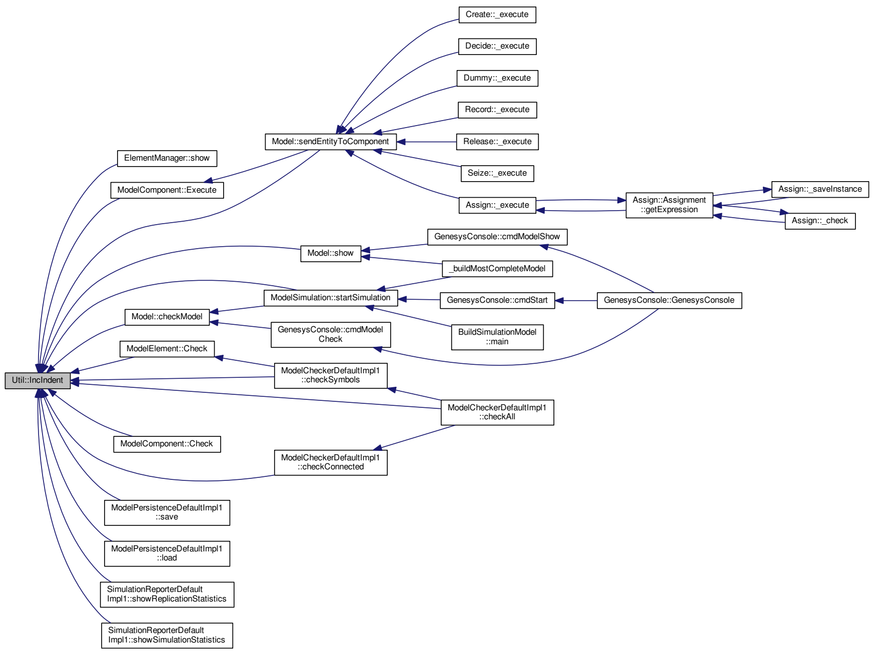
5.118.3.5 Util::identitifcation Util::GetLastIdOfType (std::string *objtype*) [static]

Here is the caller graph for this function:



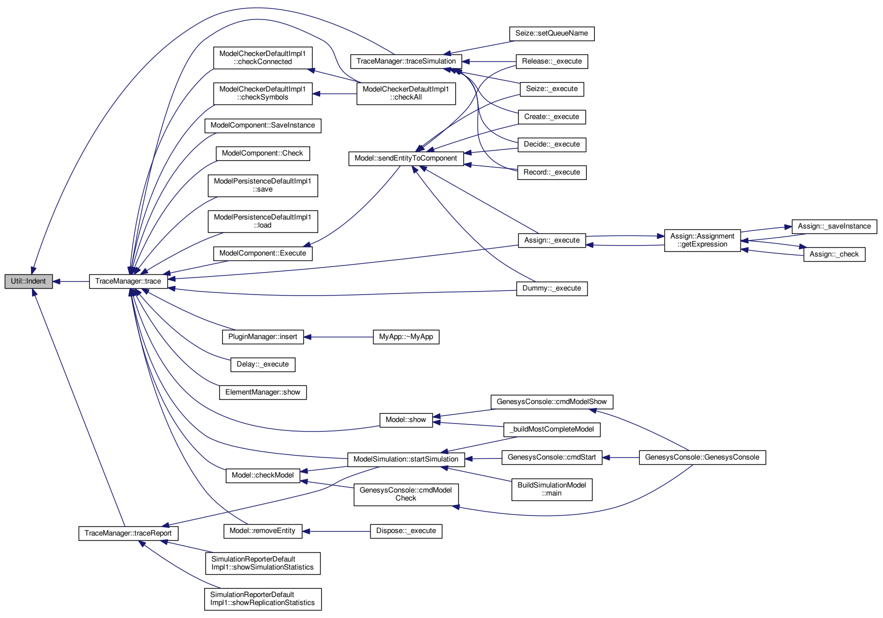
5.118.3.6 void Util::IncIndent ( ) [static]

Here is the caller graph for this function:


Create::_execute
Decide::_execute
Dummy::_execute
Record::_execute
Release::_execute
Seize::_execute
Assign::_execute
Model::sendEntityToComponent
ElementManager::show
ModelComponent::Execute
Assign::Assignment ::getExpression
Assign::_saveInstance
Assign::_check
GenesysConsole::cmdModelShow
Model::show
_buildMostCompleteModel
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
Model::checkModel
GenesysConsole::cmdModel Check
BuildSimulationModel ::main
ModelElement::Check
ModelCheckerDefaultImpl1 ::checkSymbols
Util::IncIndent
ModelCheckerDefaultImpl1 ::checkAll
ModelComponent::Check
ModelCheckerDefaultImpl1 ::checkConnected
ModelPersistenceDefaultImpl1 ::save
ModelPersistenceDefaultImpl1 ::load
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics

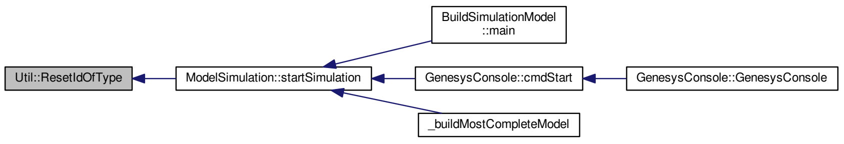
5.118.3.7 std::string Util::Indent ( ) [static]

Here is the caller graph for this function:


Util::Indent
TraceManager::trace
TraceManager::traceSimulation
Seize::setQueueName
Release::_execute
Seize::_execute
Create::_execute
Decide::_execute
Record::_execute
ModelCheckerDefaultImpl1 ::checkConnected
ModelCheckerDefaultImpl1 ::checkSymbols
ModelCheckerDefaultImpl1 ::checkAll
ModelComponent::SaveInstance
ModelComponent::Check
ModelPersistenceDefaultImpl1 ::save
ModelPersistenceDefaultImpl1 ::load
ModelComponent::Execute
Model::sendEntityToComponent
Assign::_execute
Assign::Assignment ::getExpression
Assign::_saveInstance
Assign::_check
Dummy::_execute
PluginManager::insert
MyApp::~MyApp
Delay::_execute
ElementManager::show
Model::show
GenesysConsole::cmdModelShow
_buildMostCompleteModel
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
BuildSimulationModel ::main
Model::checkModel
GenesysConsole::cmdModel Check
TraceManager::traceReport
Model::removeEntity
Dispose::_execute
SimulationReporterDefault Impl1::showSimulationStatistics
SimulationReporterDefault Impl1::showReplicationStatistics

5.118.3.8 void Util::ResetIdOfType (std::string *objtype*) [static]

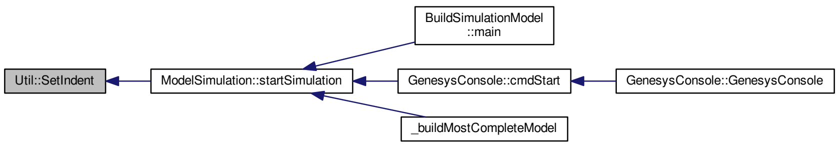
Here is the caller graph for this function:


BuildSimulationModel ::main
Util::ResetIdOfType
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.118.3.9 void Util::SepKeyVal (std::string *str,* std::string ∗ *key,* std::string ∗ *value*) [static]

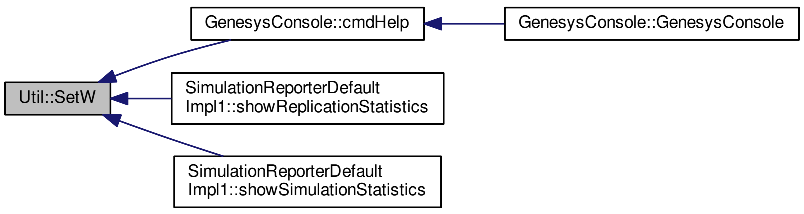
5.118.3.10 void Util::SetIndent (const unsigned short *indent*) [static]

Here is the caller graph for this function:


BuildSimulationModel ::main
Util::SetIndent
ModelSimulation::startSimulation
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

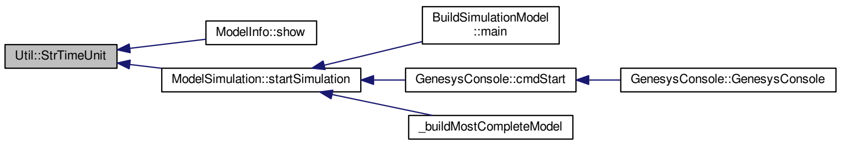
5.118.3.11 std::string Util::SetW (std::string *text*, unsigned short *width*) [static]

Here is the caller graph for this function:


Util::SetW
GenesysConsole::cmdHelp
GenesysConsole::GenesysConsole
SimulationReporterDefault Impl1::showReplicationStatistics
SimulationReporterDefault Impl1::showSimulationStatistics

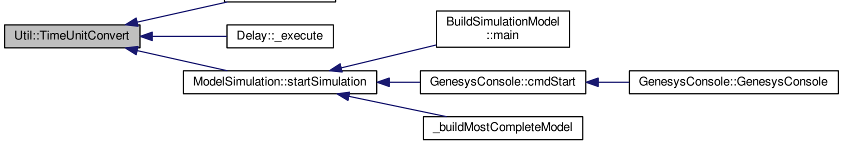
5.118.3.12 std::string Util::StrTimeUnit (Util::TimeUnit *timeUnit*) [static]

Here is the caller graph for this function:


Util::StrTimeUnit
ModelInfo::show
ModelSimulation::startSimulation
BuildSimulationModel ::main
GenesysConsole::cmdStart
GenesysConsole::GenesysConsole
_buildMostCompleteModel

5.118.3.13 double Util::TimeUnitConvert (Util::TimeUnit *timeUnit1*, Util::TimeUnit *timeUnit2*) [static]

Here is the caller graph for this function:



5.118.3.14 template<class T > static std::string Util::TypeOf ( ) [inline], [static]

Return the name of the class used as T.

5.118.4 Member Data Documentation

5.118.4.1 unsigned int Util::_S_indentation [static]

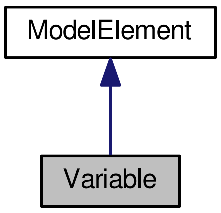
The documentation for this class was generated from the following files:

- Util.h
- Util.cpp

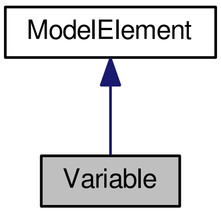
5.119 Variable Class Reference

`#include <Variable.h>`

Inheritance diagram for Variable:



Collaboration diagram for Variable:



Public Member Functions

- Variable ()
- Variable (std::string name)
- Variable (const Variable &orig)
- virtual ∼Variable ()
- virtual std::string show ()
- double getValue ()
- double getValue (std::string index)
- void setValue (double value)
- void setValue (std::string index, double value)

Static Public Member Functions

- static PluginInformation ∗ GetPluginInformation ()
- static ModelElement ∗ LoadInstance (ElementManager ∗elems, std::map< std::string, std::string > ∗fields)

Protected Member Functions

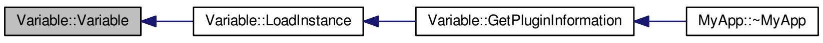
- virtual bool _loadInstance (std::map< std::string, std::string > ∗fields)
- virtual std::map< std::string, std::string > ∗ _saveInstance ()
- virtual bool _check (std::string ∗errorMessage)

Additional Inherited Members

5.119.1 Constructor & Destructor Documentation

5.119.1.1 Variable::Variable ( )

Here is the caller graph for this function:



5.119.1.2 Variable::Variable (std::string *name*)

5.119.1.3 Variable::Variable (const Variable & *orig*)

5.119.1.4 Variable::∼Variable ( ) [virtual]

5.119.2 Member Function Documentation

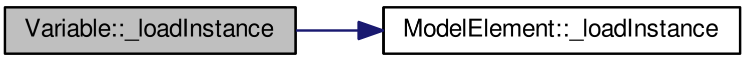
5.119.2.1 bool Variable::_check (std::string ∗ *errorMessage*) [protected], [virtual]

Reimplemented from ModelElement.

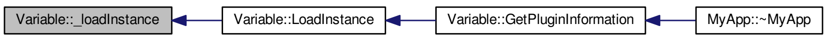
5.119.2.2 bool Variable::_loadInstance (std::map< std::string, std::string > * *fields*) [protected], [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



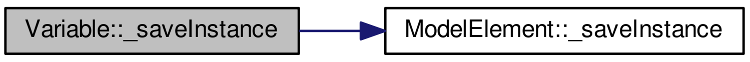
Here is the caller graph for this function:



5.119.2.3 std::map< std::string, std::string > * Variable::_saveInstance ( ) [protected], [virtual]

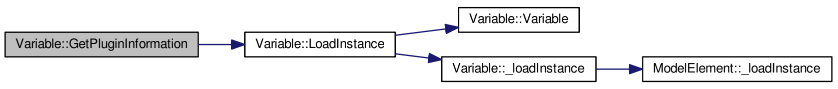
Reimplemented from ModelElement.

Here is the call graph for this function:

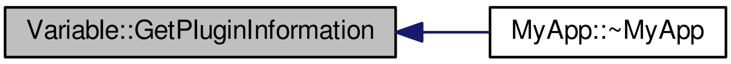


5.119.2.4 PluginInformation * Variable::GetPluginInformation ( ) [static]

Here is the call graph for this function:



Here is the caller graph for this function:

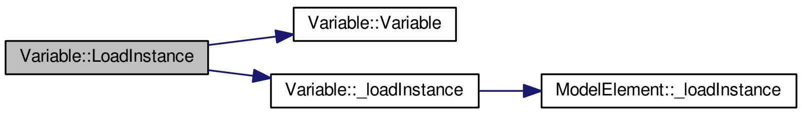


5.119.2.5 double Variable::getValue ( )

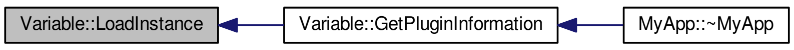
5.119.2.6 double Variable::getValue (std::string *index*)

5.119.2.7 **ModelElement** $*$ Variable::LoadInstance (**ElementManager** $*$ *elems,* std::map$<$ std::string, std::string $>$ $*$ *fields*) [static]

Here is the call graph for this function:

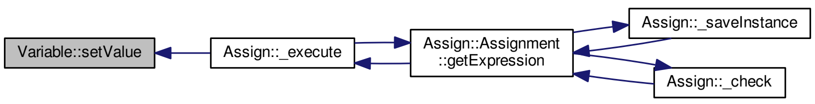


Here is the caller graph for this function:



5.119.2.8 void Variable::setValue (double *value*)

Here is the caller graph for this function:

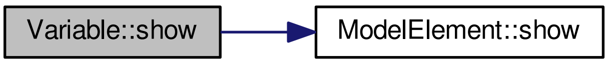


5.119.2.9 void Variable::setValue (std::string *index*, double *value*)

5.119.2.10 std::string Variable::show ( ) [virtual]

Reimplemented from ModelElement.

Here is the call graph for this function:



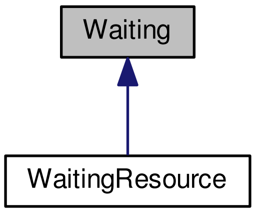
The documentation for this class was generated from the following files:

- Variable.h
- Variable.cpp

5.120 Waiting Class Reference

`#include <Waiting.h>`

Inheritance diagram for Waiting:



Public Member Functions

- Waiting (Entity ∗entity, ModelComponent ∗component, double timeStartedWaiting)
- Waiting (const Waiting &orig)
- virtual ∼Waiting ()
- virtual std::string show ()
- double getTimeStartedWaiting () const
- ModelComponent ∗ getComponent () const
- Entity ∗ getEntity () const

5.120.1 Constructor & Destructor Documentation

5.120.1.1 Waiting::Waiting (**Entity** ∗ *entity*, **ModelComponent** ∗ *component*, double *timeStartedWaiting*)

5.120.1.2 Waiting::Waiting (const **Waiting** & *orig*)

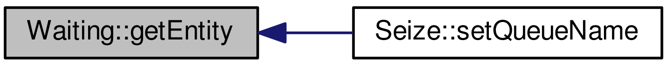
5.120.1.3 Waiting::∼Waiting ( ) `[virtual]`

5.120.2 Member Function Documentation

5.120.2.1 **ModelComponent** ∗ Waiting::getComponent ( ) const

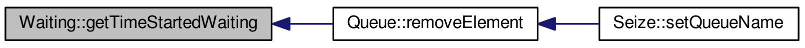
5.120.2.2 **Entity** ∗ Waiting::getEntity ( ) const

Here is the caller graph for this function:



5.120.2.3 double Waiting::getTimeStartedWaiting ( ) const

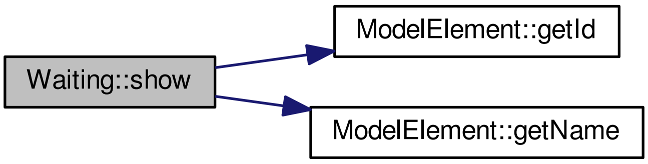
Here is the caller graph for this function:



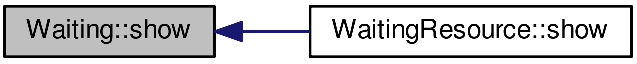
5.120.2.4 std::string Waiting::show ( ) `[virtual]`

Reimplemented in WaitingResource.

Here is the call graph for this function:



Here is the caller graph for this function:



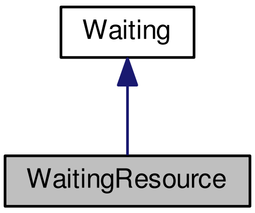
The documentation for this class was generated from the following files:

- Waiting.h
- Waiting.cpp

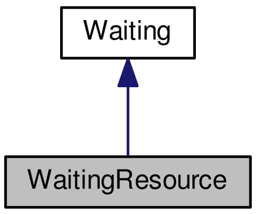
5.121 WaitingResource Class Reference

```
#include <WaitingResource.h>
```

Inheritance diagram for WaitingResource:



Collaboration diagram for WaitingResource:



Public Member Functions

- WaitingResource (Entity ∗entity, ModelComponent ∗component, double timeStartedWaiting, unsigned int quantity)
- WaitingResource (const WaitingResource &orig)
- virtual ∼WaitingResource ()
- virtual std::string show ()
- unsigned int getQuantity () const

5.121.1 Constructor & Destructor Documentation

5.121.1.1 WaitingResource::WaitingResource (**Entity** ∗ *entity,* **ModelComponent** ∗ *component,* double *timeStartedWaiting,* unsigned int *quantity*)

5.121.1.2 WaitingResource::WaitingResource (const **WaitingResource** & *orig*)

5.121.1.3 WaitingResource::∼WaitingResource ( ) `[virtual]`

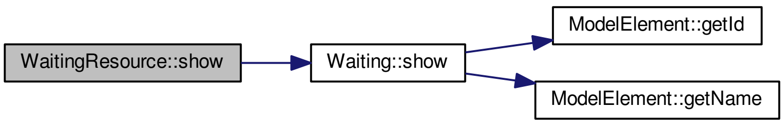
5.121.2 Member Function Documentation

5.121.2.1 unsigned int WaitingResource::getQuantity ( ) const

5.121.2.2 std::string WaitingResource::show ( ) `[virtual]`

Reimplemented from Waiting.

Here is the call graph for this function:



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

- WaitingResource.h
- WaitingResource.cpp

Chapter 6

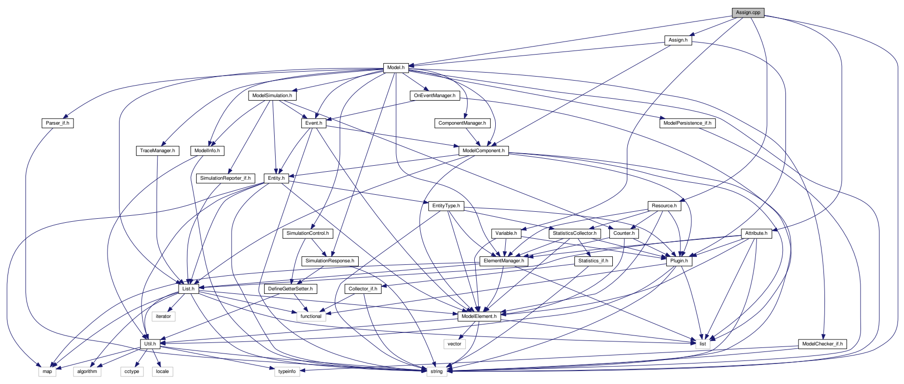
File Documentation

6.1 .dep.inc File Reference

6.2 Assign.cpp File Reference

```
#include "Assign.h"
#include <string>
#include "Model.h"
#include "Variable.h"
#include "Attribute.h"
#include "Resource.h"
```

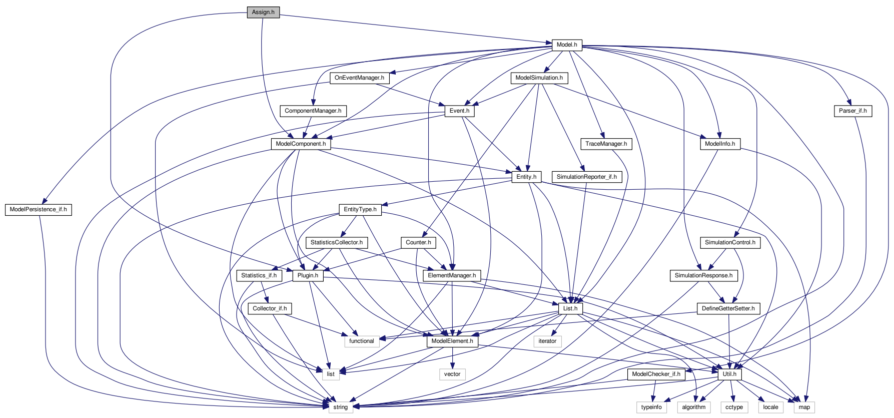
Include dependency graph for Assign.cpp:



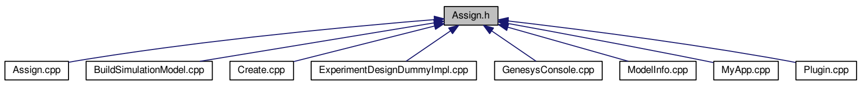
6.3 Assign.h File Reference

```
#include "ModelComponent.h"
#include "Model.h"
#include "Plugin.h"
```

Include dependency graph for Assign.h:



This graph shows which files directly or indirectly include this file:



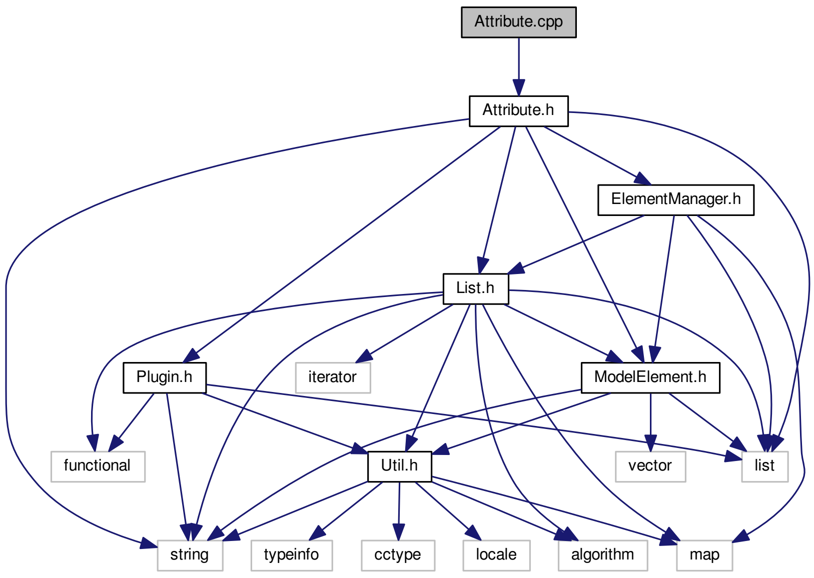
Classes

- class Assign
- class Assign::Assignment

6.4 Attribute.cpp File Reference

```
#include "Attribute.h"
```

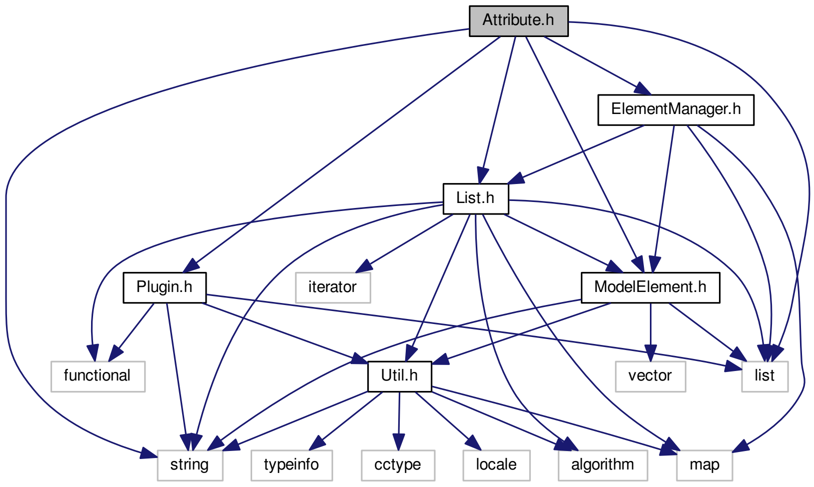
Include dependency graph for Attribute.cpp:



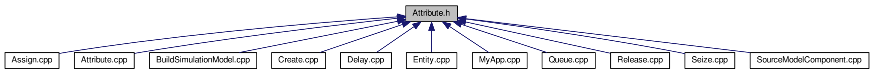
6.5 Attribute.h File Reference

```
#include <string>
#include <list>
#include "List.h"
#include "ModelElement.h"
#include "ElementManager.h"
#include "Plugin.h"
```

Include dependency graph for Attribute.h:



This graph shows which files directly or indirectly include this file:



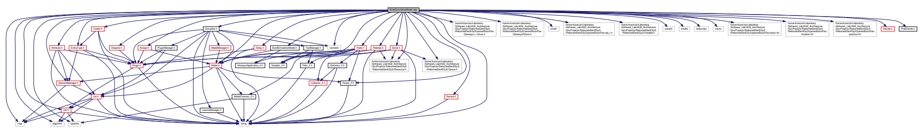
Classes

- class Attribute

6.6 BuildSimulationModel.cpp File Reference

```
#include "BuildSimulationModel.h"
#include "Simulator.h"
#include "Traits.h"
#include "Create.h"
#include "Delay.h"
#include "Dispose.h"
#include "Seize.h"
#include "Release.h"
#include "Assign.h"
#include "Record.h"
#include "Decide.h"
#include "ElementManager.h"
#include "EntityType.h"
#include "Attribute.h"
#include "ProbDistrib.h"
```

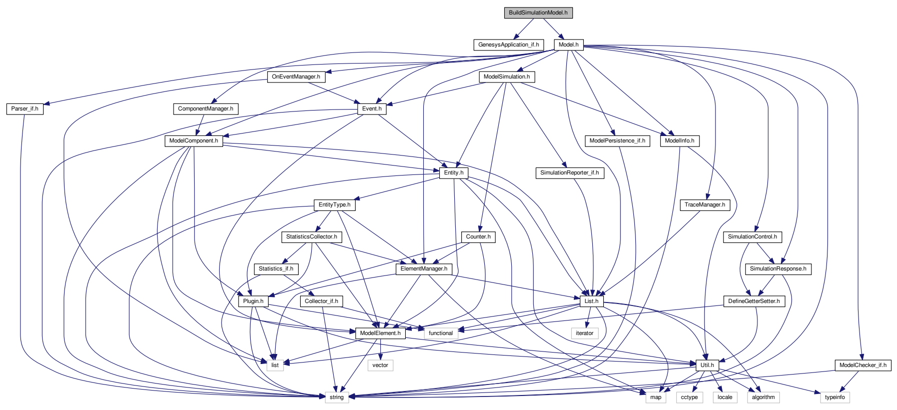
Include dependency graph for BuildSimulationModel.cpp:



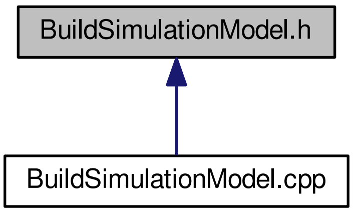
6.7 BuildSimulationModel.h File Reference

```
#include "GenesysApplication_if.h"
#include "Model.h"
```

Include dependency graph for BuildSimulationModel.h:



This graph shows which files directly or indirectly include this file:



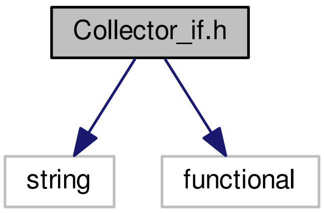
Classes

- class BuildSimulationModel

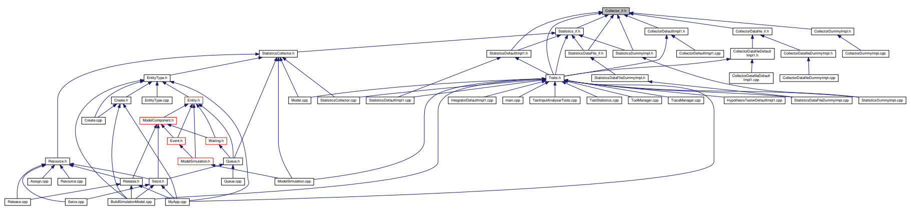
6.8 Collector_if.h File Reference

```
#include <string>
#include <functional>
```

Include dependency graph for Collector_if.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Collector_if

Typedefs

- typedef std::function< void(double) > CollectorAddValueHandler
- typedef std::function< void() > CollectorClearHandler

Functions

- template<typename Class >
  CollectorAddValueHandler SetCollectorAddValueHandler (void(Class::∗function)(double), Class ∗object)
- template<typename Class >
  CollectorClearHandler SetCollectorClearHandler (void(Class::∗function)(), Class ∗object)

6.8.1 Typedef Documentation

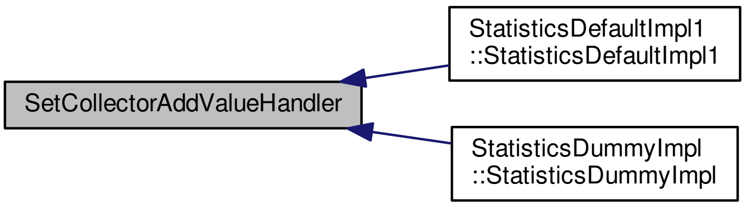
6.8.1.1 typedef std::function< void(double) > CollectorAddValueHandler

6.8.1.2 typedef std::function< void() > CollectorClearHandler

6.8.2 Function Documentation

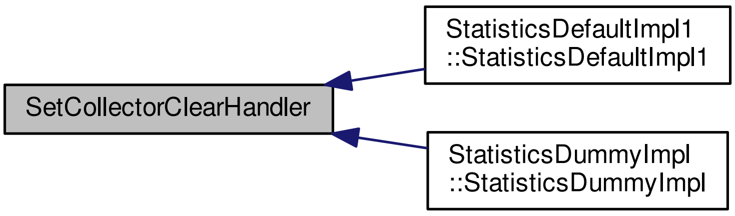
6.8.2.1 template<typename Class > CollectorAddValueHandler SetCollectorAddValueHandler (void(Class::∗)(double) *function,* Class ∗ *object*)

Here is the caller graph for this function:


SetCollectorAddValueHandler
StatisticsDefaultImpl1::StatisticsDefaultImpl1
StatisticsDummyImpl::StatisticsDummyImpl

6.8.2.2 template<typename Class > CollectorClearHandler SetCollectorClearHandler (void(Class::∗)() *function,* Class ∗ *object*)

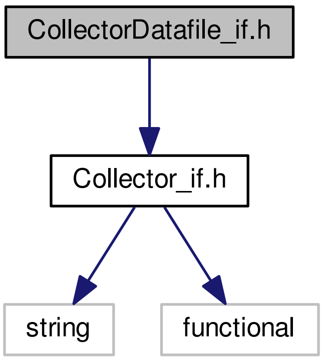
Here is the caller graph for this function:


SetCollectorClearHandler
StatisticsDefaultImpl1::StatisticsDefaultImpl1
StatisticsDummyImpl::StatisticsDummyImpl

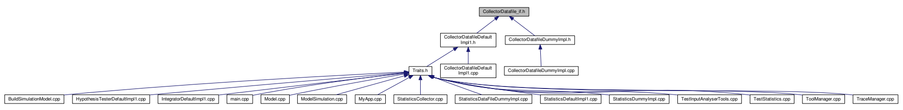
6.9 CollectorDatafile_if.h File Reference

```
#include "Collector_if.h"
```

Include dependency graph for CollectorDatafile_if.h:



This graph shows which files directly or indirectly include this file:


CollectorDatafile_if.h
CollectorDatafileDefault Impl1.h
CollectorDatafileDummyImpl.h
Traits.h
CollectorDatafileDefault Impl1.cpp
CollectorDatafileDummyImpl.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

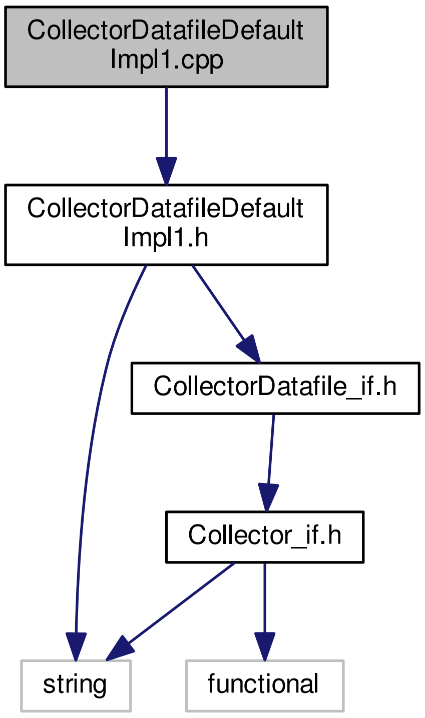
Classes

- class CollectorDatafile_if

6.10 CollectorDatafileDefaultImpl1.cpp File Reference

```
#include "CollectorDatafileDefaultImpl1.h"
```

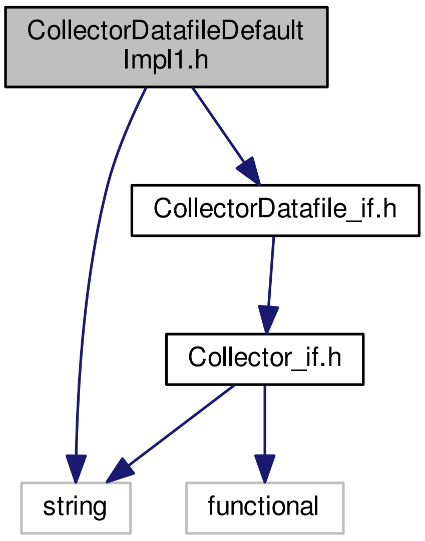
Include dependency graph for CollectorDatafileDefaultImpl1.cpp:



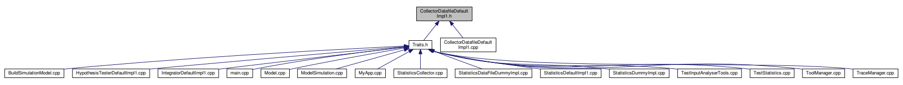
6.11 CollectorDatafileDefaultImpl1.h File Reference

```
#include <string>
#include "CollectorDatafile_if.h"
```

Include dependency graph for CollectorDatafileDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:


CollectorDatafileDefault Impl1.h
Traits.h
CollectorDatafileDefault Impl1.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

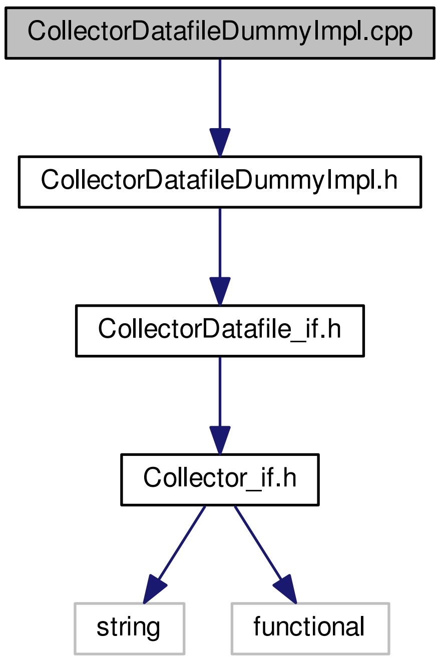
Classes

- class CollectorDatafileDefaultImpl1

6.12 CollectorDatafileDummyImpl.cpp File Reference

```
#include "CollectorDatafileDummyImpl.h"
```

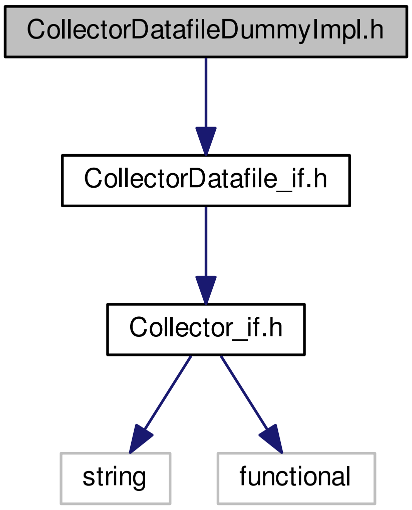
Include dependency graph for CollectorDatafileDummyImpl.cpp:



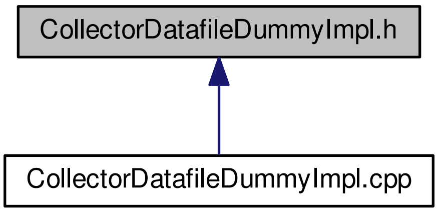
6.13 CollectorDatafileDummyImpl.h File Reference

```
#include "CollectorDatafile_if.h"
```

Include dependency graph for CollectorDatafileDummyImpl.h:



This graph shows which files directly or indirectly include this file:



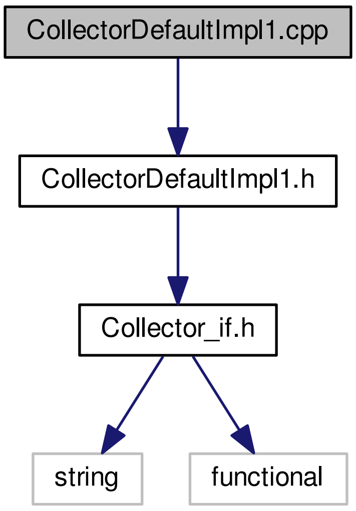
Classes

- class CollectorDatafileDummyImpl

6.14 CollectorDefaultImpl1.cpp File Reference

```
#include "CollectorDefaultImpl1.h"
```

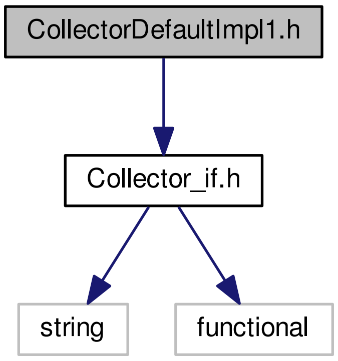
Include dependency graph for CollectorDefaultImpl1.cpp:



6.15 CollectorDefaultImpl1.h File Reference

```
#include "Collector_if.h"
```

Include dependency graph for CollectorDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:

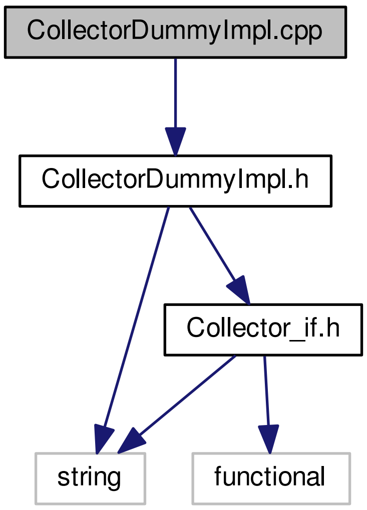
Classes

- class CollectorDefaultImpl1

6.16 CollectorDummyImpl.cpp File Reference

```
#include "CollectorDummyImpl.h"
```

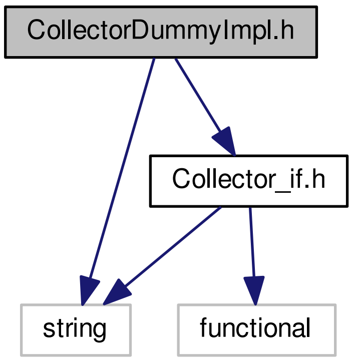
Include dependency graph for CollectorDummyImpl.cpp:



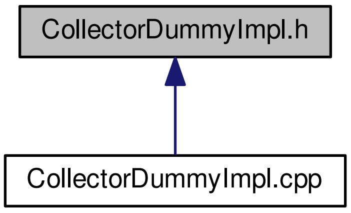
6.17 CollectorDummyImpl.h File Reference

```
#include <string>
#include "Collector_if.h"
```

Include dependency graph for CollectorDummyImpl.h:



This graph shows which files directly or indirectly include this file:



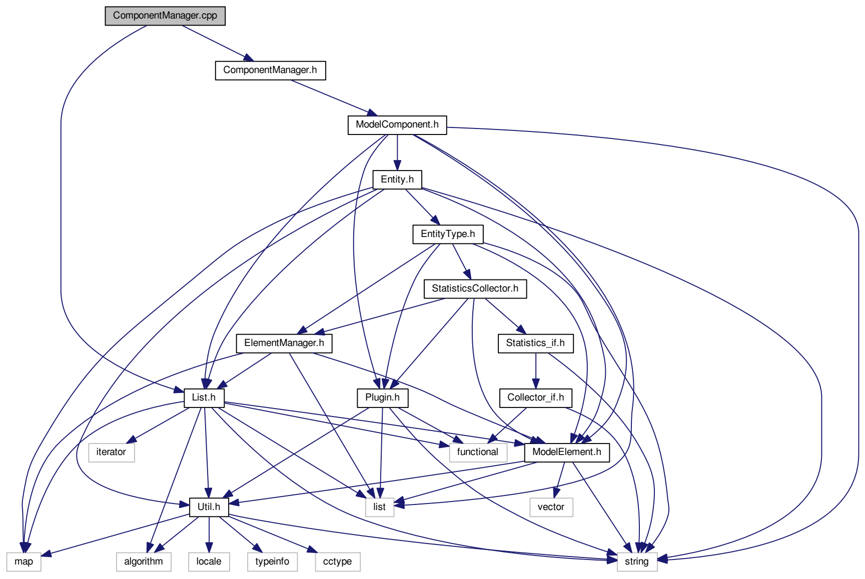
Classes

- class CollectorDummyImpl

6.18 ComponentManager.cpp File Reference

```
#include "ComponentManager.h"
#include "List.h"
```

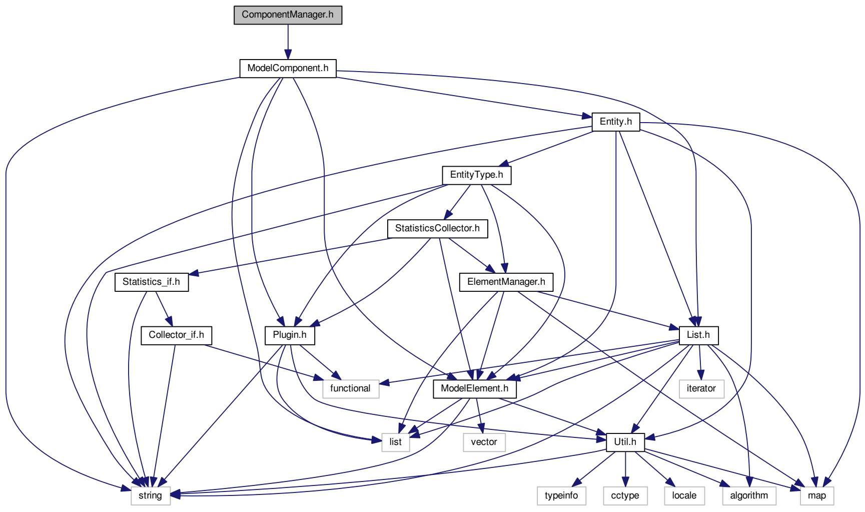
Include dependency graph for ComponentManager.cpp:



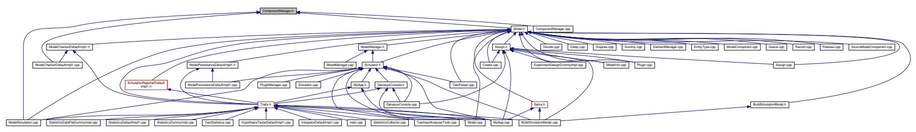
6.19 ComponentManager.h File Reference

```
#include "ModelComponent.h"
```

Include dependency graph for ComponentManager.h:



This graph shows which files directly or indirectly include this file:



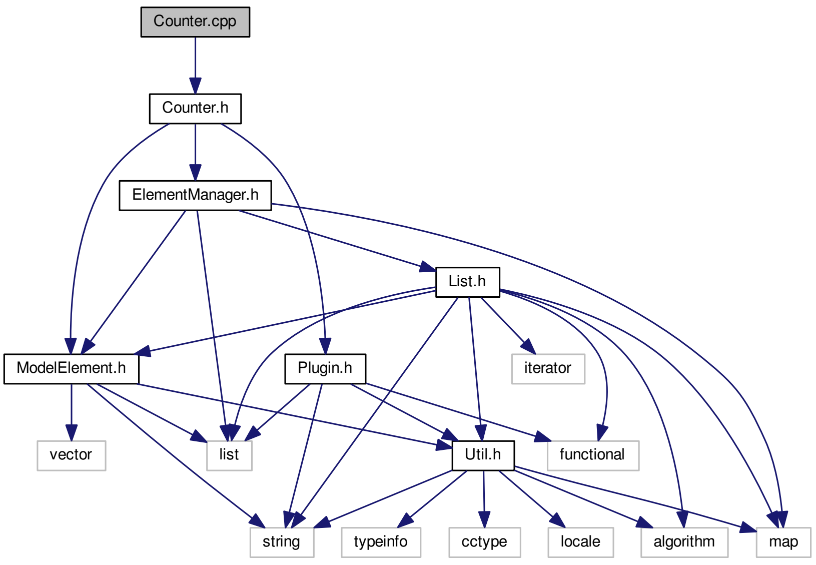
Classes

- class ComponentManager

6.20 Counter.cpp File Reference

```
#include "Counter.h"
```

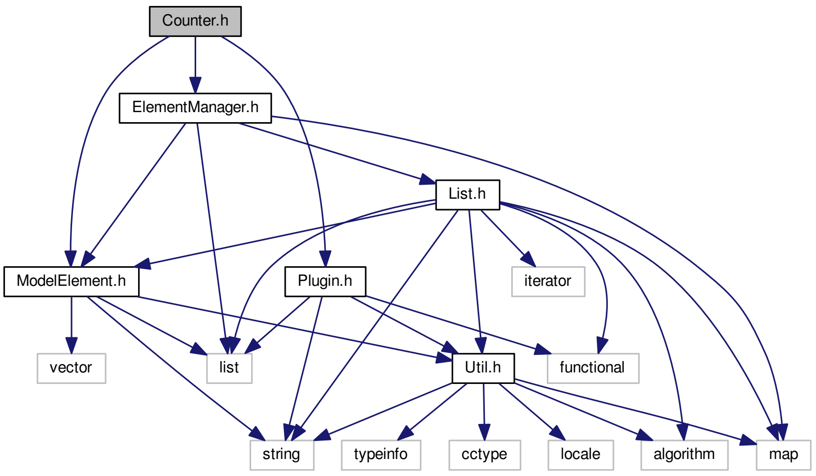
Include dependency graph for Counter.cpp:



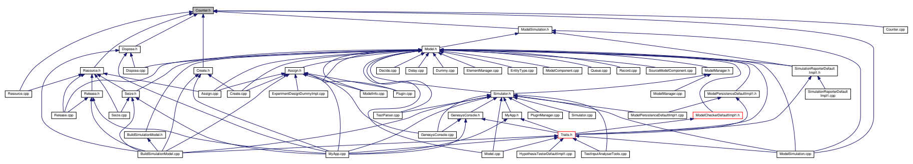
6.21 Counter.h File Reference

```
#include "ModelElement.h"
#include "ElementManager.h"
#include "Plugin.h"
```

Include dependency graph for Counter.h:



This graph shows which files directly or indirectly include this file:



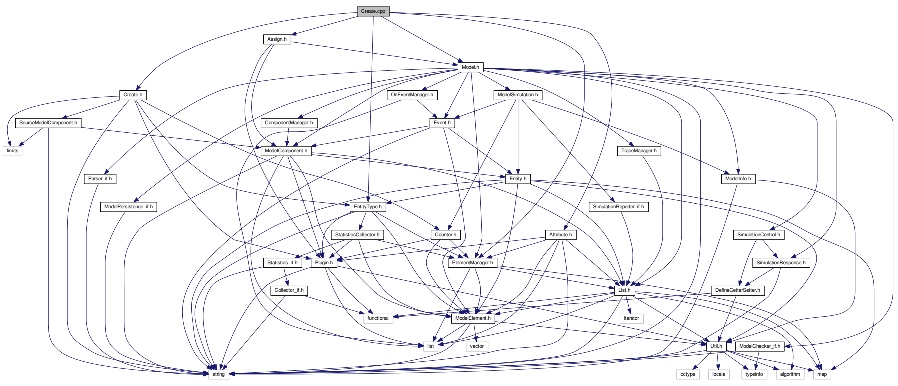
Classes

- class Counter

6.22 Create.cpp File Reference

```
#include "Create.h"
#include "Model.h"
#include "EntityType.h"
#include "ElementManager.h"
#include "Attribute.h"
#include "Assign.h"
```

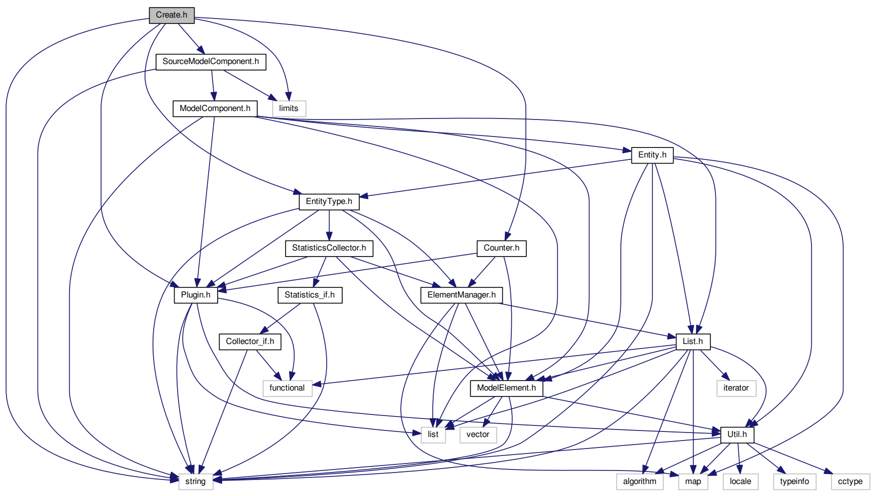
Include dependency graph for Create.cpp:



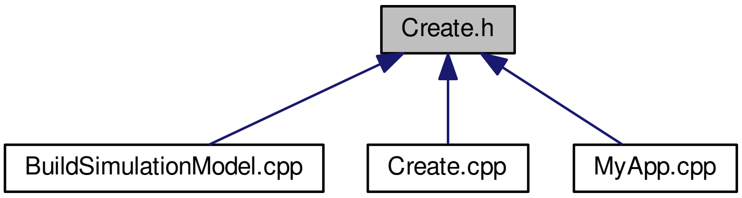
6.23 Create.h File Reference

```
#include <string>
#include <limits>
#include "SourceModelComponent.h"
#include "EntityType.h"
#include "Counter.h"
#include "Plugin.h"
```

Include dependency graph for Create.h:



This graph shows which files directly or indirectly include this file:



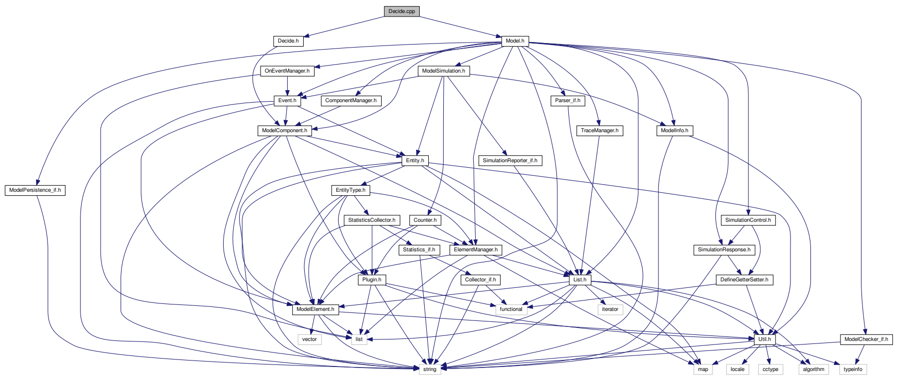
Classes

- class Create

6.24 Decide.cpp File Reference

```
#include "Decide.h"
#include "Model.h"
```

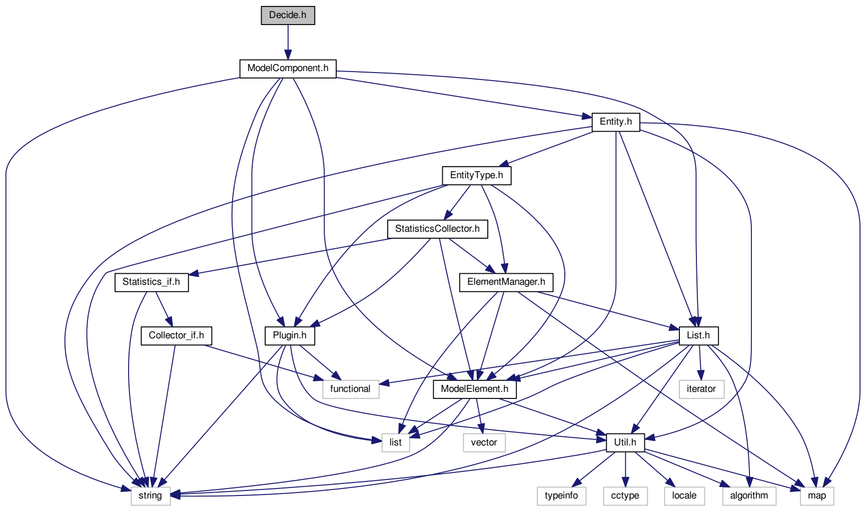
Include dependency graph for Decide.cpp:



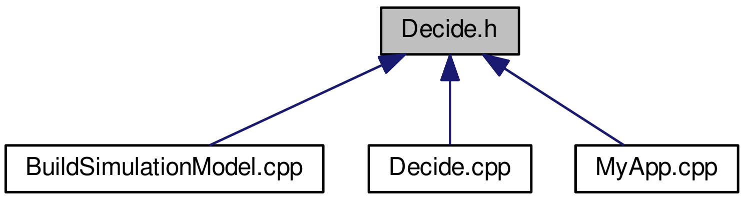
6.25 Decide.h File Reference

```
#include "ModelComponent.h"
```

Include dependency graph for Decide.h:



This graph shows which files directly or indirectly include this file:



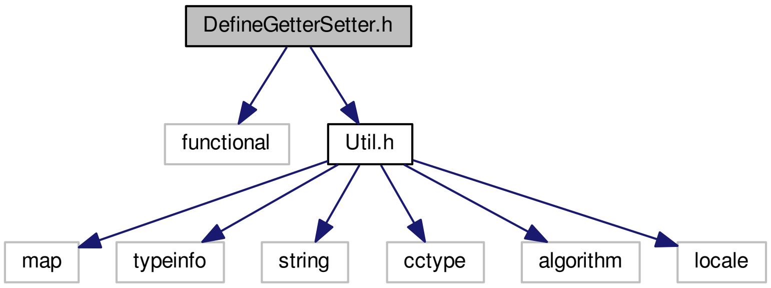
Classes

- class Decide

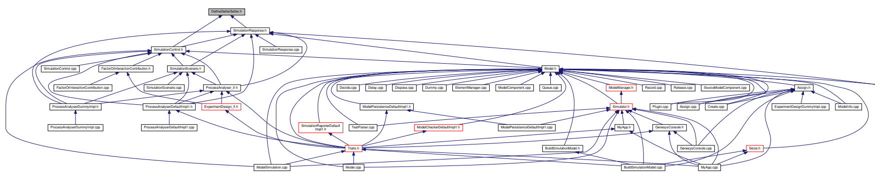
6.26 DefineGetterSetter.h File Reference

```
#include <functional>
#include "Util.h"
```

Include dependency graph for DefineGetterSetter.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef std::function< double() > GetterMember
- typedef std::function< void(double) > SetterMember

Functions

- template<typename Class >
  GetterMember DefineGetterMember (Class ∗object, double(Class::∗function)())
- template<typename Class >
  SetterMember DefineSetterMember (Class ∗object, void(Class::∗function)(double))
- template<typename Class >
  GetterMember DefineGetterMember (Class ∗object, unsigned int(Class::∗function)() const)
- template<typename Class >
  SetterMember DefineSetterMember (Class ∗object, void(Class::∗function)(unsigned int))
- template<typename Class >
  GetterMember DefineGetterMember (Class ∗object, bool(Class::∗function)() const)
- template<typename Class >
  SetterMember DefineSetterMember (Class ∗object, void(Class::∗function)(bool))
- template<typename Class >
  GetterMember DefineGetterMember (Class ∗object, std::string(Class::∗function)() const)
- template<typename Class >
  SetterMember DefineSetterMember (Class ∗object, void(Class::∗function)(std::string) const)
- template<typename Class >
  GetterMember DefineGetterMember (Class ∗object, Util::TimeUnit(Class::∗function)() const)
- template<typename Class >
  SetterMember DefineSetterMember (Class ∗object, void(Class::∗function)(Util::TimeUnit))

6.26.1 Typedef Documentation

6.26.1.1 typedef std::function< double() > **GetterMember**

6.26.1.2 typedef std::function< void(double) > **SetterMember**

6.26.2 Function Documentation

6.26.2.1 template< typename Class > **GetterMember** DefineGetterMember (Class ∗ *object,* double(Class::∗)() *function*)

6.26.2.2 template< typename Class > **GetterMember** DefineGetterMember (Class ∗ *object,* unsigned int(Class::∗)() const *function*)

6.26.2.3 template< typename Class > **GetterMember** DefineGetterMember (Class ∗ *object,* bool(Class::∗)() const *function*)

6.26.2.4 template< typename Class > **GetterMember** DefineGetterMember (Class ∗ *object,* std::string(Class::∗)() const *function*)

6.26.2.5 template< typename Class > **GetterMember** DefineGetterMember (Class ∗ *object,* **Util::TimeUnit**(Class::∗)() const *function*)

6.26.2.6 template< typename Class > **SetterMember** DefineSetterMember (Class ∗ *object,* void(Class::∗)(double) *function*)

6.26.2.7 template< typename Class > **SetterMember** DefineSetterMember (Class ∗ *object,* void(Class::∗)(unsigned int) *function*)

6.26.2.8 template< typename Class > **SetterMember** DefineSetterMember (Class ∗ *object,* void(Class::∗)(bool) *function*)

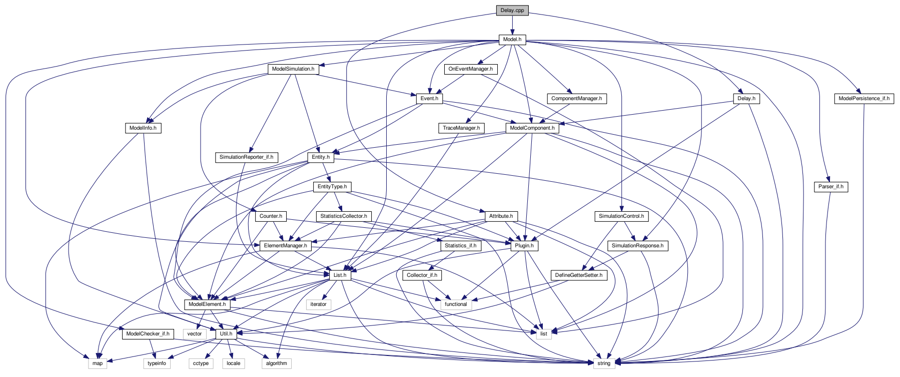
6.26.2.9 template< typename Class > **SetterMember** DefineSetterMember (Class ∗ *object,* void(Class::∗)(std::string) const *function*)

6.26.2.10 template< typename Class > **SetterMember** DefineSetterMember (Class ∗ *object,* void(Class::∗)(**Util::TimeUnit**) *function*)

6.27 Delay.cpp File Reference

```
#include "Delay.h"
#include "Model.h"
#include "Attribute.h"
```

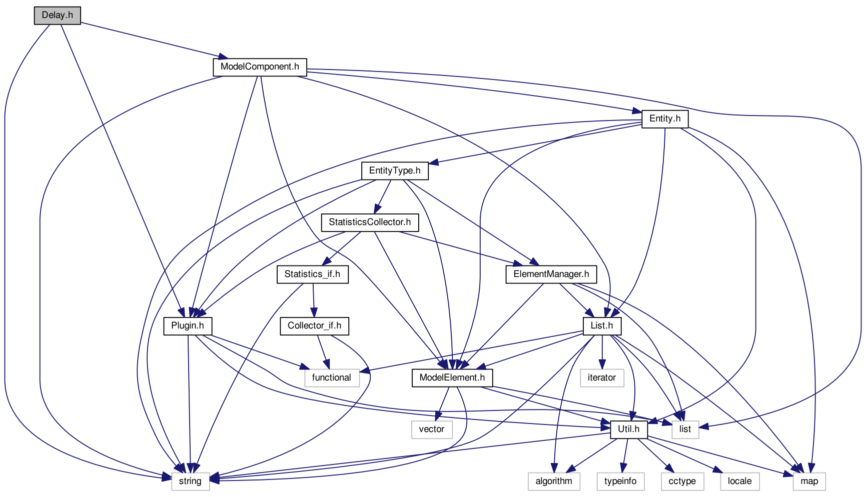
Include dependency graph for Delay.cpp:



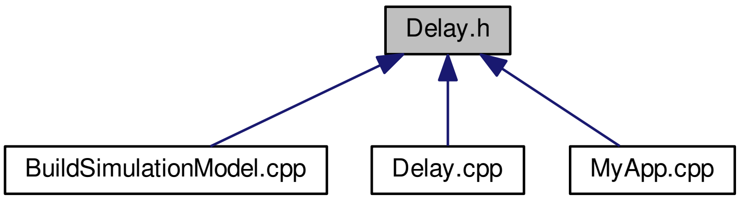
6.28 Delay.h File Reference

```
#include <string>
#include "ModelComponent.h"
#include "Plugin.h"
```

Include dependency graph for Delay.h:



This graph shows which files directly or indirectly include this file:



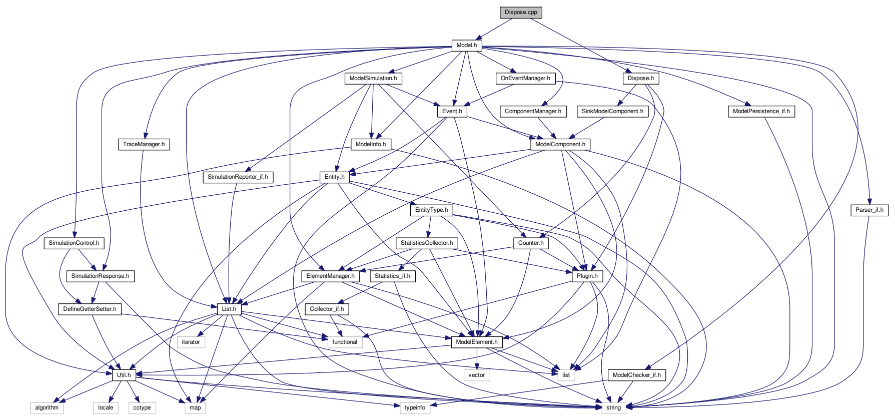
Classes

- class Delay

6.29 Dispose.cpp File Reference

```
#include "Dispose.h"
#include "Model.h"
```

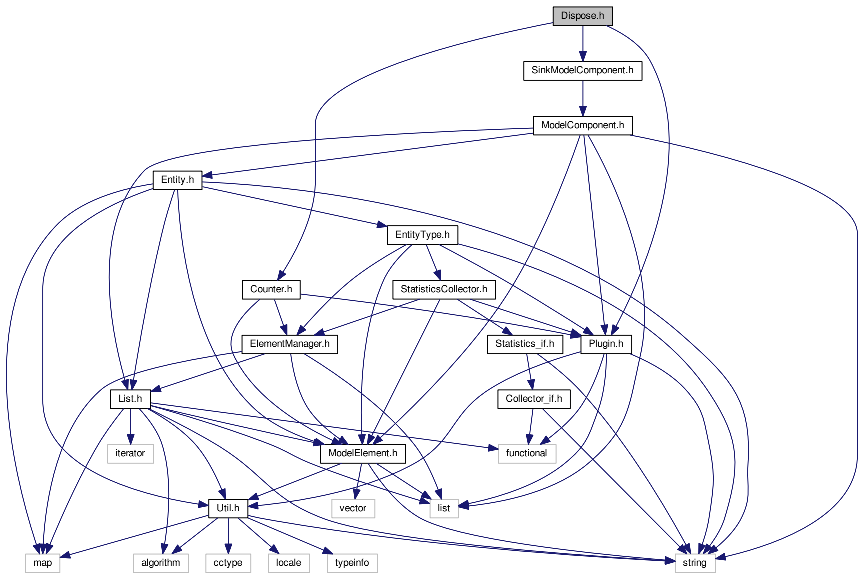
Include dependency graph for Dispose.cpp:



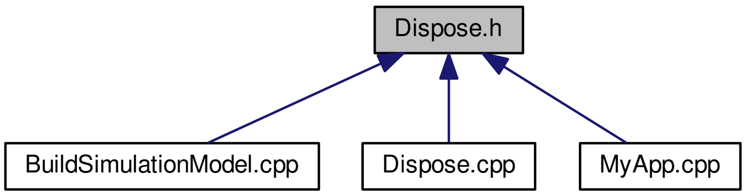
6.30 Dispose.h File Reference

```
#include "SinkModelComponent.h"
#include "Counter.h"
#include "Plugin.h"
```

Include dependency graph for Dispose.h:



This graph shows which files directly or indirectly include this file:



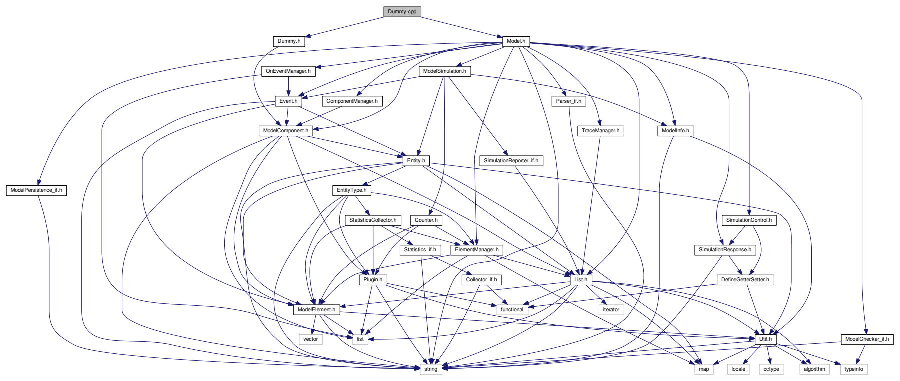
Classes

- class Dispose

6.31 Dummy.cpp File Reference

```
#include "Dummy.h"
#include "Model.h"
```

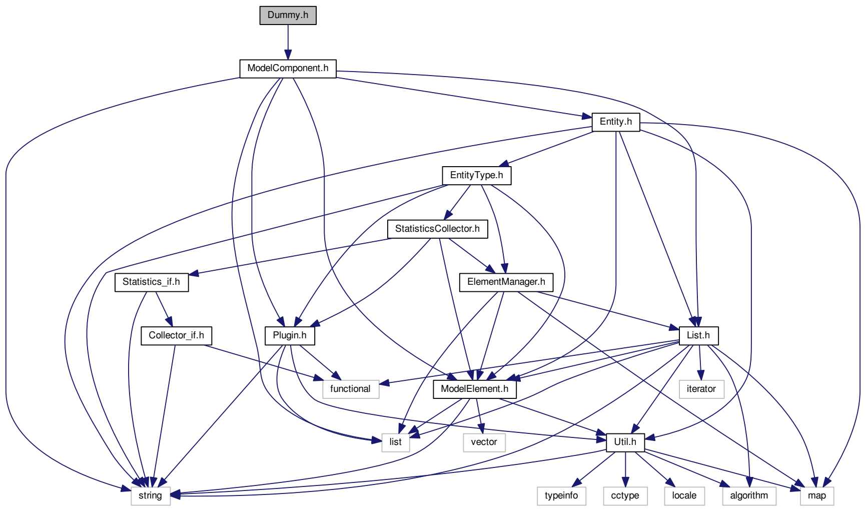
Include dependency graph for Dummy.cpp:



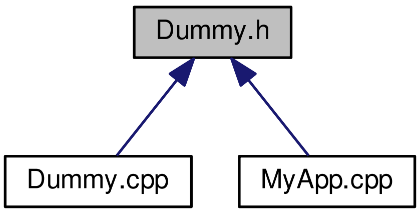
6.32 Dummy.h File Reference

```
#include "ModelComponent.h"
```

Include dependency graph for Dummy.h:



This graph shows which files directly or indirectly include this file:



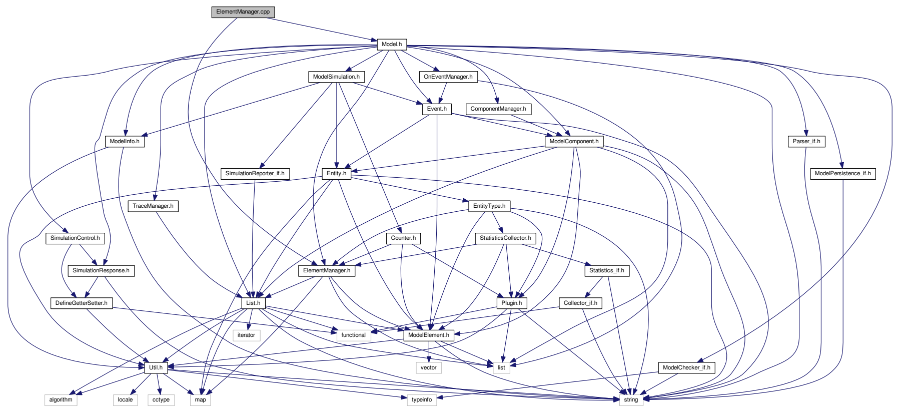
Classes

- class Dummy

6.33 ElementManager.cpp File Reference

```
#include "ElementManager.h"
#include "Model.h"
```

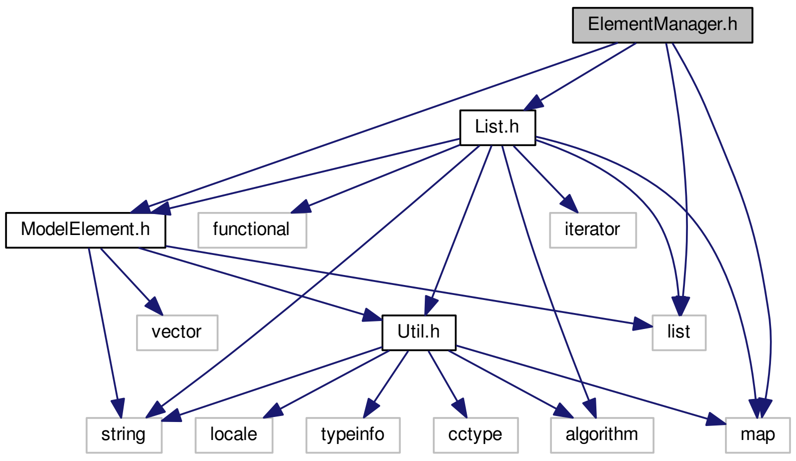
Include dependency graph for ElementManager.cpp:



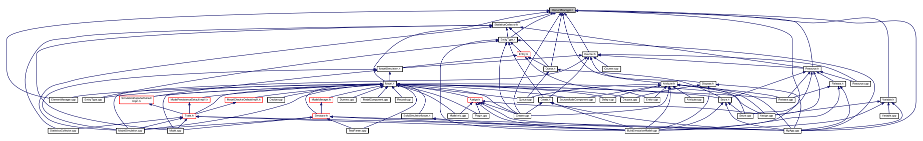
6.34 ElementManager.h File Reference

```
#include <list>
#include <map>
#include "List.h"
#include "ModelElement.h"
```

Include dependency graph for ElementManager.h:



This graph shows which files directly or indirectly include this file:



Classes

- class ElementManager

6.35 ElementManager_if.h File Reference

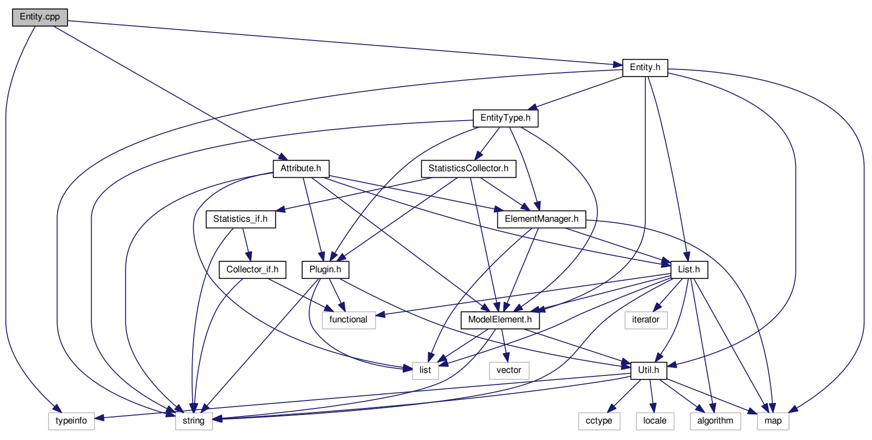
Classes

- class ElementManager_if

6.36 Entity.cpp File Reference

```
#include <typeinfo>
#include "Entity.h"
#include "Attribute.h"
```

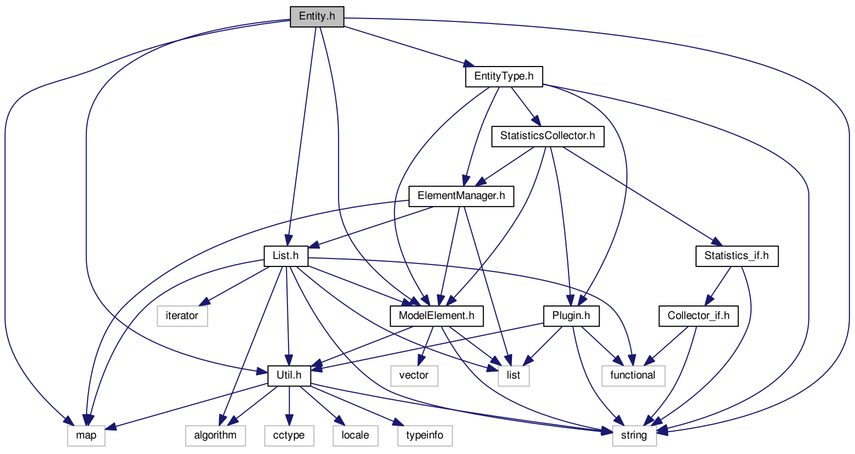
Include dependency graph for Entity.cpp:



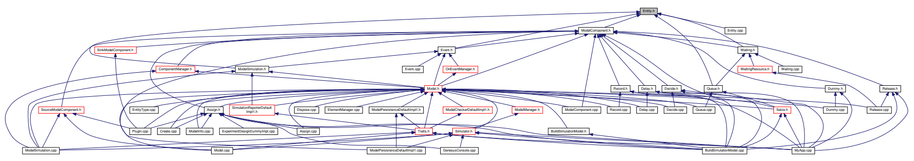
6.37 Entity.h File Reference

```
#include <string>
#include <map>
#include "Util.h"
#include "List.h"
#include "ModelElement.h"
#include "EntityType.h"
```

Include dependency graph for Entity.h:



This graph shows which files directly or indirectly include this file:



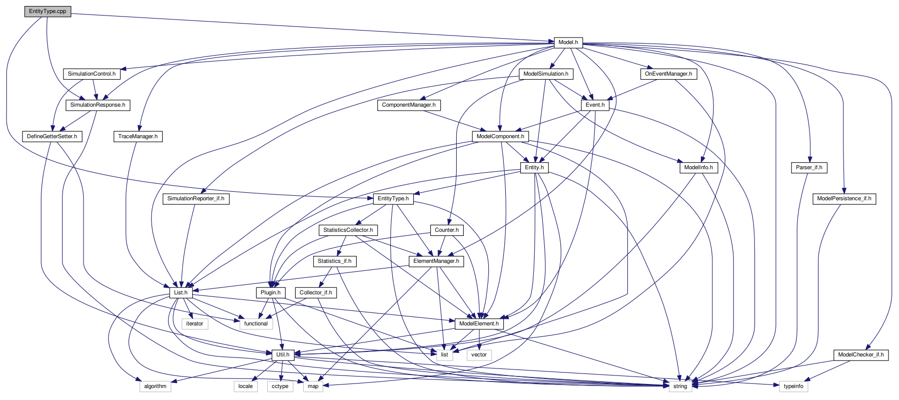
Classes

- class Entity

6.38 EntityType.cpp File Reference

```
#include "EntityType.h"
#include "Model.h"
#include "SimulationResponse.h"
```

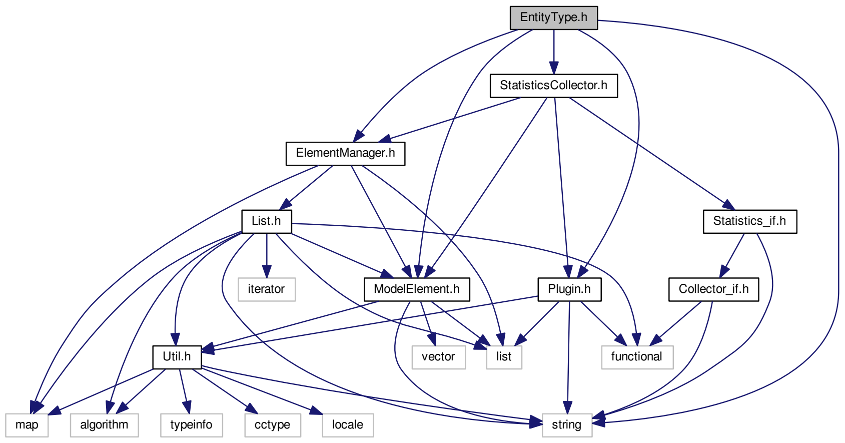
Include dependency graph for EntityType.cpp:



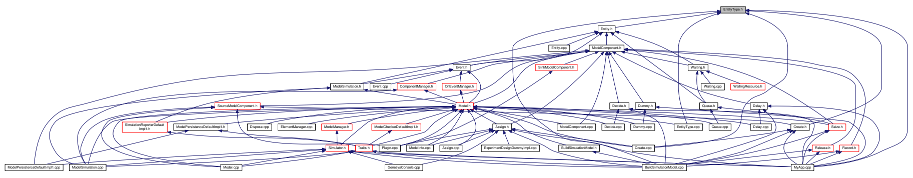
6.39 EntityType.h File Reference

```
#include <string>
#include "ModelElement.h"
#include "StatisticsCollector.h"
#include "ElementManager.h"
#include "Plugin.h"
```

Include dependency graph for EntityType.h:



This graph shows which files directly or indirectly include this file:



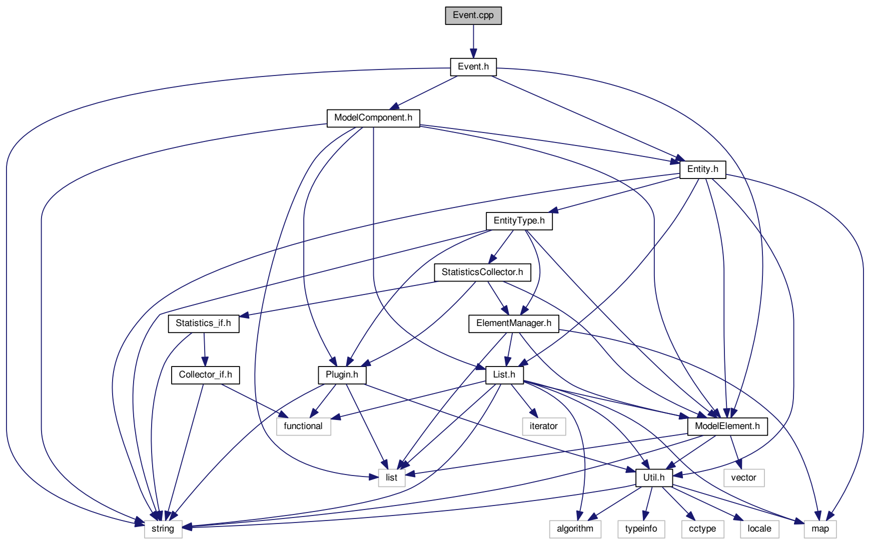
Classes

- class EntityType

6.40 Event.cpp File Reference

```
#include "Event.h"
```

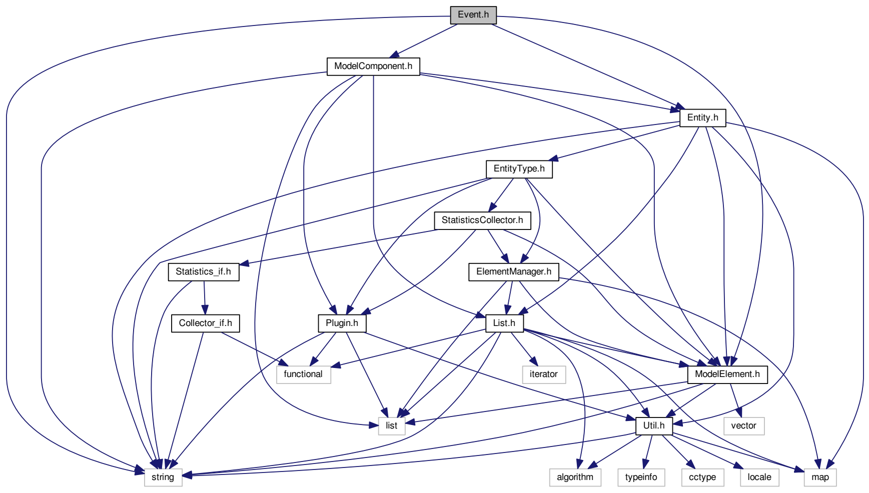
Include dependency graph for Event.cpp:



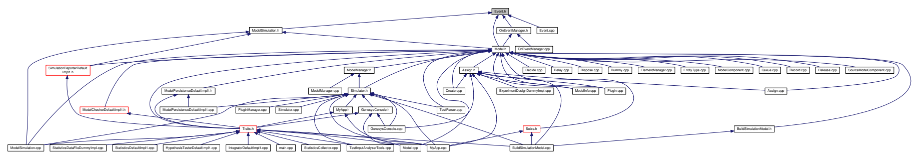
6.41 Event.h File Reference

```
#include <string>
#include "ModelElement.h"
#include "Entity.h"
#include "ModelComponent.h"
```

Include dependency graph for Event.h:



This graph shows which files directly or indirectly include this file:



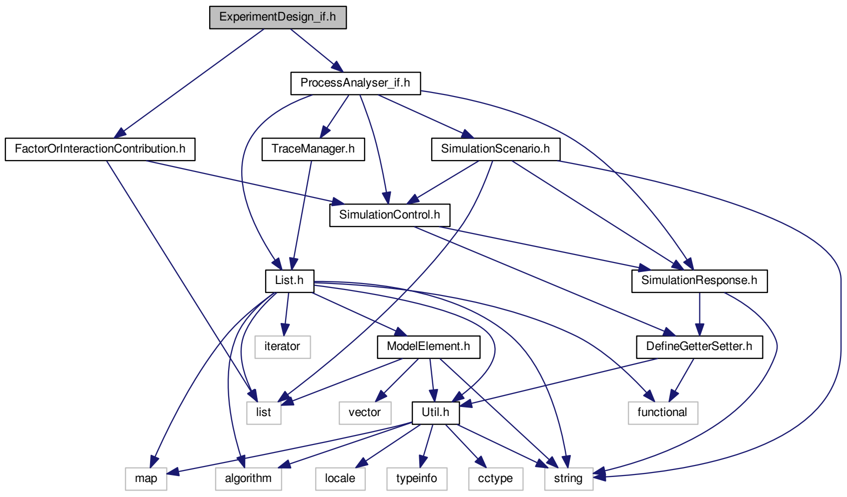
Classes

- class Event

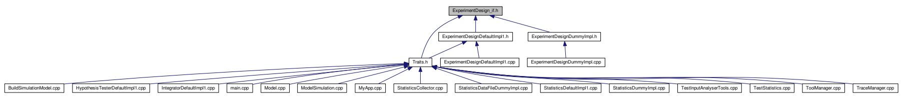
6.42 ExperimentDesign_if.h File Reference

```
#include "FactorOrInteractionContribution.h"
#include "ProcessAnalyser_if.h"
```

Include dependency graph for ExperimentDesign_if.h:



This graph shows which files directly or indirectly include this file:


ExperimentDesign_if.h
ExperimentDesignDefaultImpl1.h
ExperimentDesignDummyImpl.h
Traits.h
ExperimentDesignDefaultImpl1.cpp
ExperimentDesignDummyImpl.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

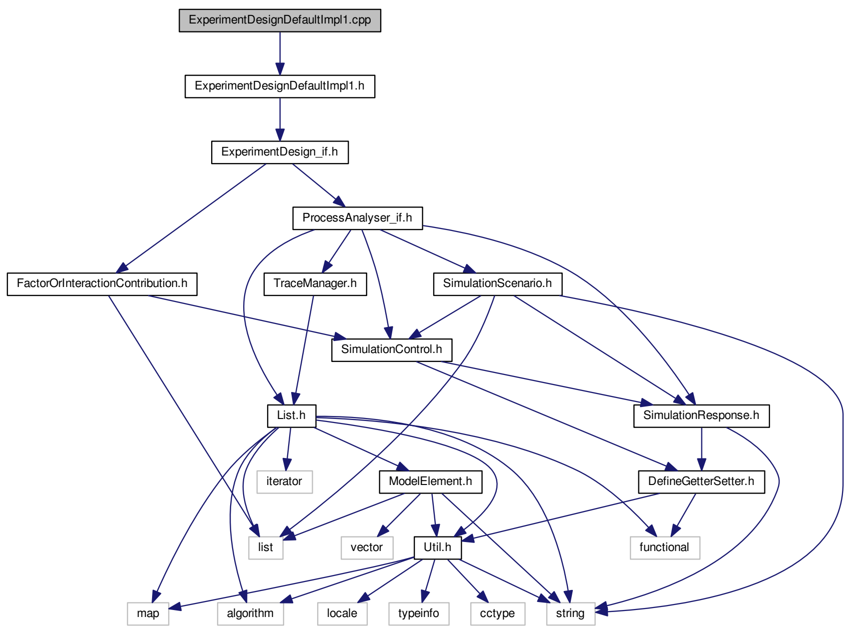
Classes

- class ExperimentDesign_if

6.43 ExperimentDesignDefaultImpl1.cpp File Reference

```
#include "ExperimentDesignDefaultImpl1.h"
```

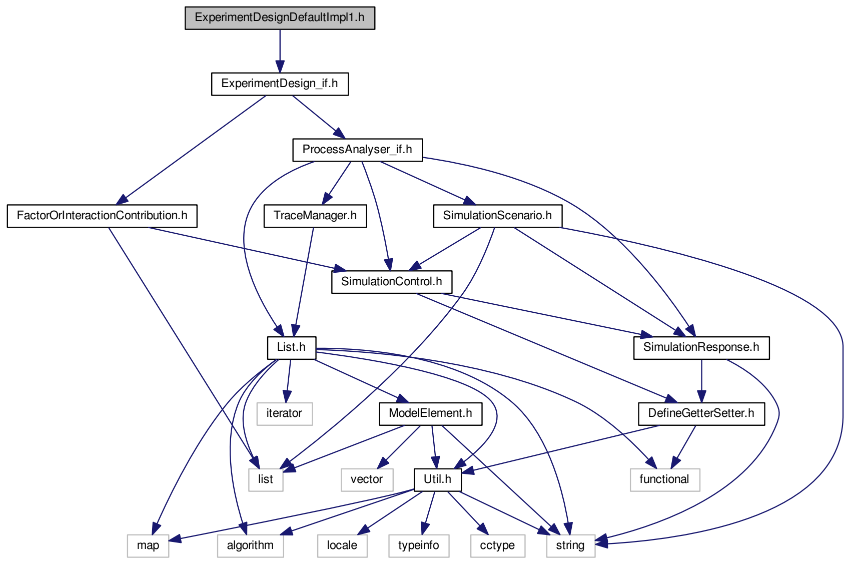
Include dependency graph for ExperimentDesignDefaultImpl1.cpp:



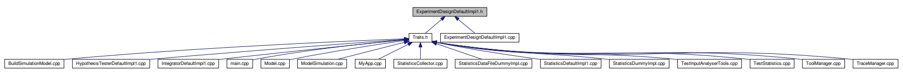
6.44 ExperimentDesignDefaultImpl1.h File Reference

```
#include "ExperimentDesign_if.h"
```

Include dependency graph for ExperimentDesignDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:


ExperimentDesignDefaultImpl1.h
Traits.h
ExperimentDesignDefaultImpl1.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

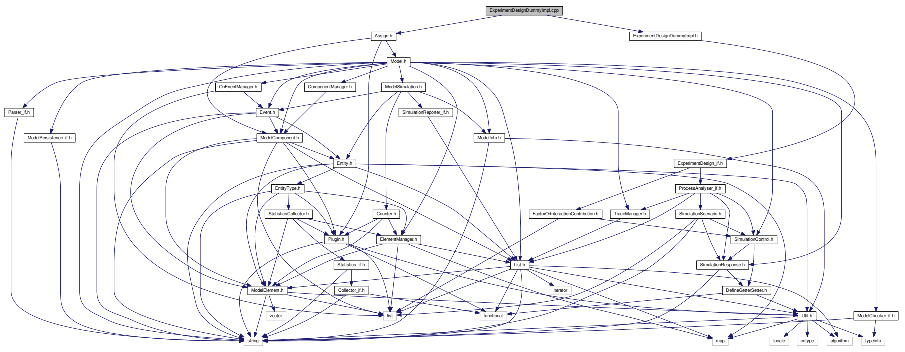
Classes

- class ExperimentDesignDefaultImpl1

6.45 ExperimentDesignDummyImpl.cpp File Reference

```
#include "ExperimentDesignDummyImpl.h"
#include "Assign.h"
```

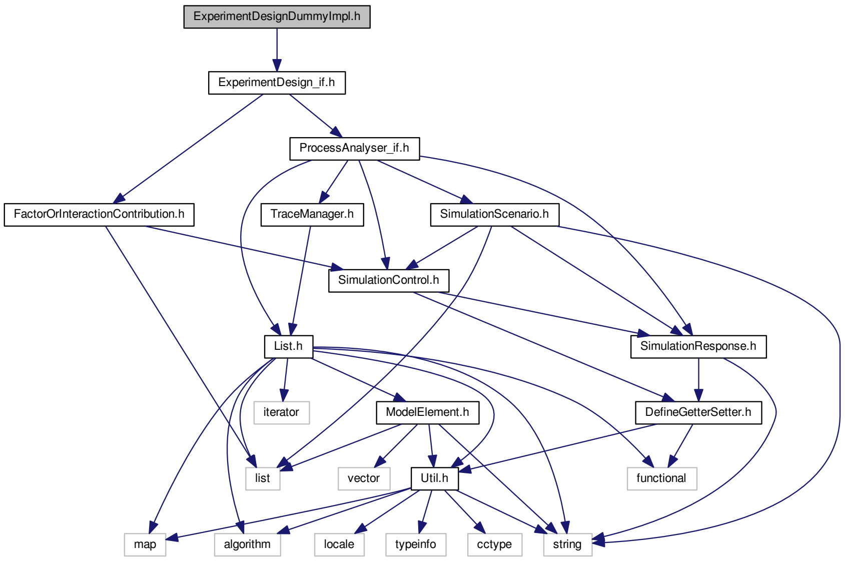
Include dependency graph for ExperimentDesignDummyImpl.cpp:



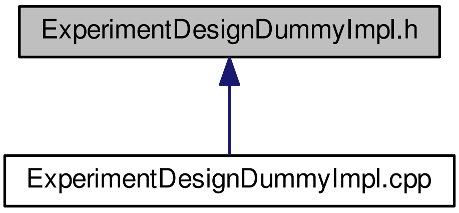
6.46 ExperimentDesignDummyImpl.h File Reference

```
#include "ExperimentDesign_if.h"
```

Include dependency graph for ExperimentDesignDummyImpl.h:



This graph shows which files directly or indirectly include this file:



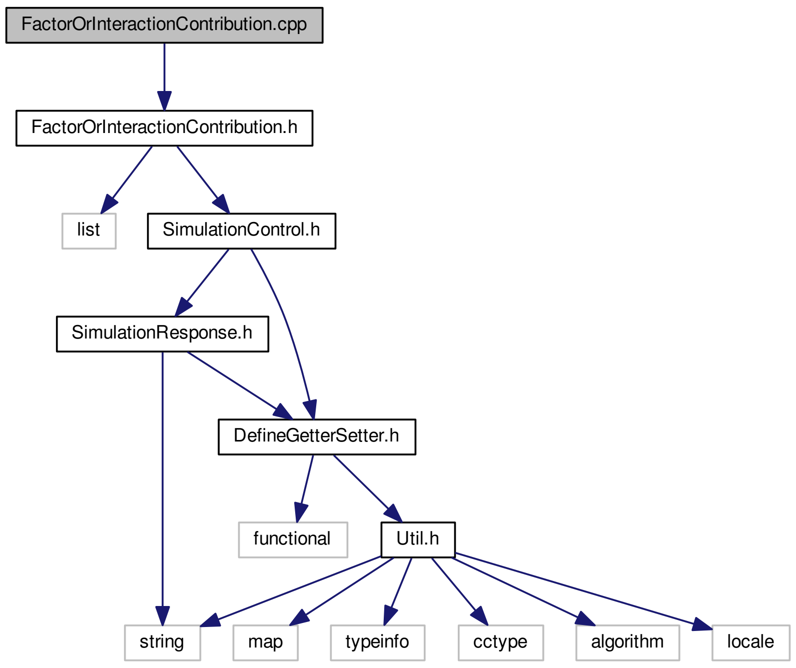
Classes

- class ExperimentDesignDummyImpl

6.47 FactorOrInteractionContribution.cpp File Reference

```
#include "FactorOrInteractionContribution.h"
```

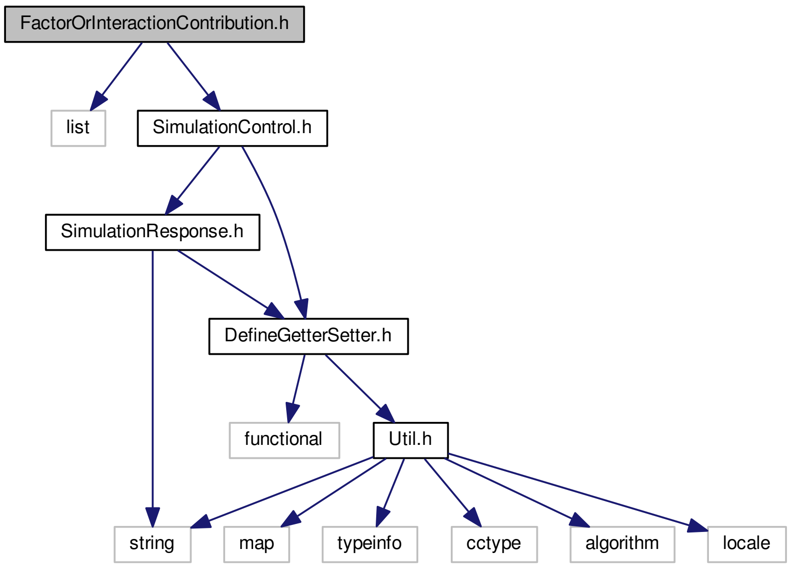
Include dependency graph for FactorOrInteractionContribution.cpp:



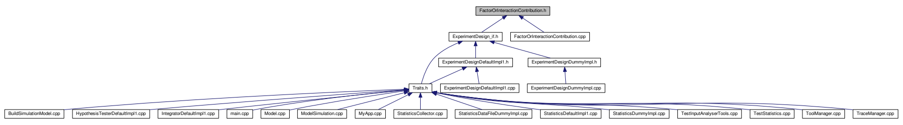
6.48 FactorOrInteractionContribution.h File Reference

```
#include <list>
#include "SimulationControl.h"
```

Include dependency graph for FactorOrInteractionContribution.h:



This graph shows which files directly or indirectly include this file:


FactorOrInteractionContribution.h
ExperimentDesign_if.h
FactorOrInteractionContribution.cpp
ExperimentDesignDefaultImpl1.h
ExperimentDesignDummyImpl.h
Traits.h
ExperimentDesignDefaultImpl1.cpp
ExperimentDesignDummyImpl.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

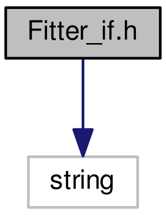
Classes

- class FactorOrInteractionContribution

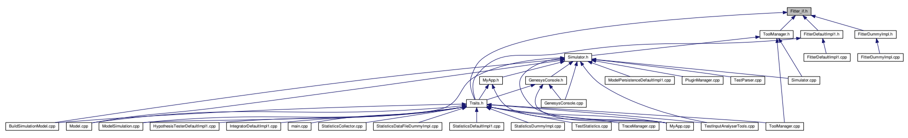
6.49 Fitter_if.h File Reference

```
#include <string>
```

Include dependency graph for Fitter_if.h:



This graph shows which files directly or indirectly include this file:



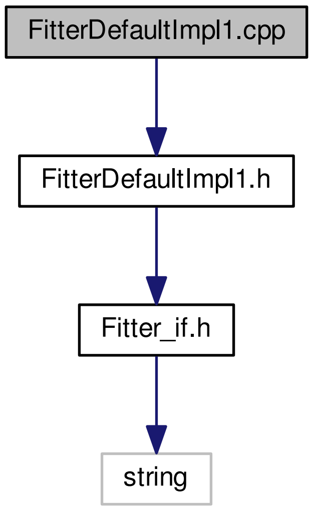
Classes

- class Fitter_if

6.50 FitterDefaultImpl1.cpp File Reference

```
#include "FitterDefaultImpl1.h"
```

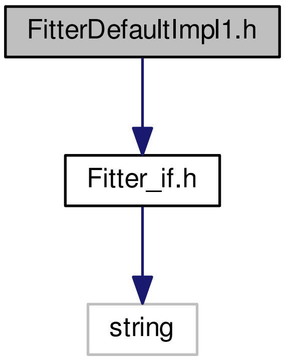
Include dependency graph for FitterDefaultImpl1.cpp:



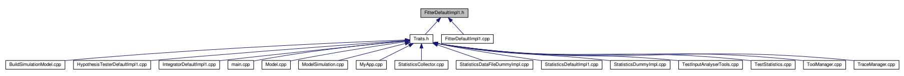
6.51 FitterDefaultImpl1.h File Reference

```
#include "Fitter_if.h"
```

Include dependency graph for FitterDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



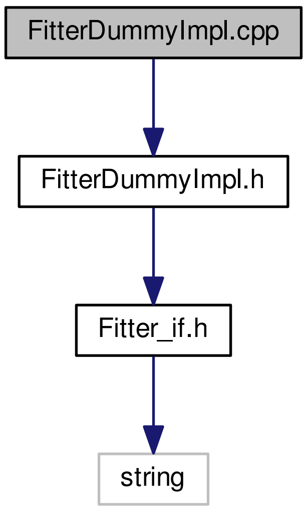
Classes

- class FitterDefaultImpl1

6.52 FitterDummyImpl.cpp File Reference

```
#include "FitterDummyImpl.h"
```

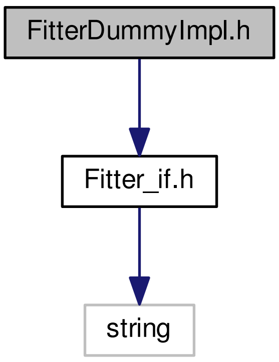
Include dependency graph for FitterDummyImpl.cpp:



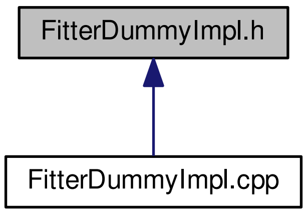
6.53 FitterDummyImpl.h File Reference

```
#include "Fitter_if.h"
```

Include dependency graph for FitterDummyImpl.h:



This graph shows which files directly or indirectly include this file:



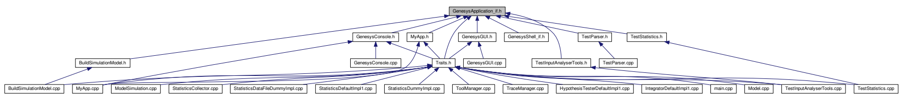
Classes

- class FitterDummyImpl

6.54 Functor.h File Reference

6.55 GenesysApplication_if.h File Reference

This graph shows which files directly or indirectly include this file:



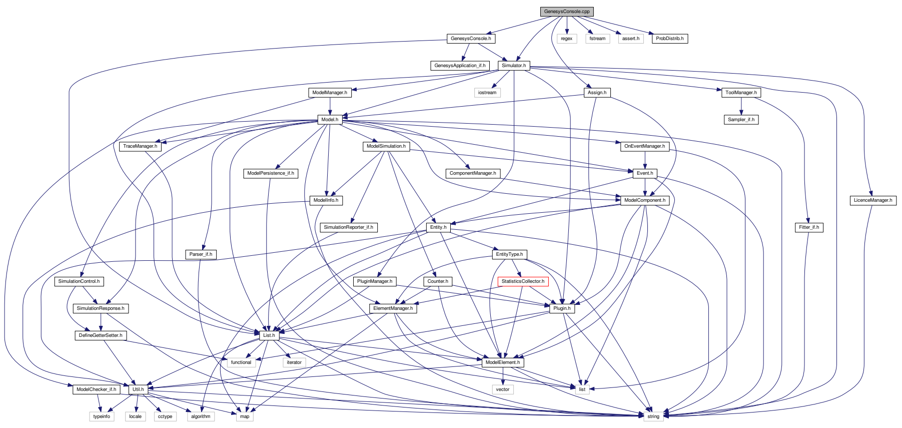
Classes

- class GenesysApplication_if

6.56 GenesysConsole.cpp File Reference

```
#include "GenesysConsole.h"
#include "Simulator.h"
#include "Assign.h"
#include <regex>
#include <fstream>
#include <assert.h>
#include "ProbDistrib.h"
```

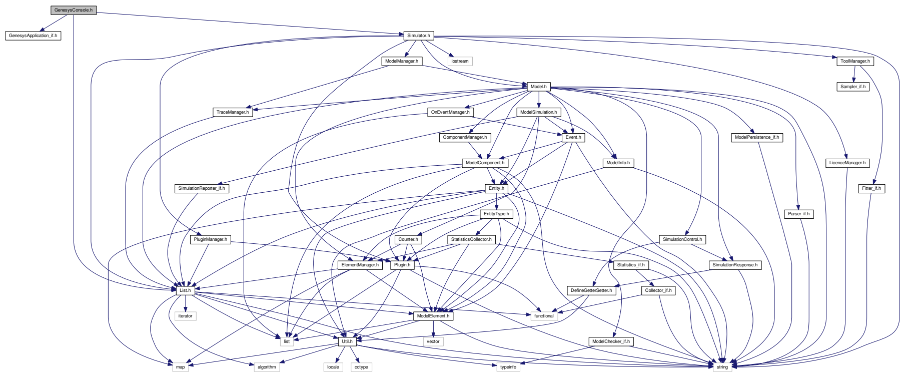
Include dependency graph for GenesysConsole.cpp:



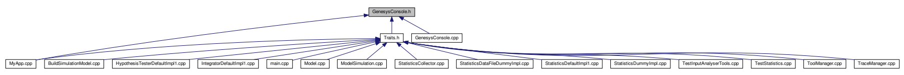
6.57 GenesysConsole.h File Reference

```
#include "GenesysApplication_if.h"
#include "Simulator.h"
#include "List.h"
```

Include dependency graph for GenesysConsole.h:



This graph shows which files directly or indirectly include this file:



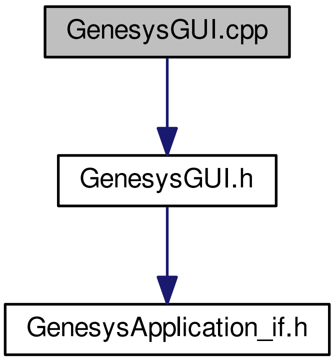
Classes

- class GenesysConsole

6.58 GenesysGUI.cpp File Reference

```
#include "GenesysGUI.h"
```

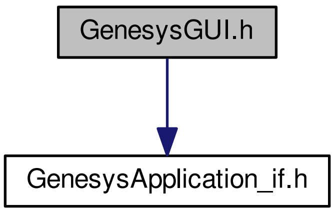
Include dependency graph for GenesysGUI.cpp:



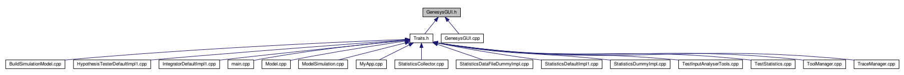
6.59 GenesysGUI.h File Reference

```
#include "GenesysApplication_if.h"
```

Include dependency graph for GenesysGUI.h:



This graph shows which files directly or indirectly include this file:



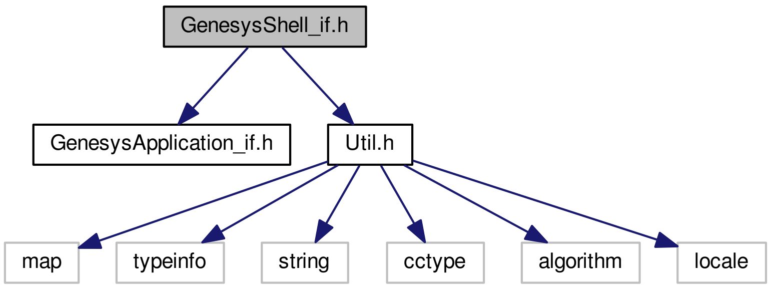
Classes

- class GenesysGUI

6.60 GenesysShell_if.h File Reference

```
#include "GenesysApplication_if.h"
#include "Util.h"
```

Include dependency graph for GenesysShell_if.h:



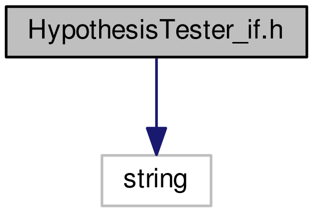
Classes

- class GenesysShell_if

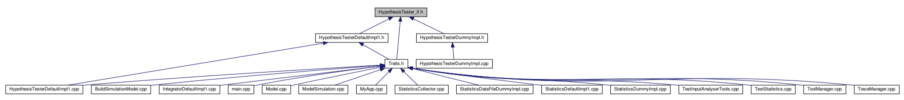
6.61 HypothesisTester_if.h File Reference

```
#include <string>
```

Include dependency graph for HypothesisTester_if.h:



This graph shows which files directly or indirectly include this file:


HypothesisTester_if.h
HypothesisTesterDefaultImpl1.h
HypothesisTesterDummyImpl.h
Traits.h
HypothesisTesterDummyImpl.cpp
HypothesisTesterDefaultImpl1.cpp
BuildSimulationModel.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

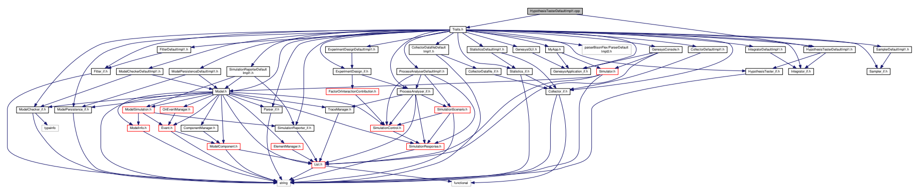
Classes

- class HypothesisTester_if

6.62 HypothesisTesterDefaultImpl1.cpp File Reference

```
#include "HypothesisTesterDefaultImpl1.h"
#include "Traits.h"
```

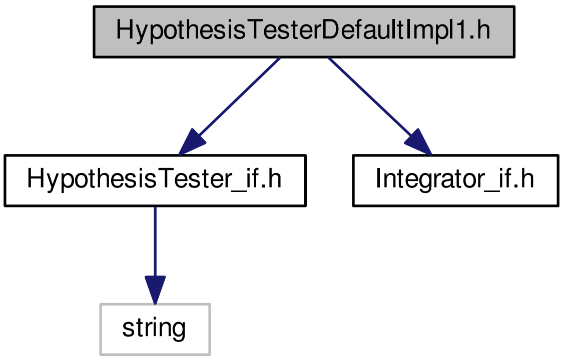
Include dependency graph for HypothesisTesterDefaultImpl1.cpp:



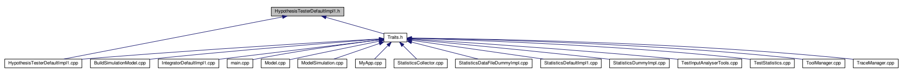
6.63 HypothesisTesterDefaultImpl1.h File Reference

```
#include "HypothesisTester_if.h"
#include "Integrator_if.h"
```

Include dependency graph for HypothesisTesterDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



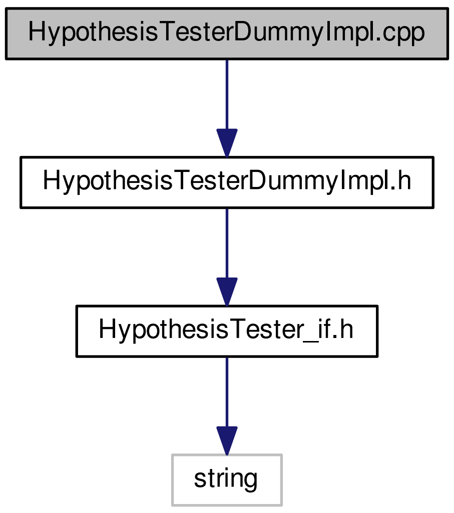
Classes

- class HypothesisTesterDefaultImpl1

6.64 HypothesisTesterDummyImpl.cpp File Reference

```
#include "HypothesisTesterDummyImpl.h"
```

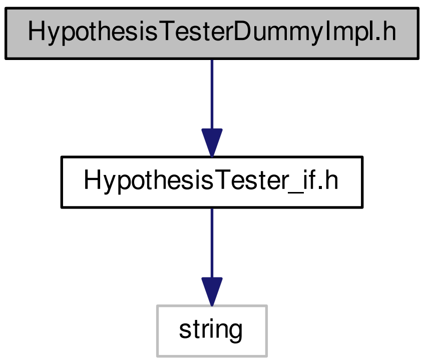
Include dependency graph for HypothesisTesterDummyImpl.cpp:



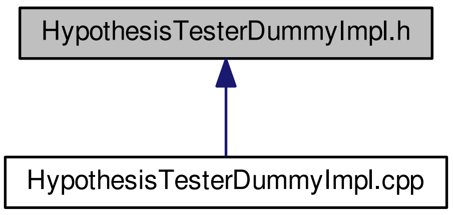
6.65 HypothesisTesterDummyImpl.h File Reference

```
#include "HypothesisTester_if.h"
```

Include dependency graph for HypothesisTesterDummyImpl.h:



This graph shows which files directly or indirectly include this file:

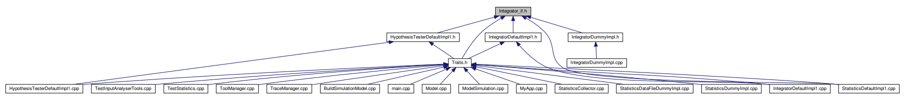


Classes

- class HypothesisTesterDummyImpl

6.66 Integrator_if.h File Reference

This graph shows which files directly or indirectly include this file:



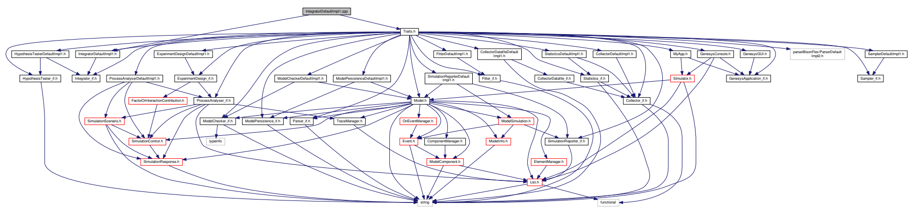
Classes

- class Integrator_if

6.67 IntegratorDefaultImpl1.cpp File Reference

```
#include "IntegratorDefaultImpl1.h"
#include "Traits.h"
```

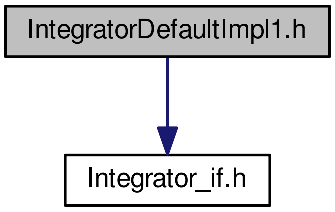
Include dependency graph for IntegratorDefaultImpl1.cpp:



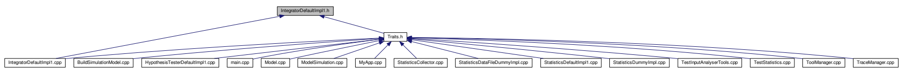
6.68 IntegratorDefaultImpl1.h File Reference

```
#include "Integrator_if.h"
```

Include dependency graph for IntegratorDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



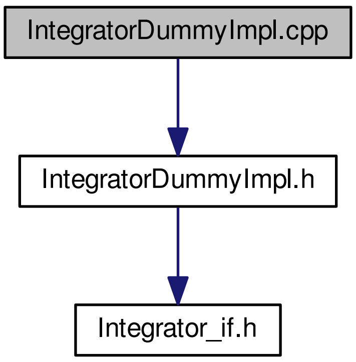
Classes

- class IntegratorDefaultImpl1

6.69 IntegratorDummyImpl.cpp File Reference

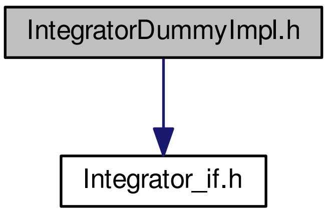
```
#include "IntegratorDummyImpl.h"
```

Include dependency graph for IntegratorDummyImpl.cpp:

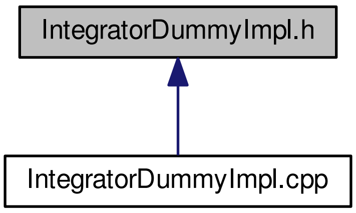


6.70 IntegratorDummyImpl.h File Reference

```
#include "Integrator_if.h"
```
Include dependency graph for IntegratorDummyImpl.h:



This graph shows which files directly or indirectly include this file:



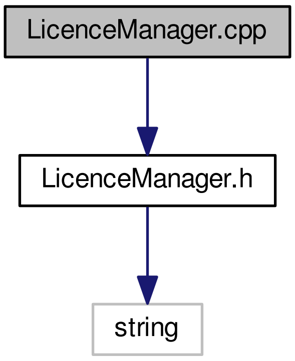
Classes

- class IntegratorDummyImpl

6.71 LicenceManager.cpp File Reference

```
#include "LicenceManager.h"
```

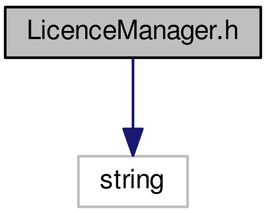
Include dependency graph for LicenceManager.cpp:



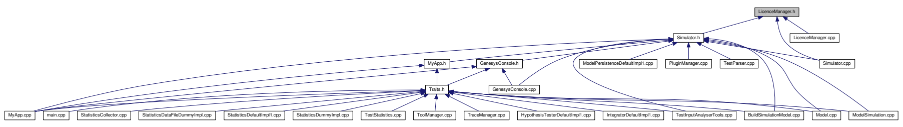
6.72 LicenceManager.h File Reference

```
#include <string>
```

Include dependency graph for LicenceManager.h:



This graph shows which files directly or indirectly include this file:



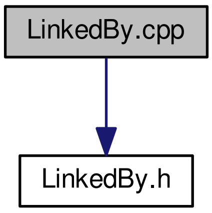
Classes

- class LicenceManager

6.73 LinkedBy.cpp File Reference

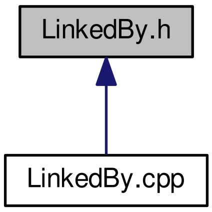
```
#include "LinkedBy.h"
```

Include dependency graph for LinkedBy.cpp:



6.74 LinkedBy.h File Reference

This graph shows which files directly or indirectly include this file:



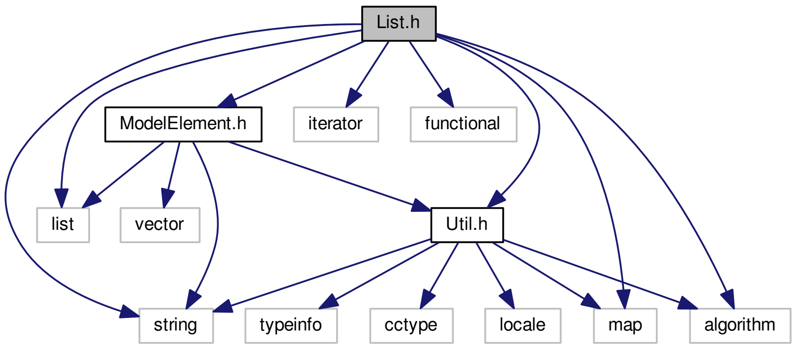
Classes

- class LinkedBy

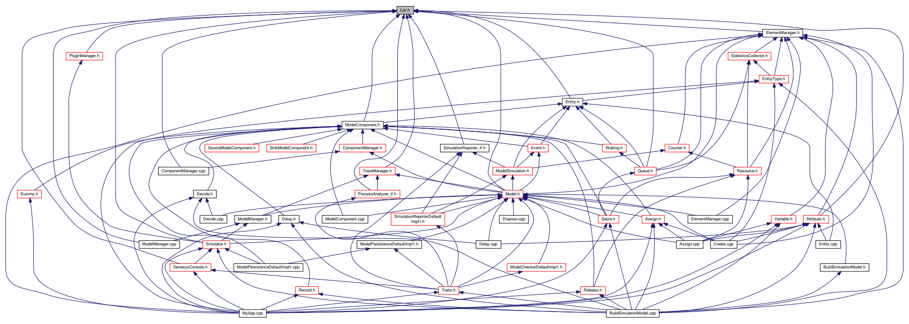
6.75 List.h File Reference

```
#include <string>
#include <list>
#include <map>
#include <iterator>
#include <functional>
#include <algorithm>
#include "Util.h"
#include "ModelElement.h"
```

Include dependency graph for List.h:



This graph shows which files directly or indirectly include this file:



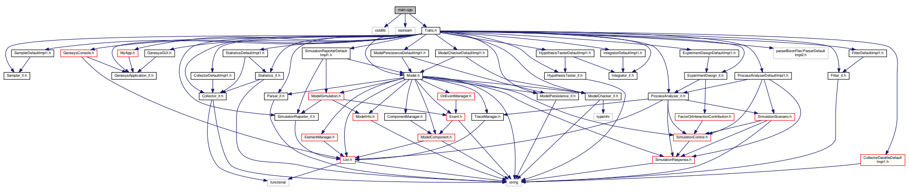
Classes

- class List< T >

6.76 main.cpp File Reference

```
#include <cstdlib>
#include <iostream>
#include "Traits.h"
```

Include dependency graph for main.cpp:



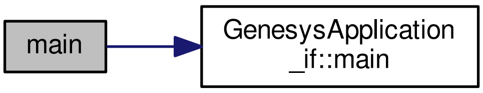
Functions

- int main (int argc, char ∗∗argv)

6.76.1 Function Documentation

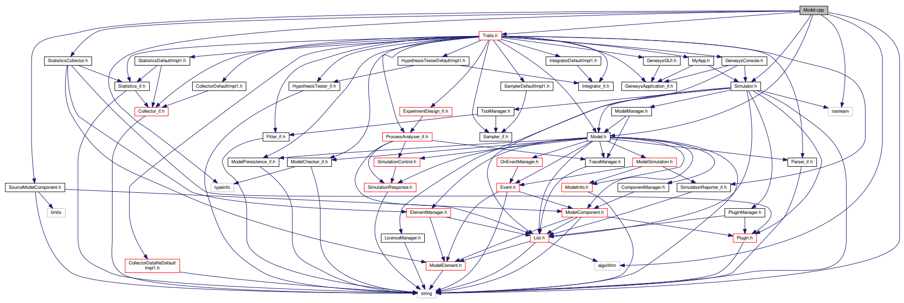
6.76.1.1 int main (int *argc,* char ∗∗ *argv*)

Here is the call graph for this function:



6.77 Model.cpp File Reference

```
#include <typeinfo>
#include <iostream>
#include <algorithm>
#include <string>
#include "Model.h"
#include "SourceModelComponent.h"
#include "Simulator.h"
#include "StatisticsCollector.h"
#include "Traits.h"
```
Include dependency graph for Model.cpp:



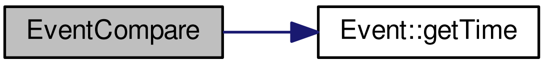
Functions

- bool EventCompare (const Event ∗a, const Event ∗b)
- double getReplicationLengthNotMemberFunction ()
- void setReplicationLengthNotMemberFunction (double value)

6.77.1 Function Documentation

6.77.1.1 bool EventCompare (const Event ∗ *a,* const Event ∗ *b*)

Here is the call graph for this function:



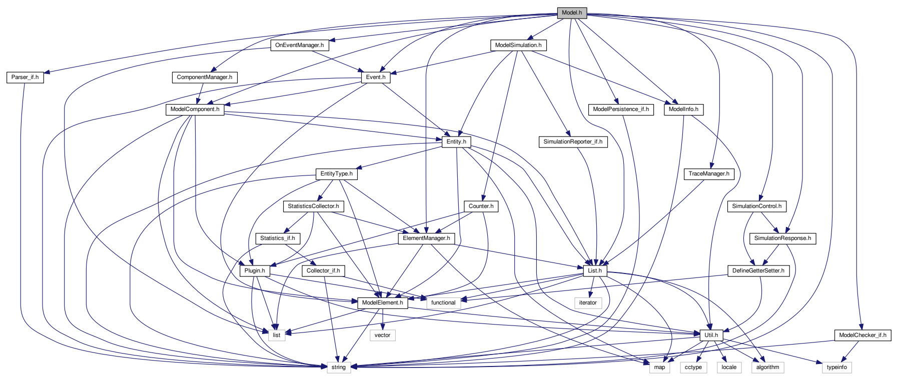
6.77.1.2 double getReplicationLengthNotMemberFunction ( )

6.77.1.3 void setReplicationLengthNotMemberFunction (double *value*)

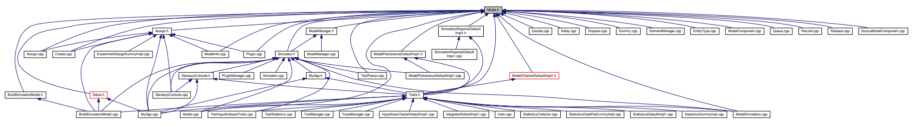
6.78 Model.h File Reference

```
#include <string>
#include "List.h"
#include "ModelComponent.h"
#include "Event.h"
#include "ModelChecker_if.h"
#include "Parser_if.h"
#include "ModelPersistence_if.h"
#include "ElementManager.h"
#include "ComponentManager.h"
#include "TraceManager.h"
#include "OnEventManager.h"
#include "ModelInfo.h"
#include "ModelSimulation.h"
#include "SimulationResponse.h"
#include "SimulationControl.h"
```

Include dependency graph for Model.h:



This graph shows which files directly or indirectly include this file:



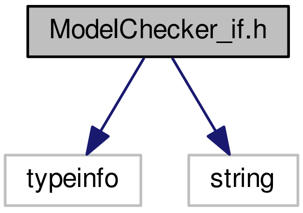
Classes

- class Model

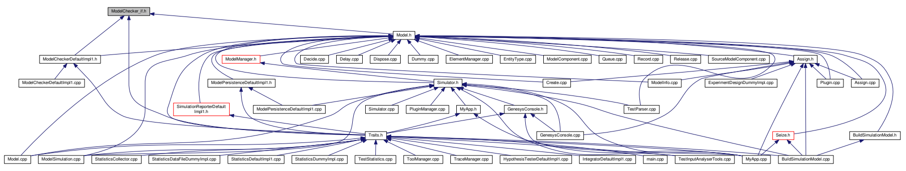
6.79 ModelChecker_if.h File Reference

```
#include <typeinfo>
#include <string>
```

Include dependency graph for ModelChecker_if.h:



This graph shows which files directly or indirectly include this file:



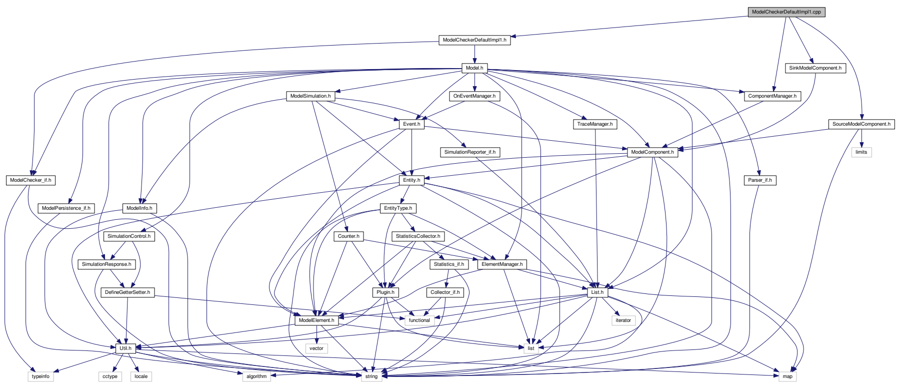
Classes

- class ModelChecker_if

6.80 ModelCheckerDefaultImpl1.cpp File Reference

```
#include "ModelCheckerDefaultImpl1.h"
#include "SourceModelComponent.h"
#include "SinkModelComponent.h"
#include "ComponentManager.h"
```

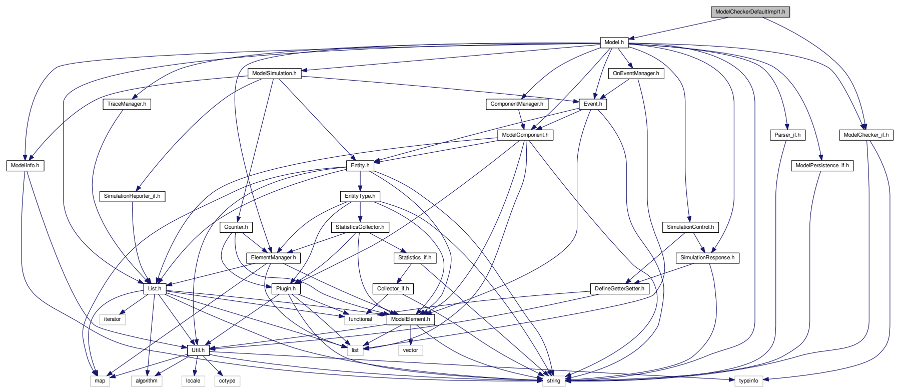
Include dependency graph for ModelCheckerDefaultImpl1.cpp:



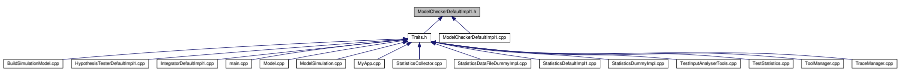
6.81 ModelCheckerDefaultImpl1.h File Reference

```
#include "ModelChecker_if.h"
#include "Model.h"
```

Include dependency graph for ModelCheckerDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



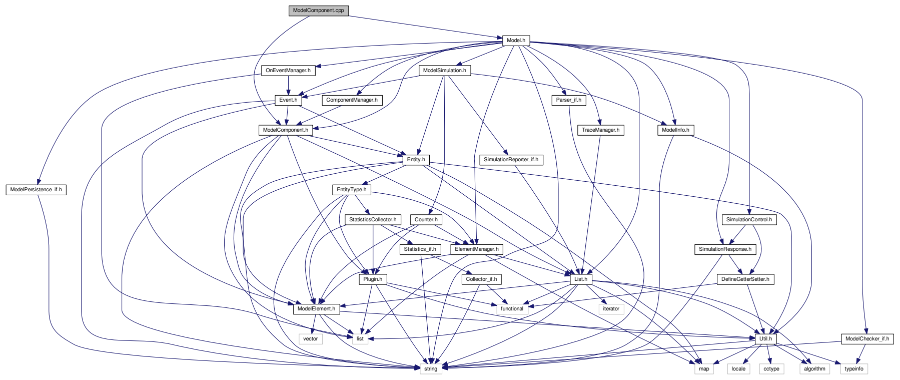
Classes

- class ModelCheckerDefaultImpl1

6.82 ModelComponent.cpp File Reference

```
#include "ModelComponent.h"
#include "Model.h"
```

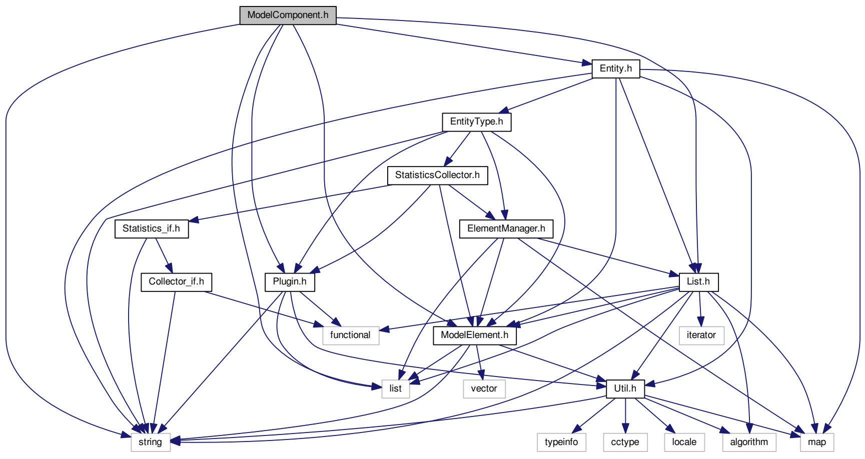
Include dependency graph for ModelComponent.cpp:



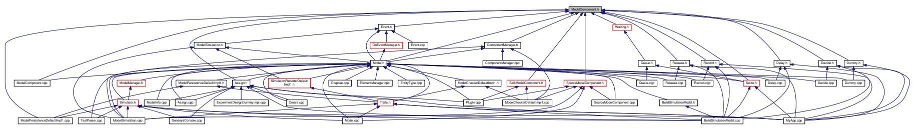
6.83 ModelComponent.h File Reference

```
#include <string>
#include <list>
#include "Plugin.h"
#include "List.h"
#include "Entity.h"
#include "ModelElement.h"
```

Include dependency graph for ModelComponent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class ModelComponent

6.84 ModelComponentManager_if.h File Reference

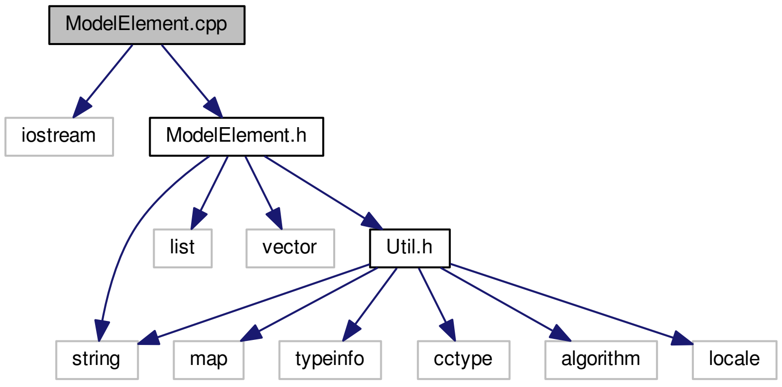
Classes

- class ModelComponentManager_if

6.85 ModelElement.cpp File Reference

```
#include <iostream>
#include "ModelElement.h"
```

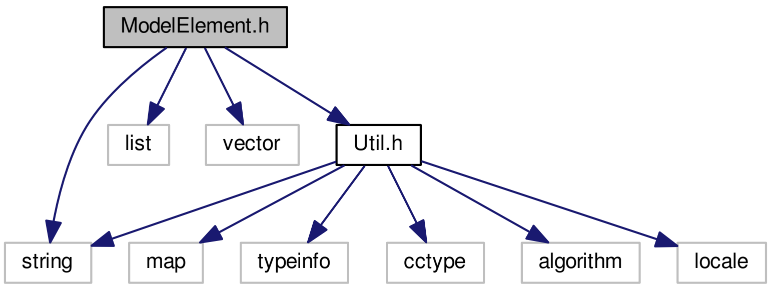
Include dependency graph for ModelElement.cpp:



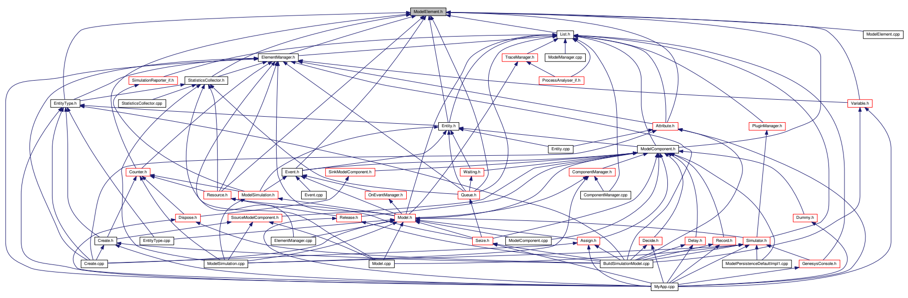
6.86 ModelElement.h File Reference

```
#include <string>
#include <list>
#include <vector>
#include "Util.h"
```

Include dependency graph for ModelElement.h:



This graph shows which files directly or indirectly include this file:



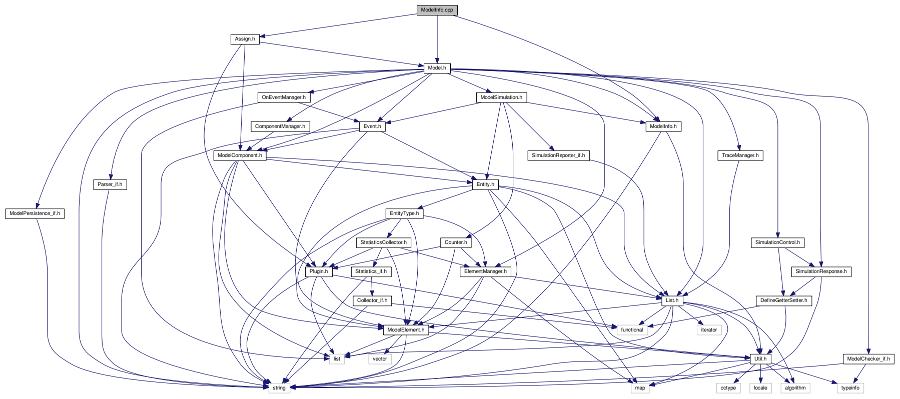
Classes

- class ModelElement

6.87 ModelInfo.cpp File Reference

```
#include "ModelInfo.h"
#include "Model.h"
#include "Assign.h"
```

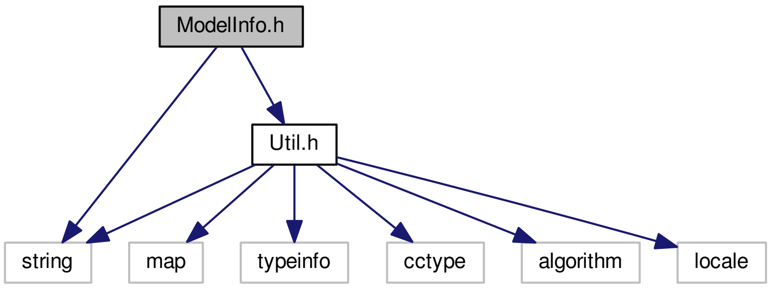
Include dependency graph for ModelInfo.cpp:



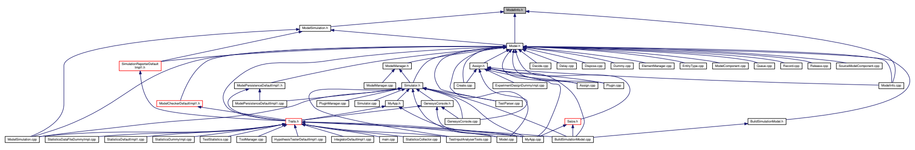
6.88 ModelInfo.h File Reference

```
#include <string>
#include "Util.h"
```

Include dependency graph for ModelInfo.h:



This graph shows which files directly or indirectly include this file:



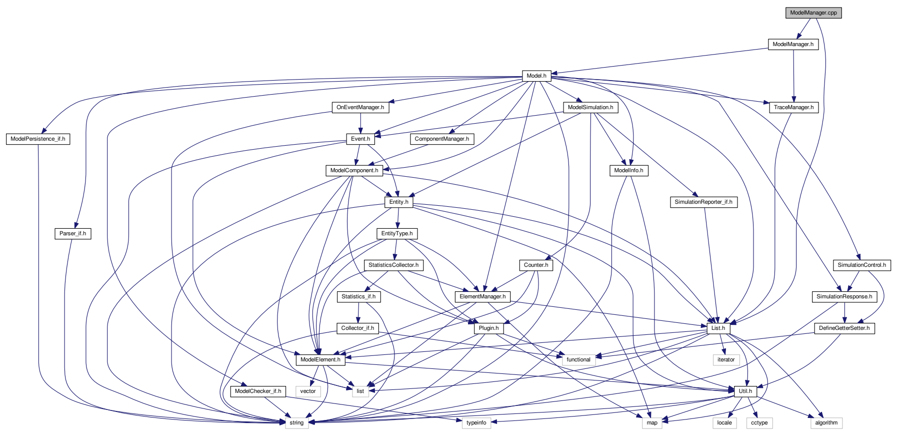
Classes

- class ModelInfo

6.89 ModelManager.cpp File Reference

```
#include "ModelManager.h"
#include "List.h"
```

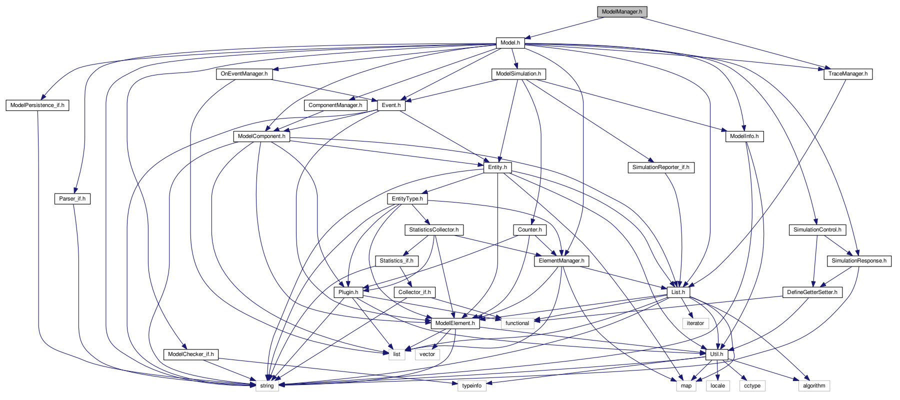
Include dependency graph for ModelManager.cpp:



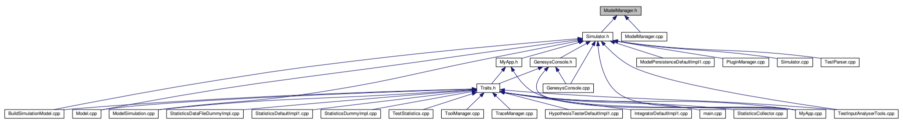
6.90 ModelManager.h File Reference

```
#include "Model.h"
#include "TraceManager.h"
```

Include dependency graph for ModelManager.h:



This graph shows which files directly or indirectly include this file:

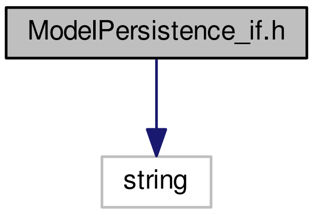


Classes

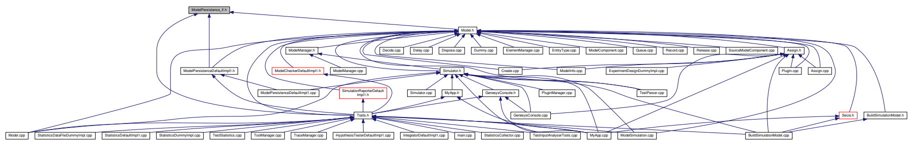
- class ModelManager

6.91 ModelPersistence_if.h File Reference

```
#include <string>
```
Include dependency graph for ModelPersistence_if.h:



This graph shows which files directly or indirectly include this file:



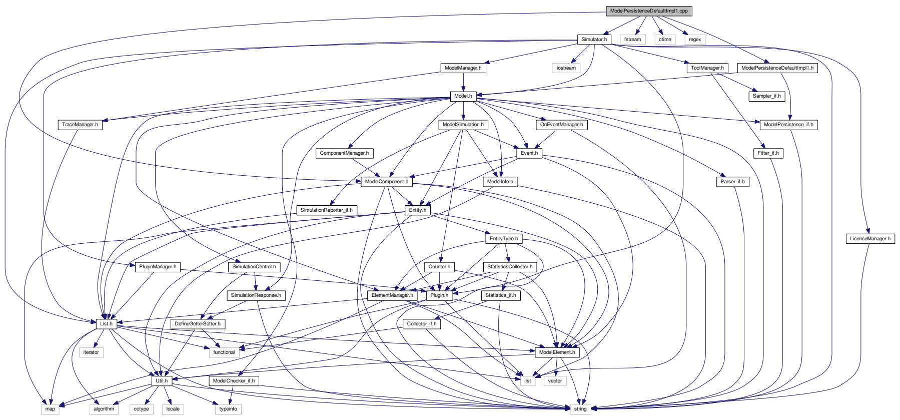
Classes

- class ModelPersistence_if

6.92 ModelPersistenceDefaultImpl1.cpp File Reference

```
#include "ModelPersistenceDefaultImpl1.h"
#include <fstream>
#include <ctime>
#include <regex>
#include "ModelComponent.h"
#include "Simulator.h"
```

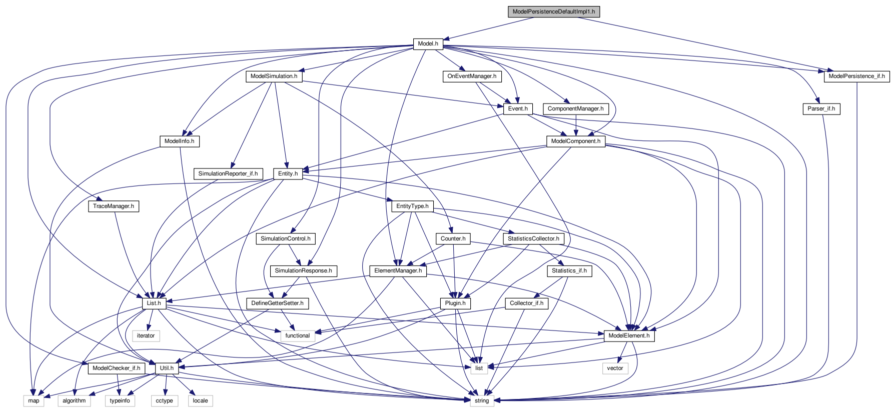
Include dependency graph for ModelPersistenceDefaultImpl1.cpp:



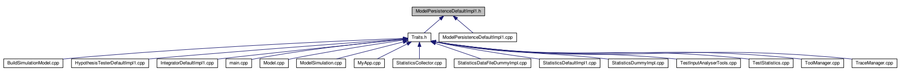
6.93 ModelPersistenceDefaultImpl1.h File Reference

```
#include "ModelPersistence_if.h"
#include "Model.h"
```

Include dependency graph for ModelPersistenceDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



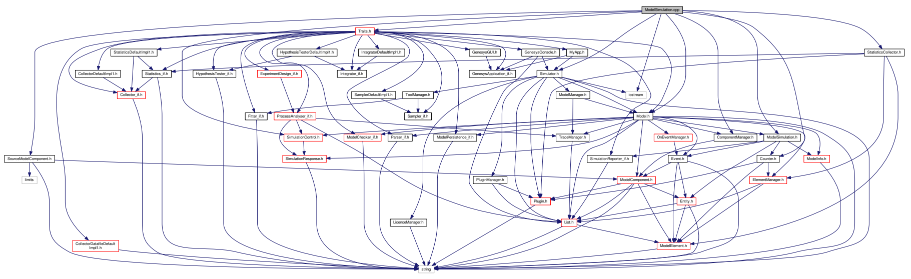
Classes

- class ModelPersistenceDefaultImpl1

6.94 ModelSimulation.cpp File Reference

```
#include "ModelSimulation.h"
#include <iostream>
#include "Model.h"
#include "Simulator.h"
#include "SourceModelComponent.h"
#include "StatisticsCollector.h"
#include "Counter.h"
#include "Traits.h"
#include "SimulationControl.h"
#include "ComponentManager.h"
```

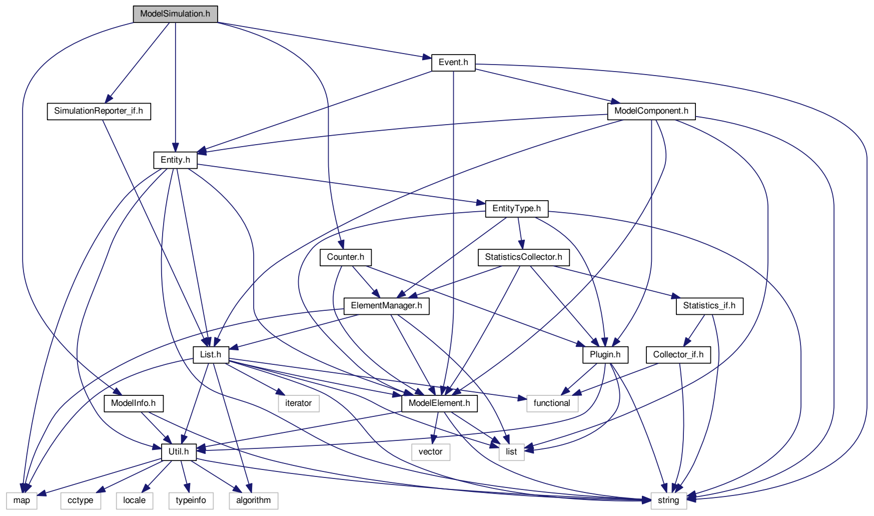
Include dependency graph for ModelSimulation.cpp:



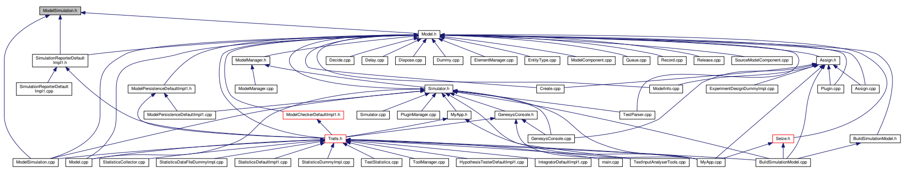
6.95 ModelSimulation.h File Reference

```
#include "Event.h"
#include "Entity.h"
#include "ModelInfo.h"
#include "SimulationReporter_if.h"
#include "Counter.h"
```

Include dependency graph for ModelSimulation.h:



This graph shows which files directly or indirectly include this file:



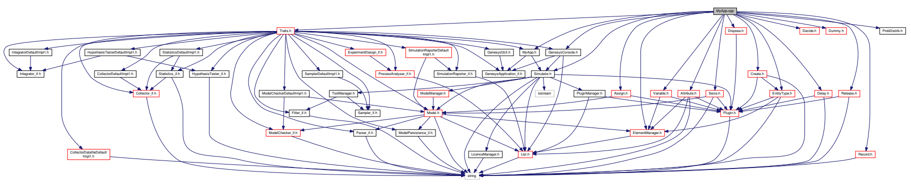
Classes

- class ModelSimulation

6.96 MyApp.cpp File Reference

```
#include "MyApp.h"
#include "GenesysConsole.h"
#include "Simulator.h"
#include "Traits.h"
#include "Create.h"
#include "Delay.h"
#include "Dispose.h"
#include "Seize.h"
#include "Release.h"
#include "Assign.h"
#include "Record.h"
#include "Decide.h"
#include "Dummy.h"
#include "ElementManager.h"
#include "EntityType.h"
#include "Attribute.h"
#include "Variable.h"
#include "ProbDistrib.h"
```

Include dependency graph for MyApp.cpp:



Functions

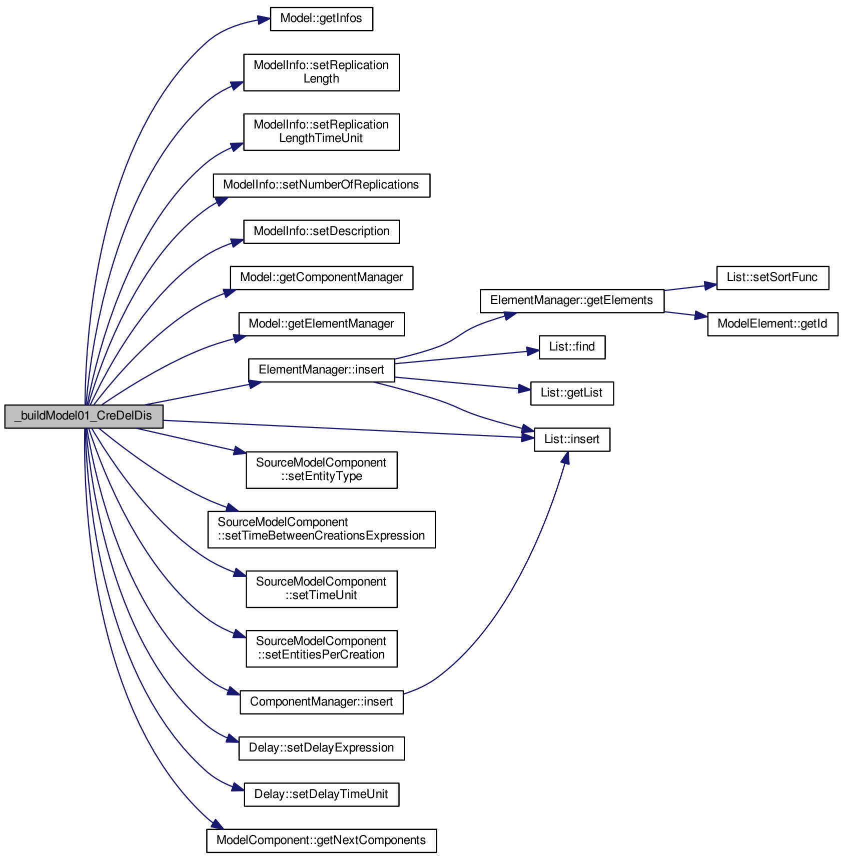
- void traceHandler (TraceEvent e)
- void traceSimulationHandler (TraceSimulationEvent e)
- void onSimulationStartHandler (SimulationEvent ∗re)
- void onReplicationStartHandler (SimulationEvent ∗re)
- void onProcessEventHandler (SimulationEvent ∗re)
- void onReplicationEndHandler (SimulationEvent ∗re)

- void onEntityRemoveHandler (SimulationEvent ∗re)
- void _buildModel01_CreDelDis (Model ∗model)
- void _buildModel02_CreDelDis (Model ∗model)
- void _buildModel03_CreSeiDelResDis (Model ∗model)
- void _buildMostCompleteModel (Model ∗model)

6.96.1 Function Documentation

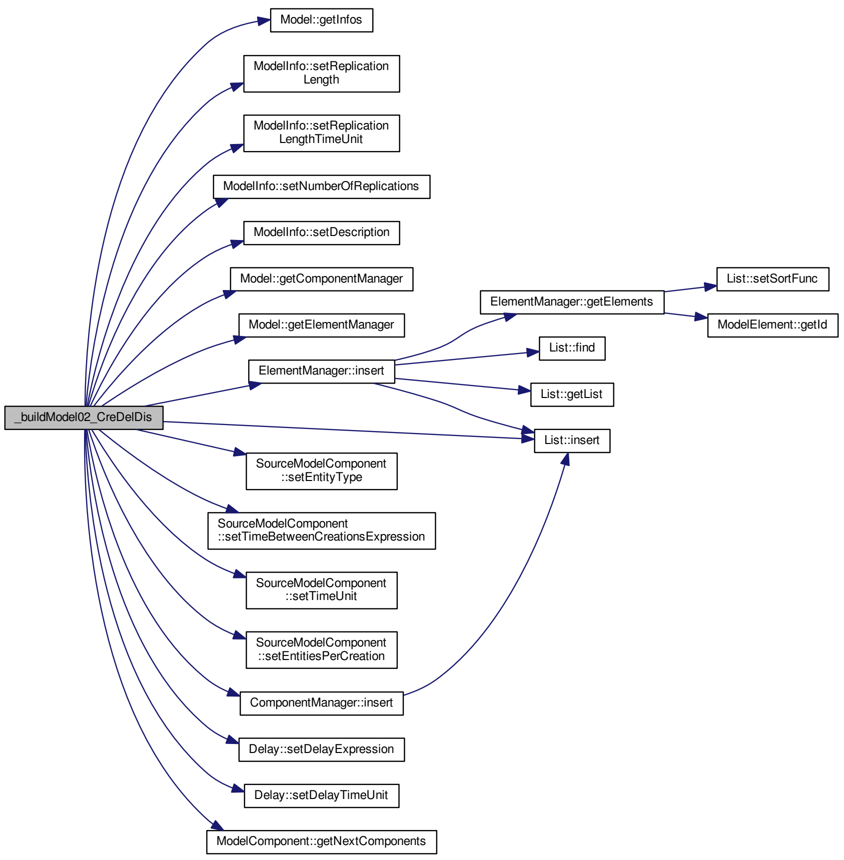
6.96.1.1 void _buildModel01_CreDelDis (**Model** ∗ *model*)

Here is the call graph for this function:


Model::getInfos
ModelInfo::setReplication Length
ModelInfo::setReplication LengthTimeUnit
ModelInfo::setNumberOfReplications
ModelInfo::setDescription
Model::getComponentManager
Model::getElementManager
ElementManager::insert
_buildModel01_CreDelDis
SourceModelComponent ::setEntityType
SourceModelComponent ::setTimeBetweenCreationsExpression
SourceModelComponent ::setTimeUnit
SourceModelComponent ::setEntitiesPerCreation
ComponentManager::insert
Delay::setDelayExpression
Delay::setDelayTimeUnit
ModelComponent::getNextComponents
ElementManager::getElements
List::find
List::getList
List::insert
List::setSortFunc
ModelElement::getId

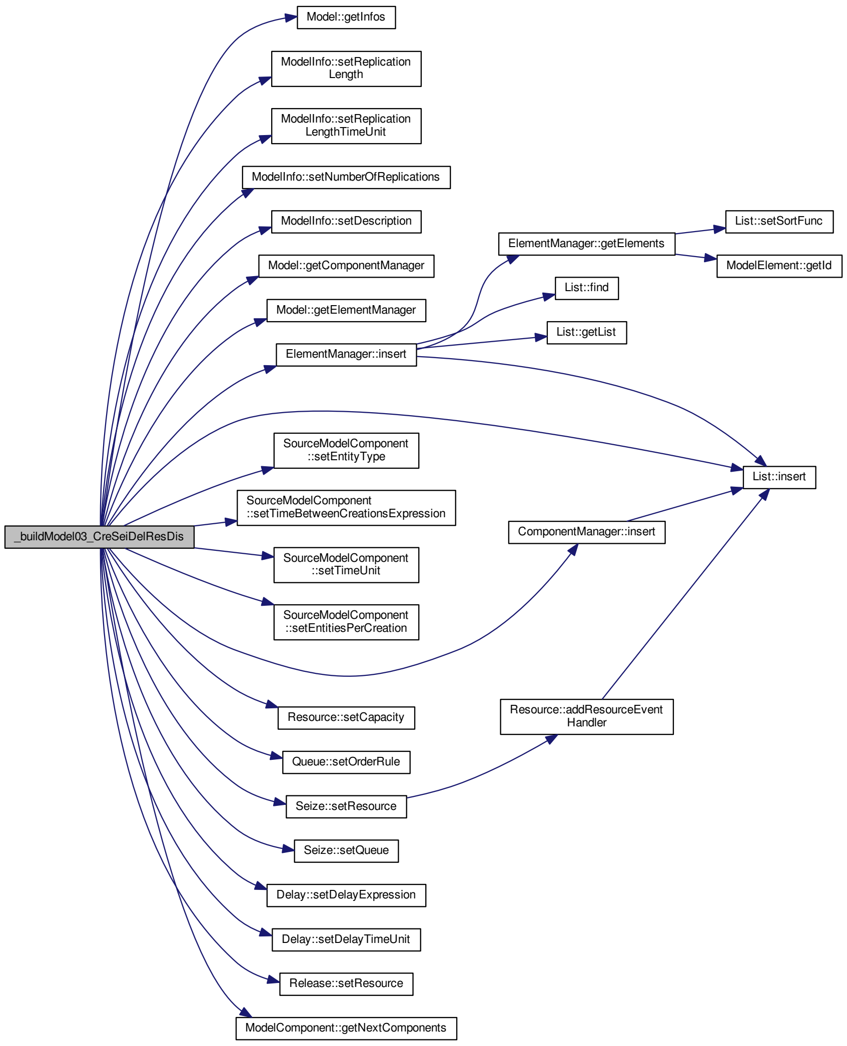
6.96.1.2 void _buildModel02_CreDelDis (Model ∗ *model*)

Here is the call graph for this function:

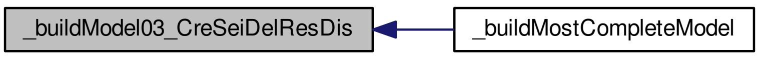

_buildModel02_CreDelDis
Model::getInfos
ModelInfo::setReplication Length
ModelInfo::setReplication LengthTimeUnit
ModelInfo::setNumberOfReplications
ModelInfo::setDescription
Model::getComponentManager
Model::getElementManager
ElementManager::insert
SourceModelComponent ::setEntityType
SourceModelComponent ::setTimeBetweenCreationsExpression
SourceModelComponent ::setTimeUnit
SourceModelComponent ::setEntitiesPerCreation
ComponentManager::insert
Delay::setDelayExpression
Delay::setDelayTimeUnit
ModelComponent::getNextComponents
ElementManager::getElements
List::find
List::getList
List::insert
List::setSortFunc
ModelElement::getId

6.96.1.3 void _buildModel03_CreSeiDelResDis (Model ∗ *model*)

Here is the call graph for this function:

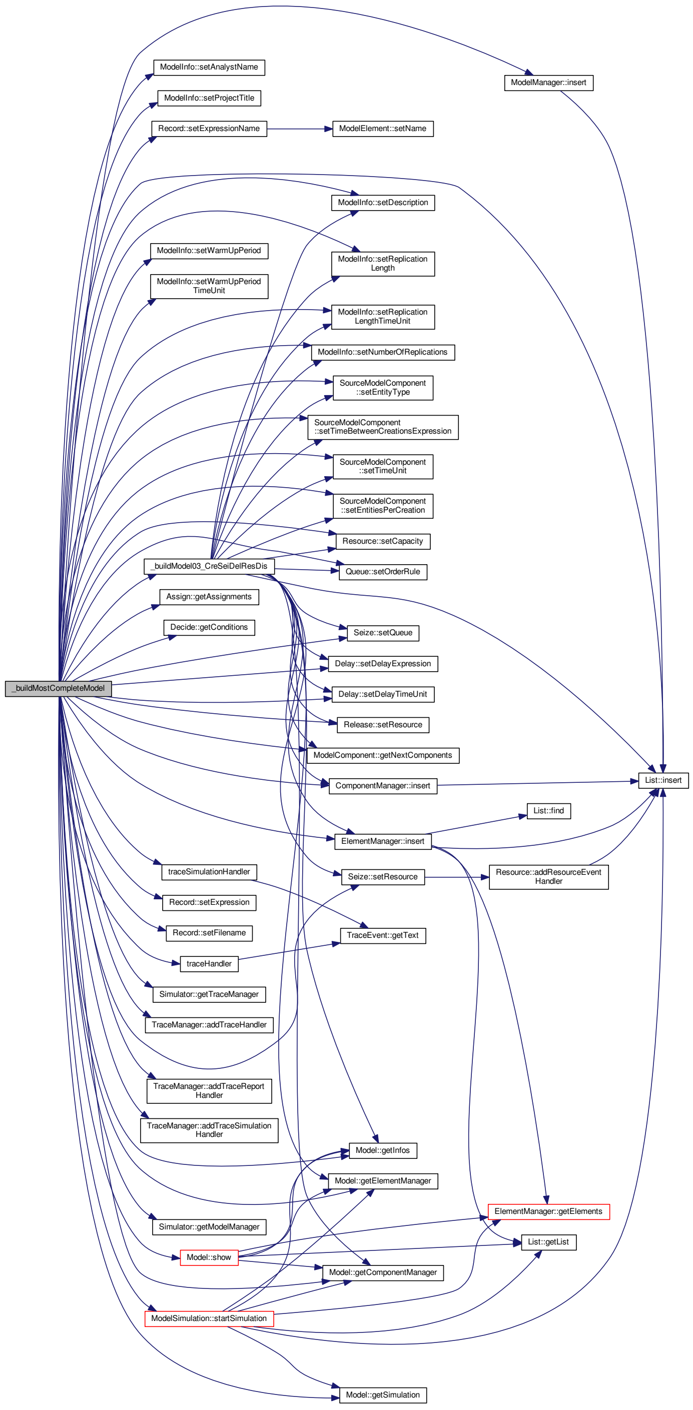

Model::getInfos
ModelInfo::setReplicationLength
ModelInfo::setReplicationLengthTimeUnit
ModelInfo::setNumberOfReplications
ModelInfo::setDescription
Model::getComponentManager
Model::getElementManager
ElementManager::insert
ElementManager::getElements
List::setSortFunc
ModelElement::getId
List::find
List::getList
List::insert
SourceModelComponent::setEntityType
SourceModelComponent::setTimeBetweenCreationsExpression
_buildModel03_CreSeiDelResDis
ComponentManager::insert
SourceModelComponent::setTimeUnit
SourceModelComponent::setEntitiesPerCreation
Resource::setCapacity
Resource::addResourceEventHandler
Queue::setOrderRule
Seize::setResource
Seize::setQueue
Delay::setDelayExpression
Delay::setDelayTimeUnit
Release::setResource
ModelComponent::getNextComponents

Here is the caller graph for this function:



6.96.1.4 void _buildMostCompleteModel (Model ∗ *model*)

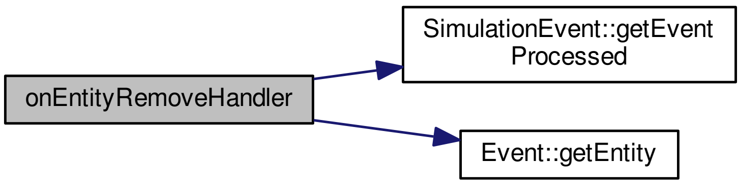
Here is the call graph for this function:



_buildMostCompleteModel
ModelInfo::setAnalystName
ModelManager::insert
ModelInfo::setProjectTitle
Record::setExpressionName
ModelElement::setName
ModelInfo::setDescription
ModelInfo::setWarmUpPeriod
ModelInfo::setReplication Length
ModelInfo::setWarmUpPeriod TimeUnit
ModelInfo::setReplication LengthTimeUnit
ModelInfo::setNumberOfReplications
SourceModelComponent ::setEntityType
SourceModelComponent ::setTimeBetweenCreationsExpression
SourceModelComponent ::setTimeUnit
SourceModelComponent ::setEntitiesPerCreation
Resource::setCapacity
_buildModel03_CreSeiDelResDis
Queue::setOrderRule
Assign::getAssignments
Decide::getConditions
Seize::setQueue
Delay::setDelayExpression
Delay::setDelayTimeUnit
Release::setResource
ModelComponent::getNextComponents
ComponentManager::insert
List::insert
List::find
ElementManager::insert
traceSimulationHandler
Seize::setResource
Resource::addResourceEvent Handler
Record::setExpression
Record::setFilename
TraceEvent::getText
traceHandler
Simulator::getTraceManager
TraceManager::addTraceHandler
TraceManager::addTraceReport Handler
TraceManager::addTraceSimulation Handler
Model::getInfos
Model::getElementManager
ElementManager::getElements
Simulator::getModelManager
List::getList
Model::show
Model::getComponentManager
ModelSimulation::startSimulation
Model::getSimulation

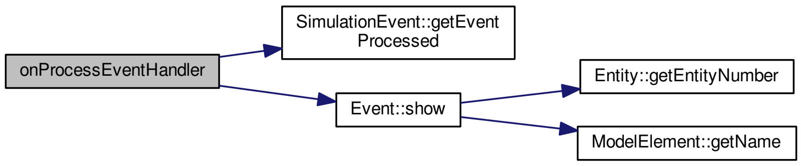
6.96.1.5 void onEntityRemoveHandler (**SimulationEvent** ∗ *re*)

Here is the call graph for this function:



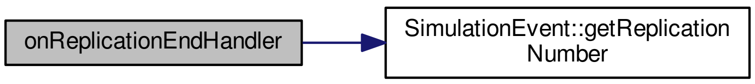
6.96.1.6 void onProcessEventHandler (**SimulationEvent** ∗ *re*)

Here is the call graph for this function:



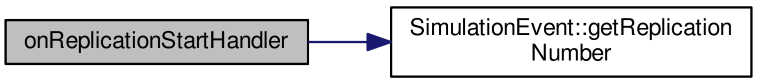
6.96.1.7 void onReplicationEndHandler (**SimulationEvent** ∗ *re*)

Here is the call graph for this function:



6.96.1.8 void onReplicationStartHandler (SimulationEvent ∗ *re*)

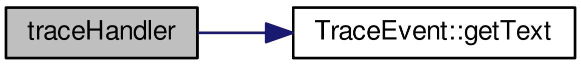
Here is the call graph for this function:


onReplicationStartHandler
SimulationEvent::getReplication Number

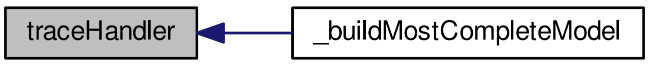
6.96.1.9 void onSimulationStartHandler (SimulationEvent ∗ *re*)

6.96.1.10 void traceHandler (TraceEvent *e*)

Here is the call graph for this function:

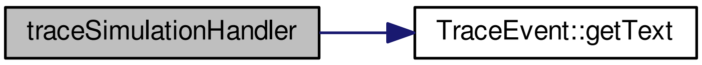


Here is the caller graph for this function:

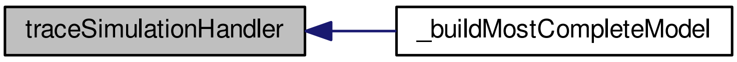


6.96.1.11 void traceSimulationHandler (TraceSimulationEvent *e*)

Here is the call graph for this function:



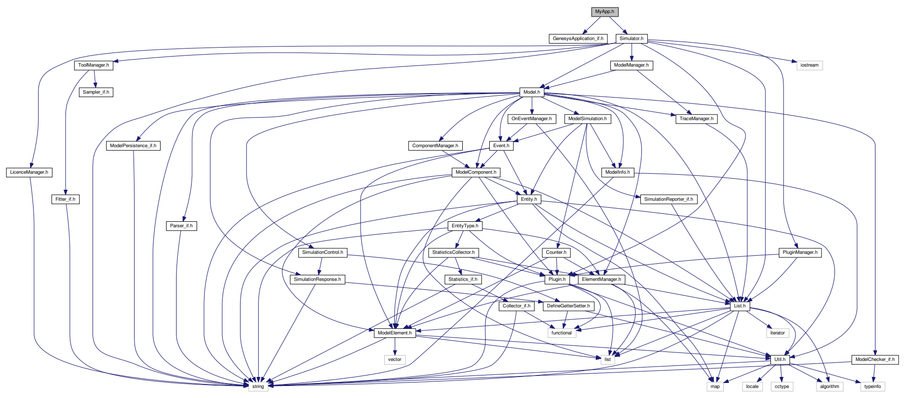
Here is the caller graph for this function:



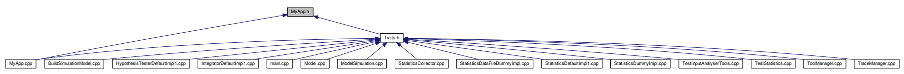
6.97 MyApp.h File Reference

```
#include "GenesysApplication_if.h"
#include "Simulator.h"
```

Include dependency graph for MyApp.h:



This graph shows which files directly or indirectly include this file:



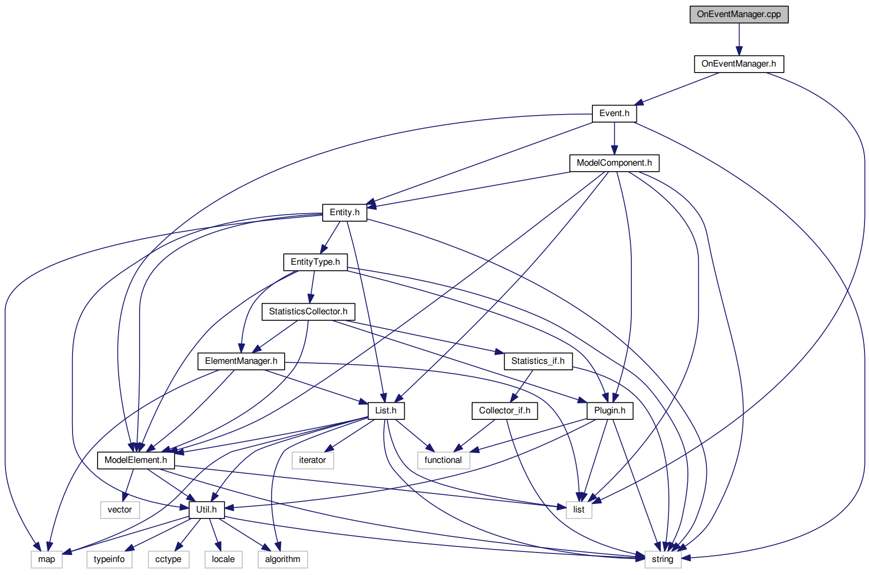
Classes

- class MyApp

6.98 OnEventManager.cpp File Reference

```
#include "OnEventManager.h"
```

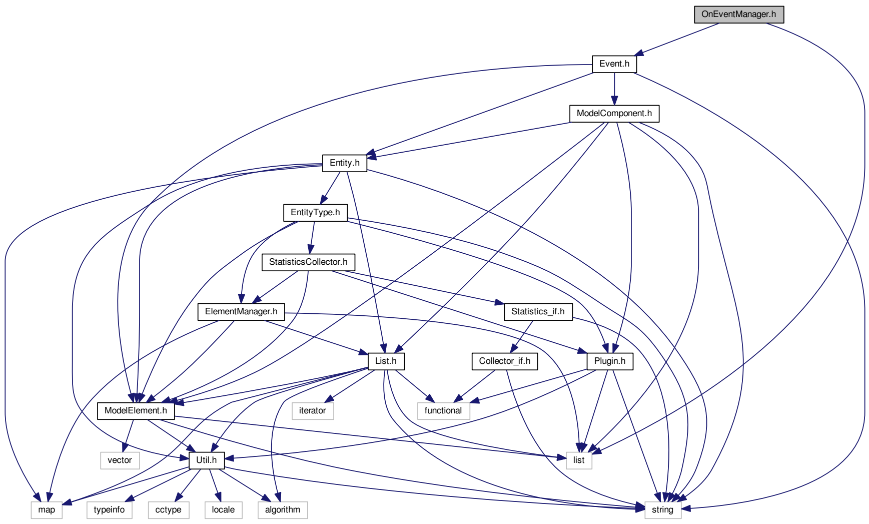
Include dependency graph for OnEventManager.cpp:



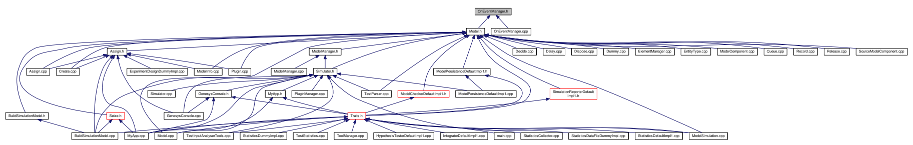
6.99 OnEventManager.h File Reference

```
#include <list>
#include "Event.h"
```

Include dependency graph for OnEventManager.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SimulationEvent
- class OnEventManager

Typedefs

- typedef void(∗ simulationEventHandler) (SimulationEvent ∗)

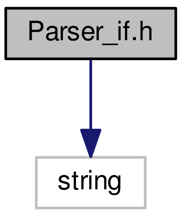
6.99.1 Typedef Documentation

6.99.1.1 typedef void(∗ simulationEventHandler) (SimulationEvent ∗)

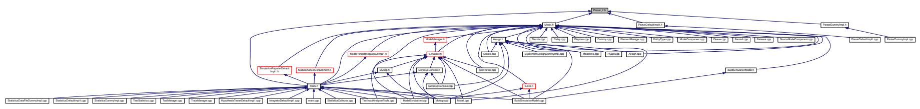
6.100 Parser_if.h File Reference

```
#include <string>
```

Include dependency graph for Parser_if.h:



This graph shows which files directly or indirectly include this file:



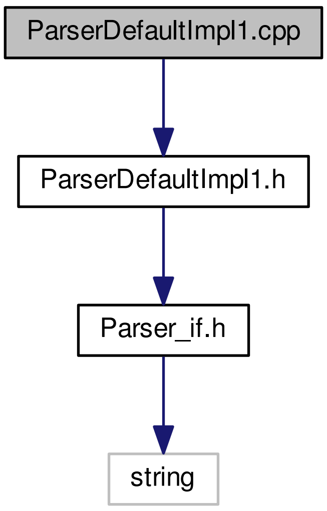
Classes

- class Parser_if

6.101 ParserDefaultImpl1.cpp File Reference

```
#include "ParserDefaultImpl1.h"
```

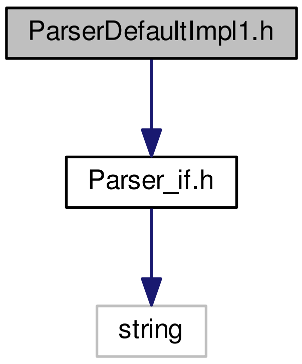
Include dependency graph for ParserDefaultImpl1.cpp:



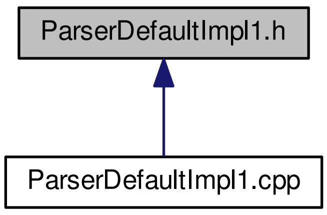
6.102 ParserDefaultImpl1.h File Reference

```
#include "Parser_if.h"
```

Include dependency graph for ParserDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



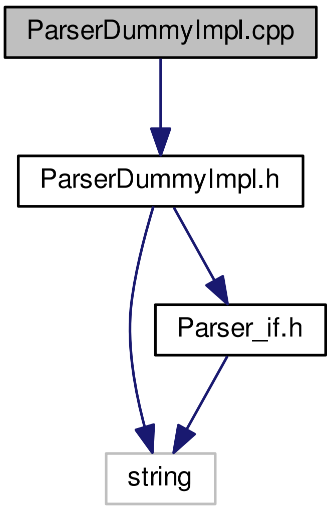
Classes

- class ParserDefaultImpl1

6.103 ParserDummyImpl.cpp File Reference

```
#include "ParserDummyImpl.h"
```

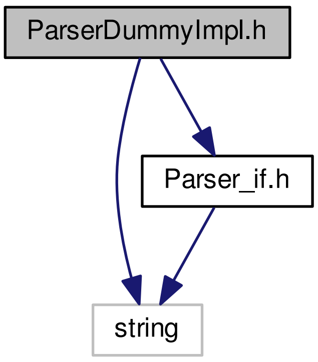
Include dependency graph for ParserDummyImpl.cpp:



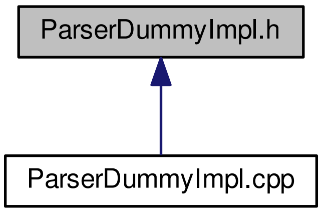
6.104 ParserDummyImpl.h File Reference

```
#include <string>
#include "Parser_if.h"
```

Include dependency graph for ParserDummyImpl.h:



This graph shows which files directly or indirectly include this file:



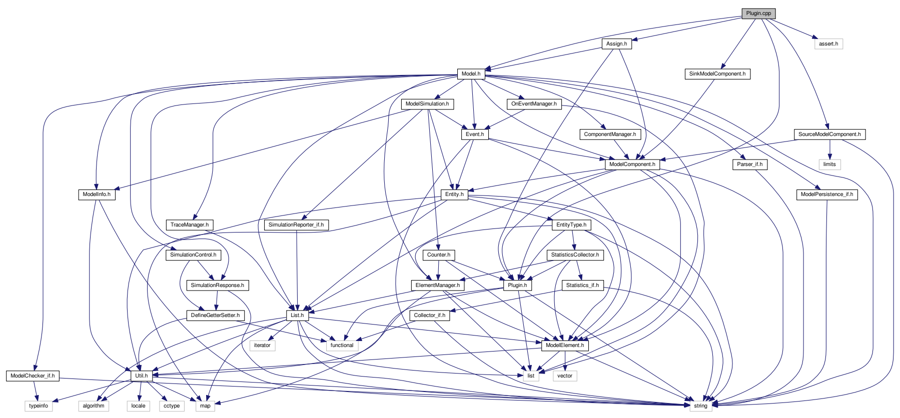
Classes

- class ParserDummyImpl

6.105 Plugin.cpp File Reference

```
#include "Plugin.h"
#include "Model.h"
#include "SourceModelComponent.h"
#include "SinkModelComponent.h"
#include "Assign.h"
#include <assert.h>
```

Include dependency graph for Plugin.cpp:

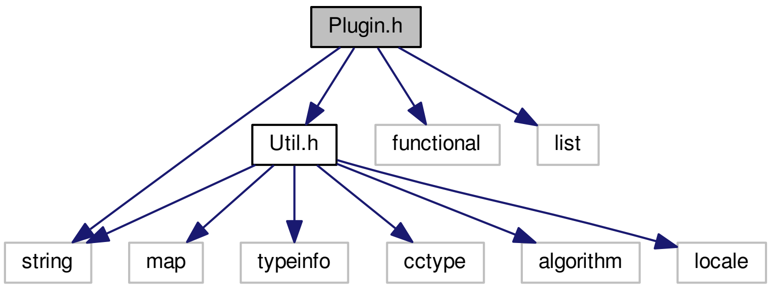


6.106 Plugin.h File Reference

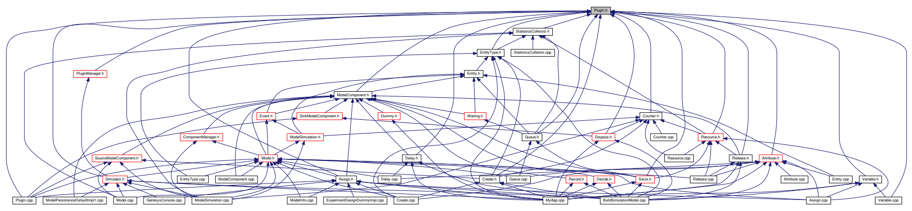
```
#include "Util.h"
```

```
#include <string>
#include <functional>
#include <list>
```

Include dependency graph for Plugin.h:



This graph shows which files directly or indirectly include this file:



Classes

- class PluginInformation
- class Plugin

Typedefs

- typedef ModelComponent ∗(∗ StaticLoaderComponentInstance) (Model ∗, std::map< std::string, std::string > ∗)
- typedef ModelElement ∗(∗ StaticLoaderElementInstance) (ElementManager ∗, std::map< std::string, std↩ ::string > ∗)
- typedef PluginInformation ∗(∗ StaticGetPluginInformation) ()

6.106.1 Typedef Documentation

6.106.1.1 typedef **PluginInformation**∗(∗ StaticGetPluginInformation) ()

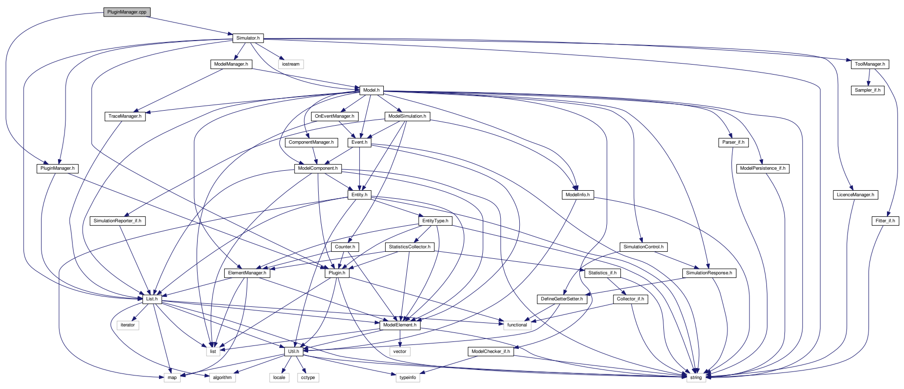
6.106.1.2 typedef **ModelComponent**∗(∗ StaticLoaderComponentInstance) (**Model** ∗, std::map< std::string, std::string > ∗)

6.106.1.3 typedef **ModelElement**∗(∗ StaticLoaderElementInstance) (**ElementManager** ∗, std::map< std::string, std::string > ∗)

6.107 PluginManager.cpp File Reference

```
#include "PluginManager.h"
#include "Simulator.h"
```

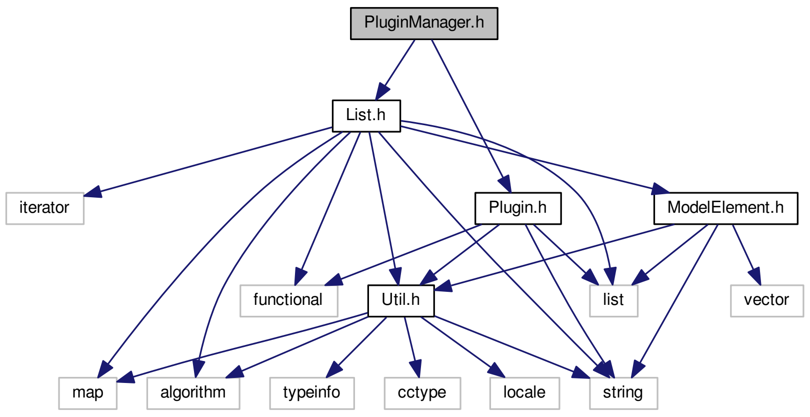
Include dependency graph for PluginManager.cpp:



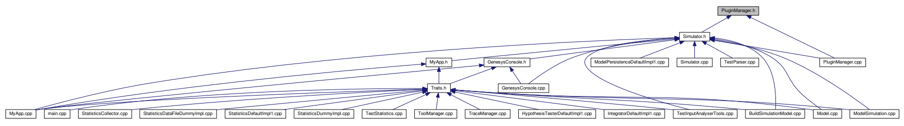
6.108 PluginManager.h File Reference

```
#include "List.h"
#include "Plugin.h"
```

Include dependency graph for PluginManager.h:



This graph shows which files directly or indirectly include this file:



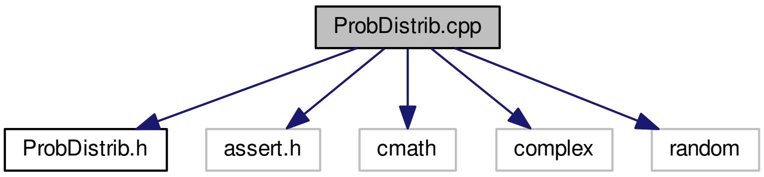
Classes

- class PluginManager

6.109 ProbDistrib.cpp File Reference

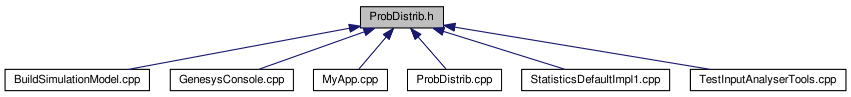
```
#include "ProbDistrib.h"
#include <assert.h>
#include <cmath>
#include <complex>
#include <random>
```

Include dependency graph for ProbDistrib.cpp:



6.110 ProbDistrib.h File Reference

This graph shows which files directly or indirectly include this file:



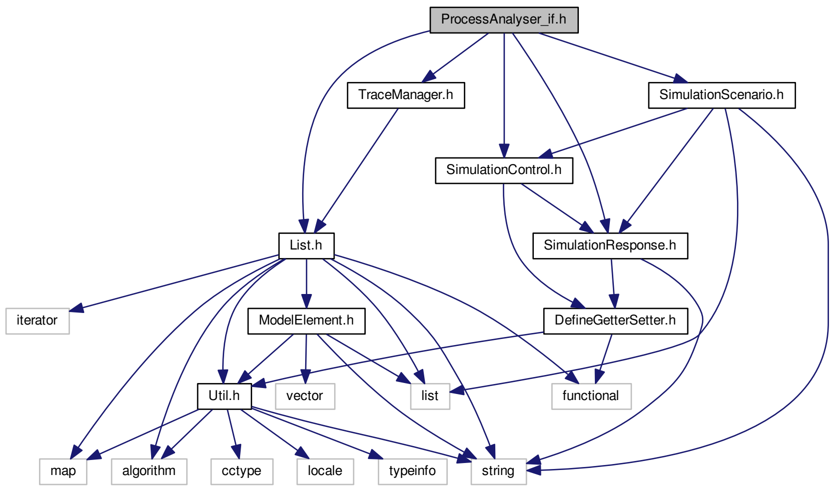
Classes

- class ProbDistrib

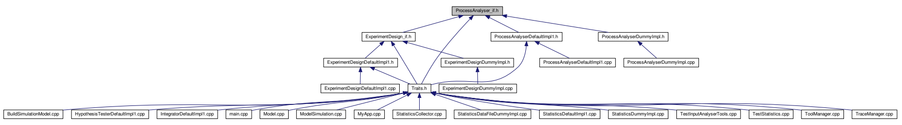
6.111 ProcessAnalyser_if.h File Reference

```
#include "List.h"
#include "SimulationScenario.h"
#include "SimulationControl.h"
#include "SimulationResponse.h"
#include "TraceManager.h"
```

Include dependency graph for ProcessAnalyser_if.h:



This graph shows which files directly or indirectly include this file:


ProcessAnalyser_if.h
ExperimentDesign_if.h
ProcessAnalyserDefaultImpl1.h
ProcessAnalyserDummyImpl.h
ExperimentDesignDefaultImpl1.h
ExperimentDesignDummyImpl.h
ProcessAnalyserDefaultImpl1.cpp
ProcessAnalyserDummyImpl.cpp
ExperimentDesignDefaultImpl1.cpp
Traits.h
ExperimentDesignDummyImpl.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

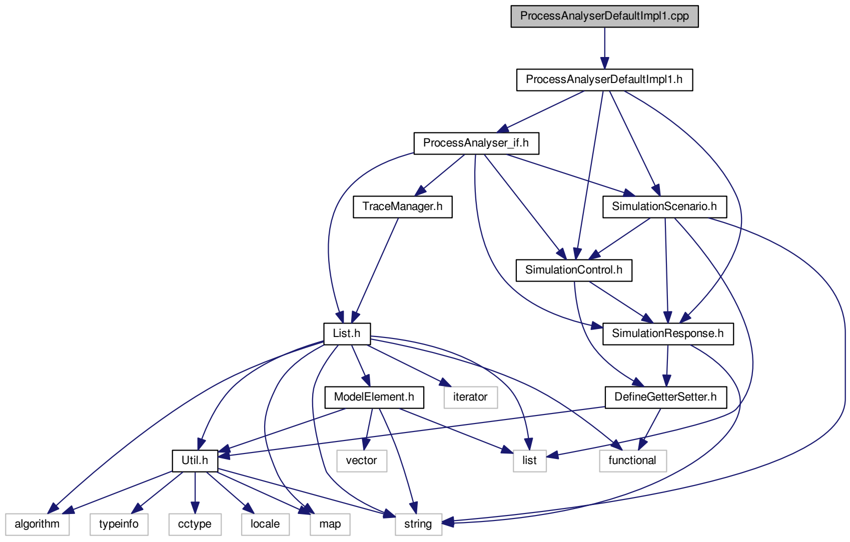
Classes

- class ProcessAnalyser_if

6.112 ProcessAnalyserDefaultImpl1.cpp File Reference

```
#include "ProcessAnalyserDefaultImpl1.h"
```

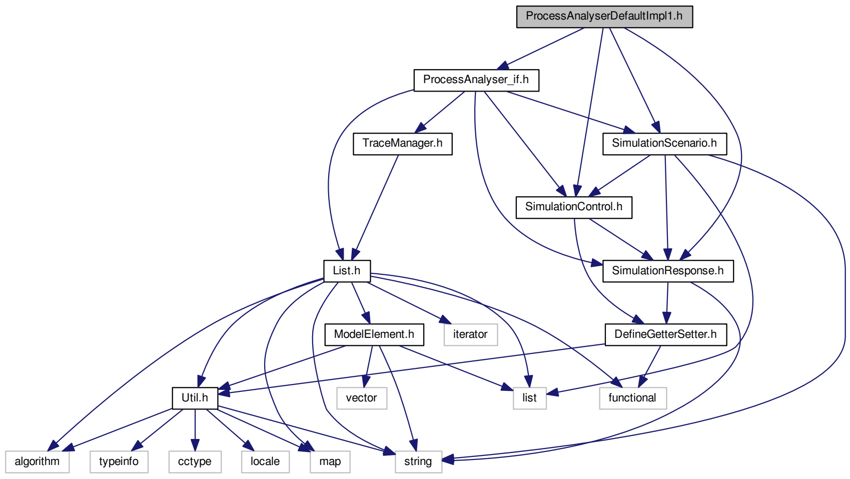
Include dependency graph for ProcessAnalyserDefaultImpl1.cpp:



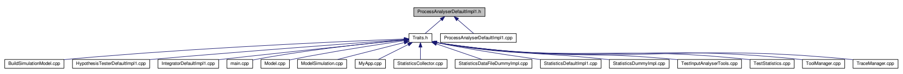
6.113 ProcessAnalyserDefaultImpl1.h File Reference

```
#include "ProcessAnalyser_if.h"
#include "SimulationScenario.h"
#include "SimulationResponse.h"
#include "SimulationControl.h"
```

Include dependency graph for ProcessAnalyserDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:


ProcessAnalyserDefaultImpl1.h
Traits.h
ProcessAnalyserDefaultImpl1.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

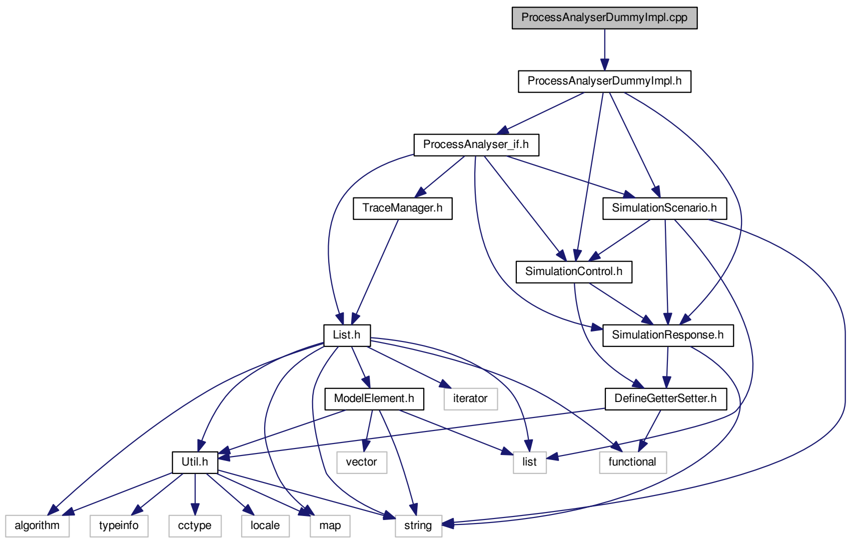
Classes

- class ProcessAnalyserDefaultImpl1

6.114 ProcessAnalyserDummyImpl.cpp File Reference

```
#include "ProcessAnalyserDummyImpl.h"
```

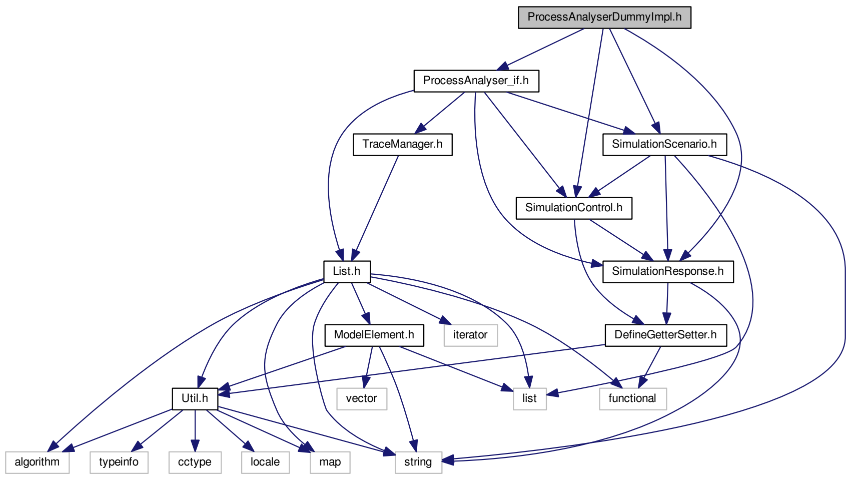
Include dependency graph for ProcessAnalyserDummyImpl.cpp:



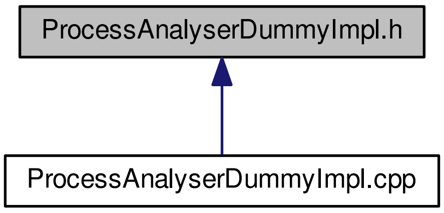
6.115 ProcessAnalyserDummyImpl.h File Reference

```
#include "ProcessAnalyser_if.h"
#include "SimulationScenario.h"
#include "SimulationResponse.h"
#include "SimulationControl.h"
```

Include dependency graph for ProcessAnalyserDummyImpl.h:



This graph shows which files directly or indirectly include this file:



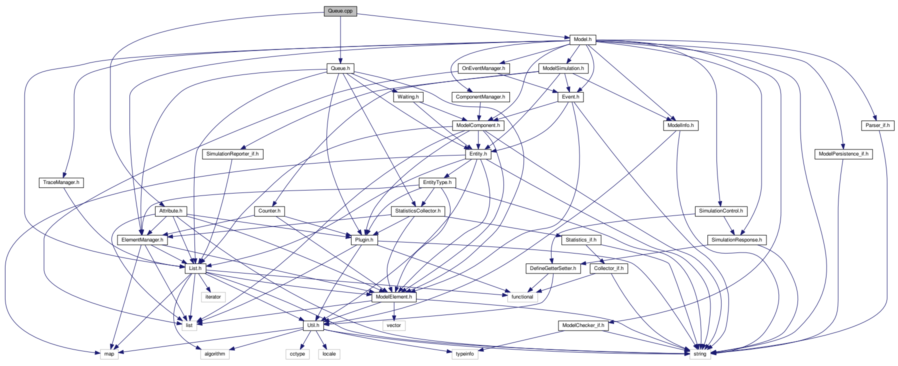
Classes

- class ProcessAnalyserDummyImpl

6.116 Queue.cpp File Reference

```
#include "Queue.h"
#include "Model.h"
#include "Attribute.h"
```

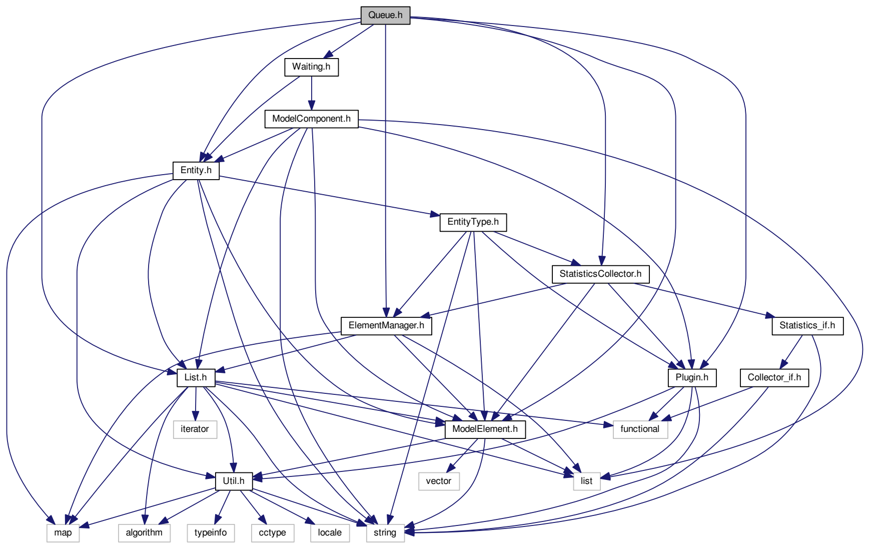
Include dependency graph for Queue.cpp:



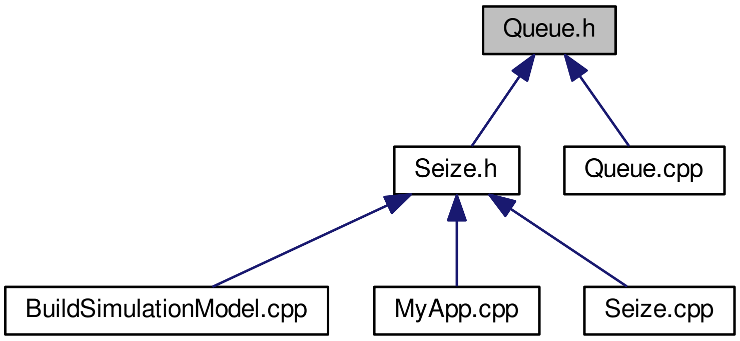
6.117 Queue.h File Reference

```
#include "ModelElement.h"
#include "List.h"
#include "Entity.h"
#include "Waiting.h"
#include "ElementManager.h"
#include "StatisticsCollector.h"
#include "Plugin.h"
```

Include dependency graph for Queue.h:



This graph shows which files directly or indirectly include this file:



Classes

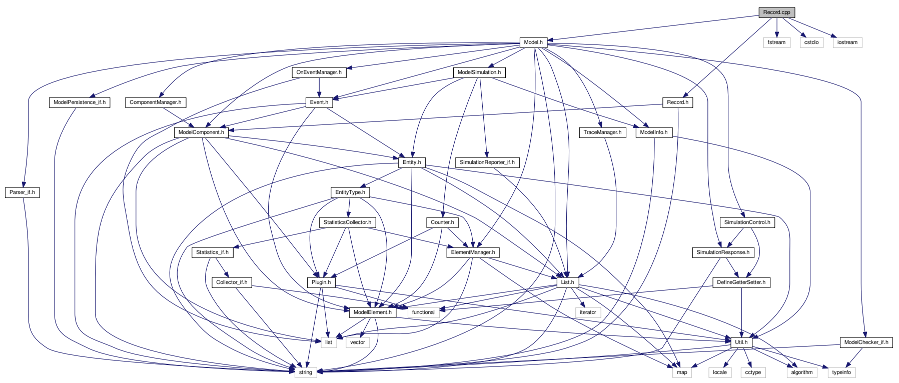
- class Queue

6.118 README.md File Reference

6.119 Record.cpp File Reference

```
#include "Record.h"
#include "Model.h"
#include <fstream>
#include <cstdio>
#include <iostream>
```

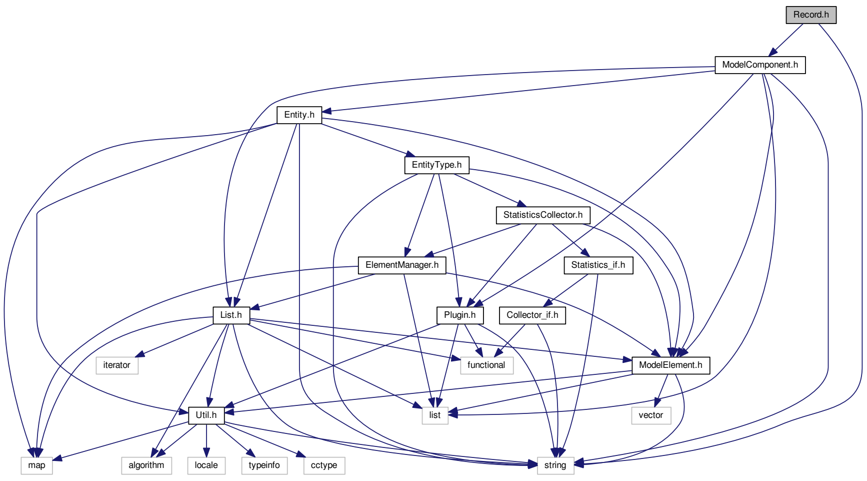
Include dependency graph for Record.cpp:



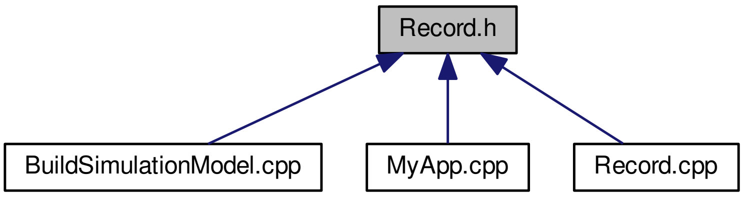
6.120 Record.h File Reference

```
#include "ModelComponent.h"
#include <string>
```

Include dependency graph for Record.h:



This graph shows which files directly or indirectly include this file:



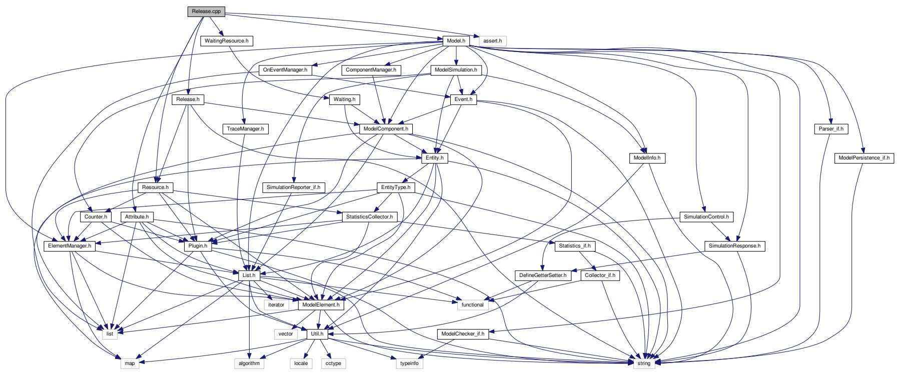
Classes

- class Record

6.121 Release.cpp File Reference

```
#include "Release.h"
#include "Model.h"
#include "WaitingResource.h"
#include "Resource.h"
#include "Attribute.h"
#include <assert.h>
```

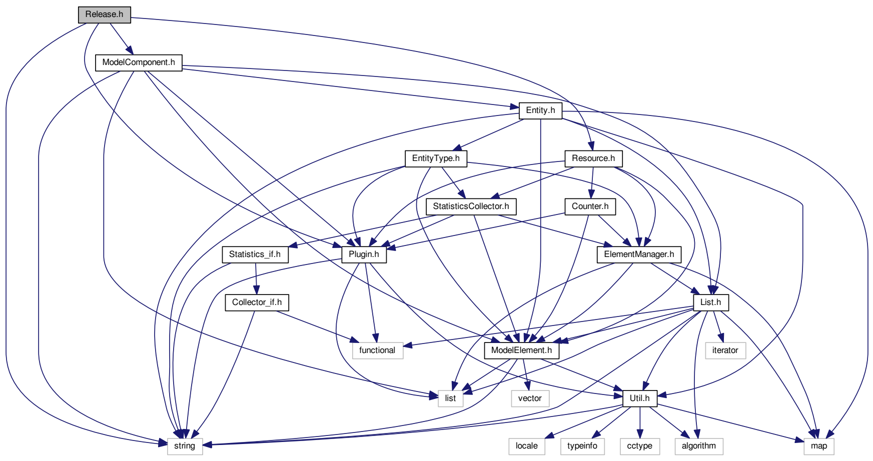
Include dependency graph for Release.cpp:



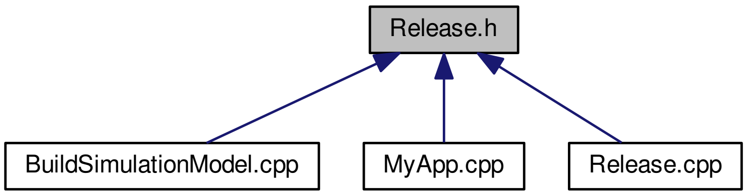
6.122 Release.h File Reference

```
#include <string>
#include "ModelComponent.h"
#include "Resource.h"
#include "Plugin.h"
```

Include dependency graph for Release.h:



This graph shows which files directly or indirectly include this file:



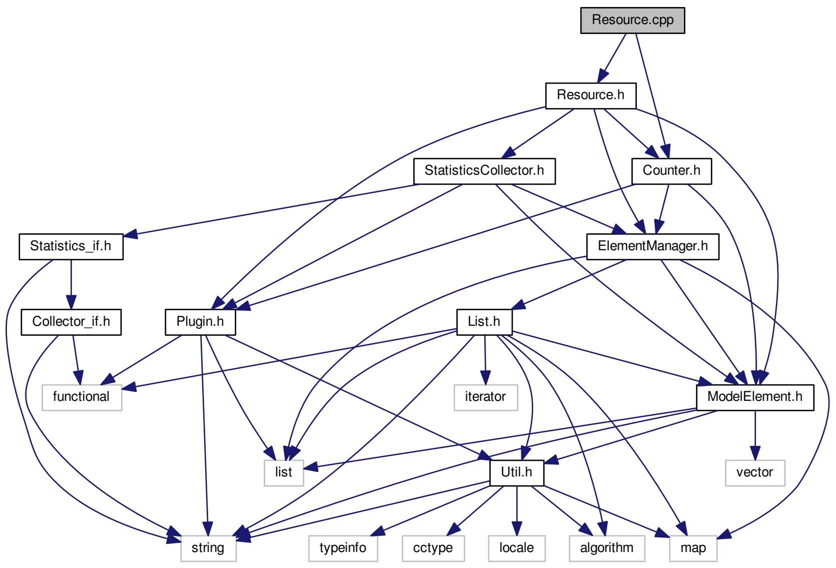
Classes

- class Release

6.123 Resource.cpp File Reference

```
#include "Resource.h"
#include "Counter.h"
```

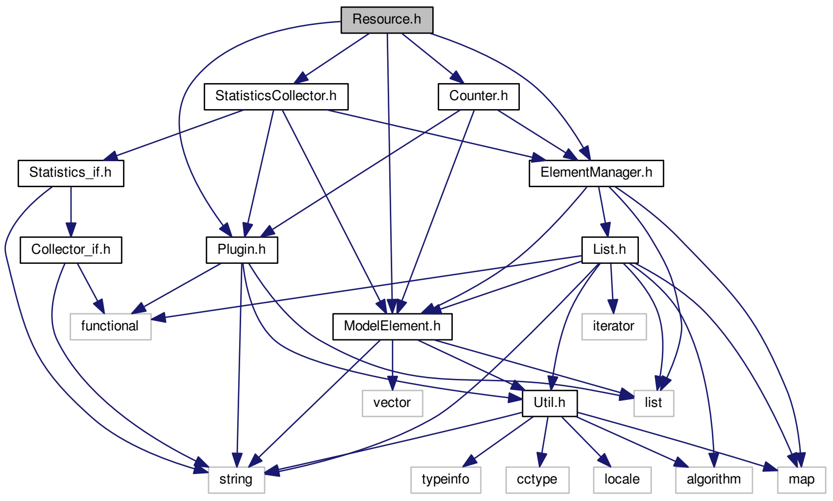
Include dependency graph for Resource.cpp:



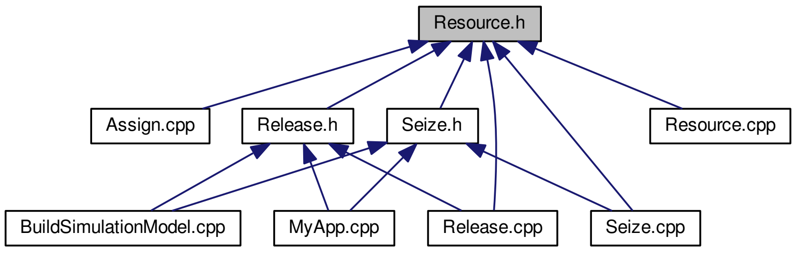
6.124 Resource.h File Reference

```
#include "ModelElement.h"
#include "StatisticsCollector.h"
#include "ElementManager.h"
#include "Counter.h"
#include "Plugin.h"
```

Include dependency graph for Resource.h:



This graph shows which files directly or indirectly include this file:

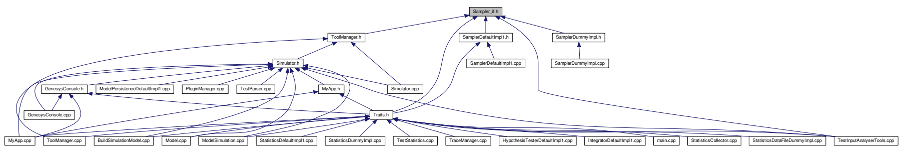


Classes

- class Resource

6.125 Sampler_if.h File Reference

This graph shows which files directly or indirectly include this file:



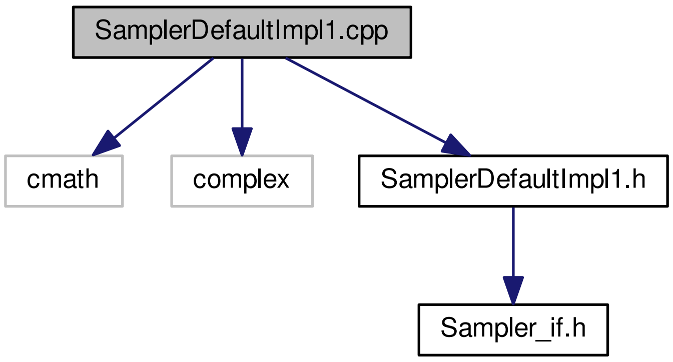
Classes

- class Sampler_if
- struct Sampler_if::RNG_Parameters

6.126 SamplerDefaultImpl1.cpp File Reference

```
#include <cmath>
#include <complex>
#include "SamplerDefaultImpl1.h"
```

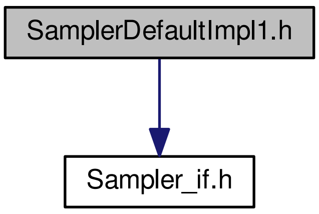
Include dependency graph for SamplerDefaultImpl1.cpp:



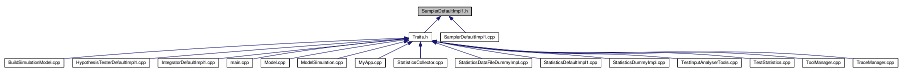
6.127 SamplerDefaultImpl1.h File Reference

```
#include "Sampler_if.h"
```

Include dependency graph for SamplerDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



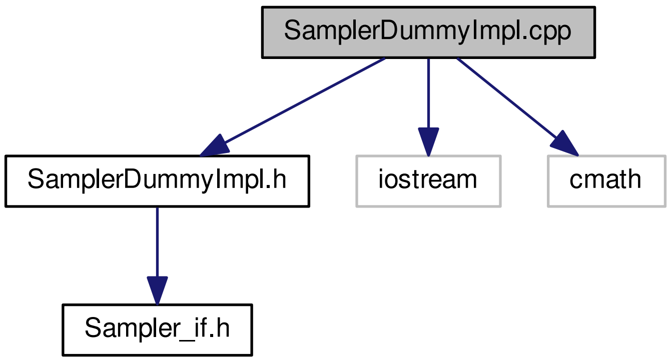
Classes

- class SamplerDefaultImpl1
- struct SamplerDefaultImpl1::DefaultImpl1RNG_Parameters

6.128 SamplerDummyImpl.cpp File Reference

```
#include "SamplerDummyImpl.h"
#include <iostream>
#include <cmath>
```

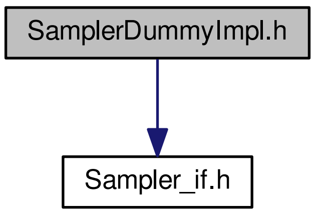
Include dependency graph for SamplerDummyImpl.cpp:



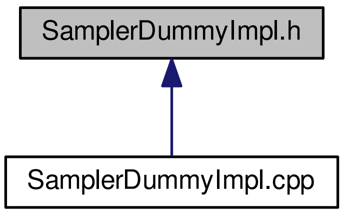
6.129 SamplerDummyImpl.h File Reference

```
#include "Sampler_if.h"
```

Include dependency graph for SamplerDummyImpl.h:



This graph shows which files directly or indirectly include this file:



Classes

- class SamplerDummyImpl
- struct SamplerDummyImpl::MyRNG_Parameters

6.130 ScenarioExperiment_if.h File Reference

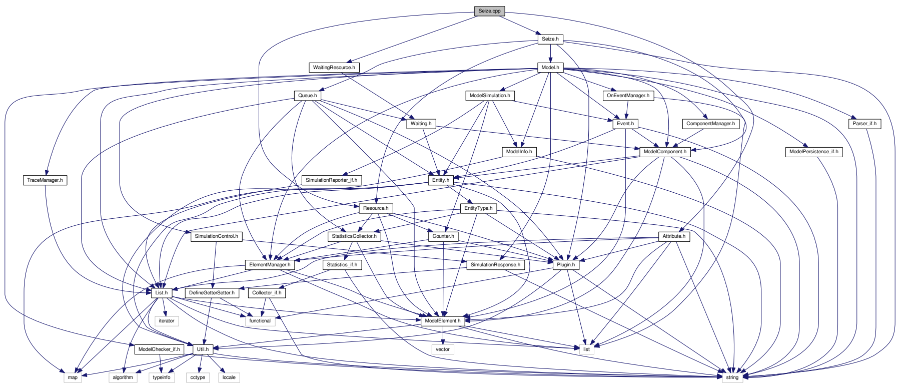
Classes

- class ScenarioExperiment_if

6.131 Seize.cpp File Reference

```
#include "Seize.h"
#include "WaitingResource.h"
#include "Resource.h"
#include "Attribute.h"
```

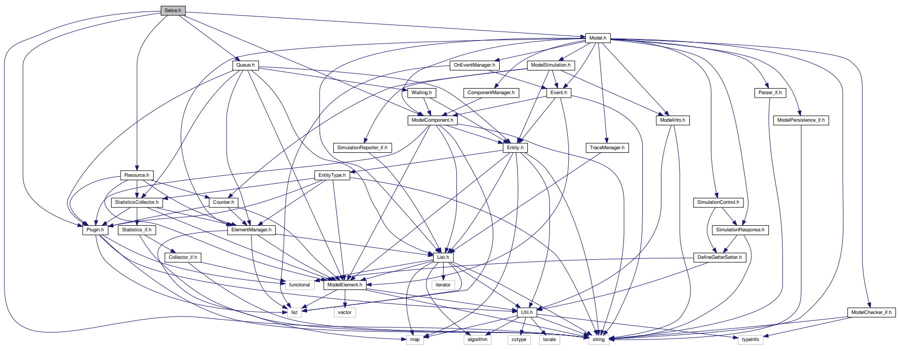
Include dependency graph for Seize.cpp:



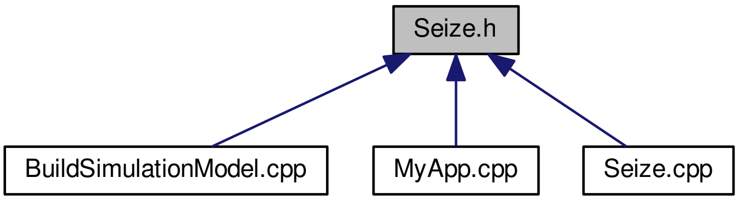
6.132 Seize.h File Reference

```
#include <string>
#include "ModelComponent.h"
#include "Model.h"
#include "Resource.h"
#include "Queue.h"
#include "Plugin.h"
```

Include dependency graph for Seize.h:



This graph shows which files directly or indirectly include this file:



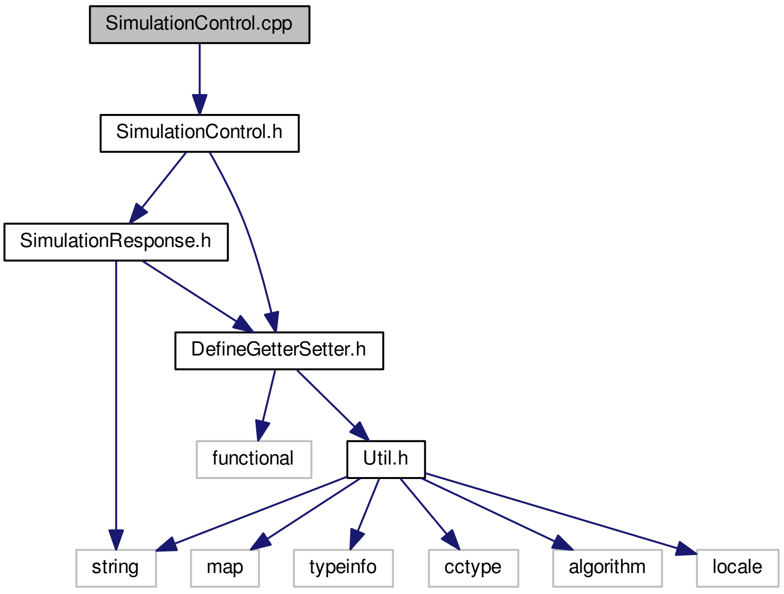
Classes

- class Seize

6.133 SimulationControl.cpp File Reference

```
#include "SimulationControl.h"
```

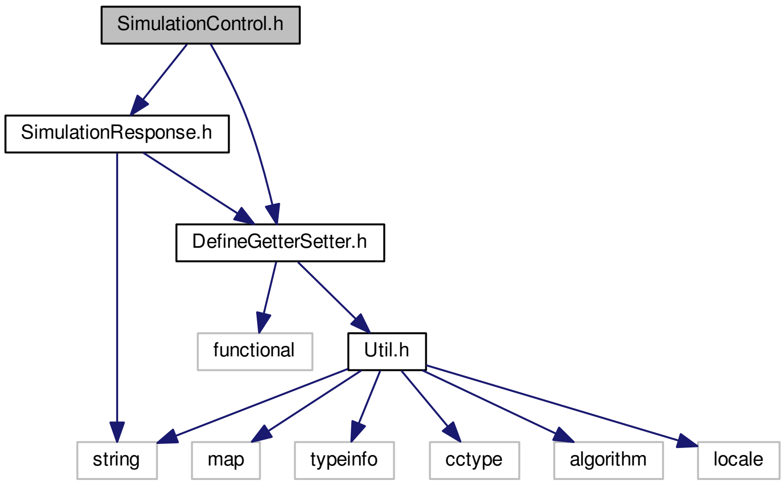
Include dependency graph for SimulationControl.cpp:



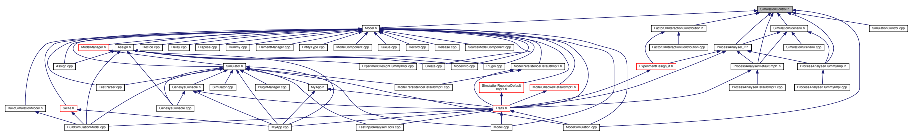
6.134 SimulationControl.h File Reference

```
#include "SimulationResponse.h"
#include "DefineGetterSetter.h"
```

Include dependency graph for SimulationControl.h:



This graph shows which files directly or indirectly include this file:



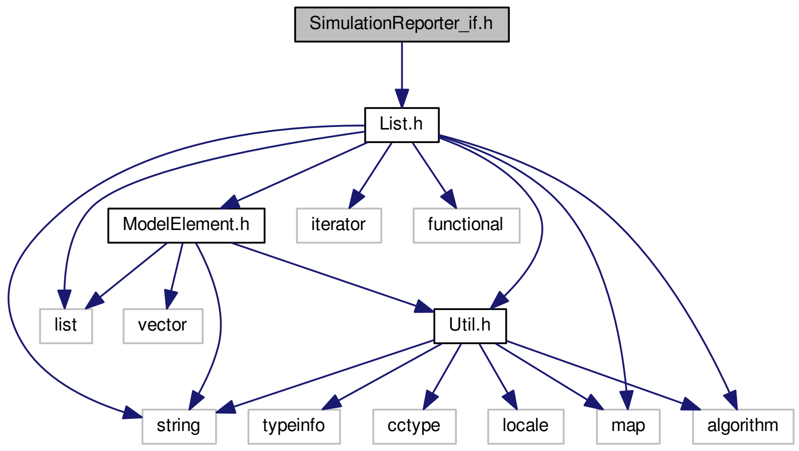
Classes

- class SimulationControl

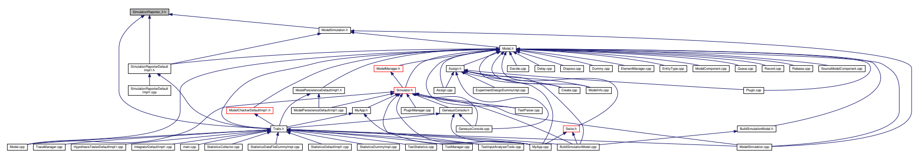
6.135 SimulationReporter_if.h File Reference

```
#include "List.h"
```

Include dependency graph for SimulationReporter_if.h:



This graph shows which files directly or indirectly include this file:



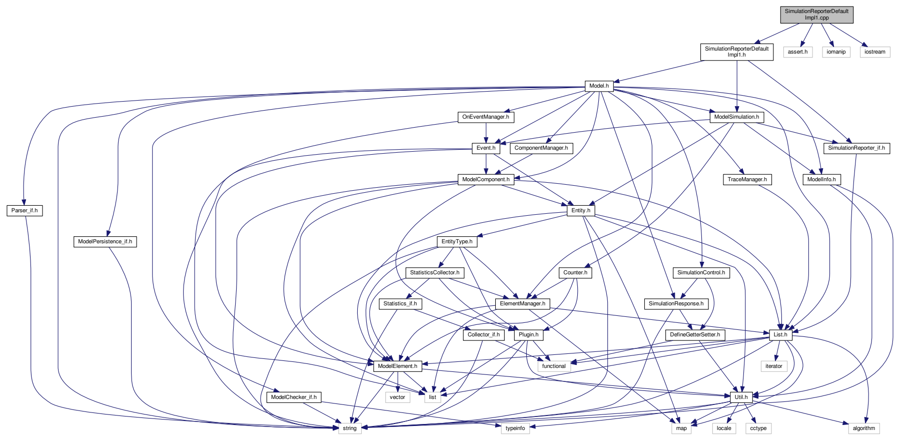
Classes

- class SimulationReporter_if

6.136 SimulationReporterDefaultImpl1.cpp File Reference

```
#include "SimulationReporterDefaultImpl1.h"
#include <assert.h>
#include <iomanip>
#include <iostream>
```

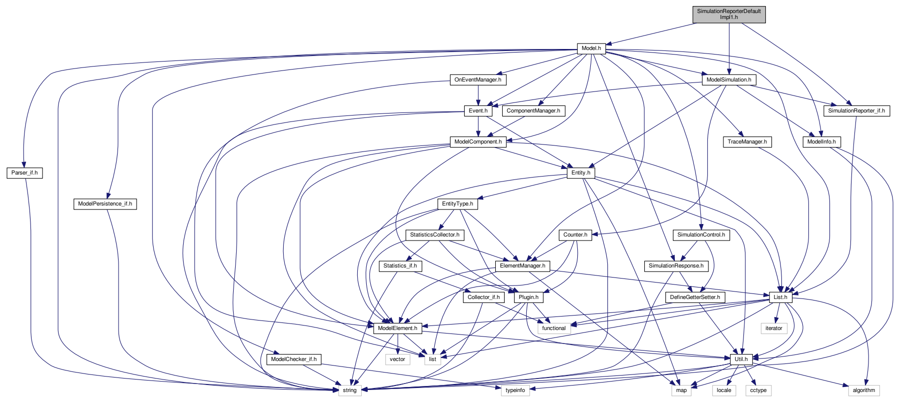
Include dependency graph for SimulationReporterDefaultImpl1.cpp:



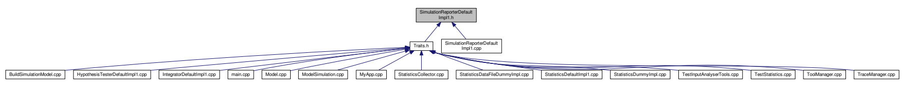
6.137 SimulationReporterDefaultImpl1.h File Reference

```
#include "SimulationReporter_if.h"
#include "ModelSimulation.h"
#include "Model.h"
```

Include dependency graph for SimulationReporterDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:


SimulationReporterDefault Impl1.h
Traits.h
SimulationReporterDefault Impl1.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

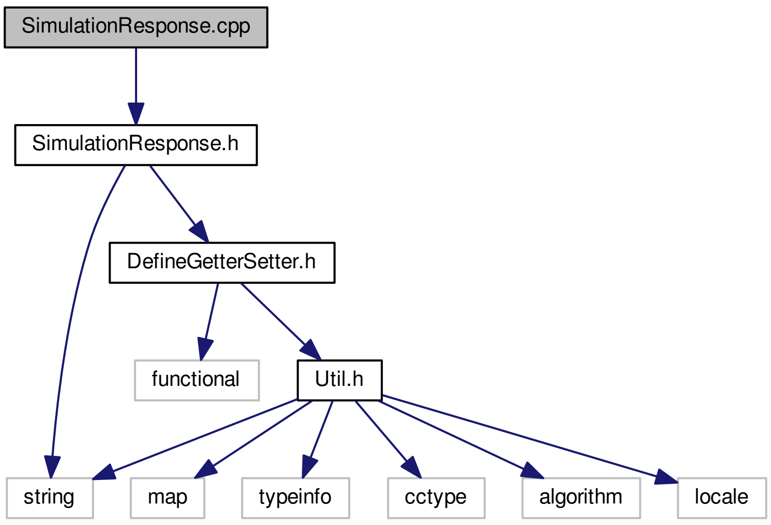
Classes

- class SimulationReporterDefaultImpl1

6.138 SimulationResponse.cpp File Reference

```
#include "SimulationResponse.h"
```

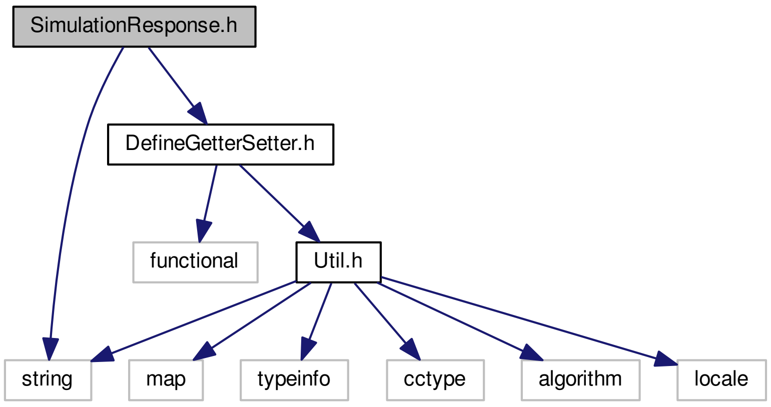
Include dependency graph for SimulationResponse.cpp:



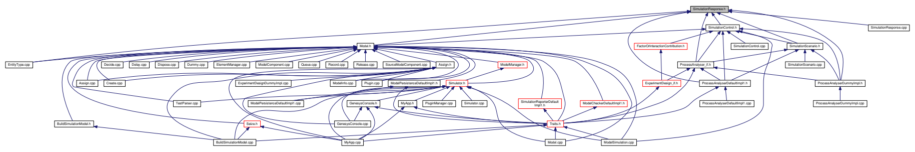
6.139 SimulationResponse.h File Reference

```
#include <string>
#include "DefineGetterSetter.h"
```

Include dependency graph for SimulationResponse.h:



This graph shows which files directly or indirectly include this file:



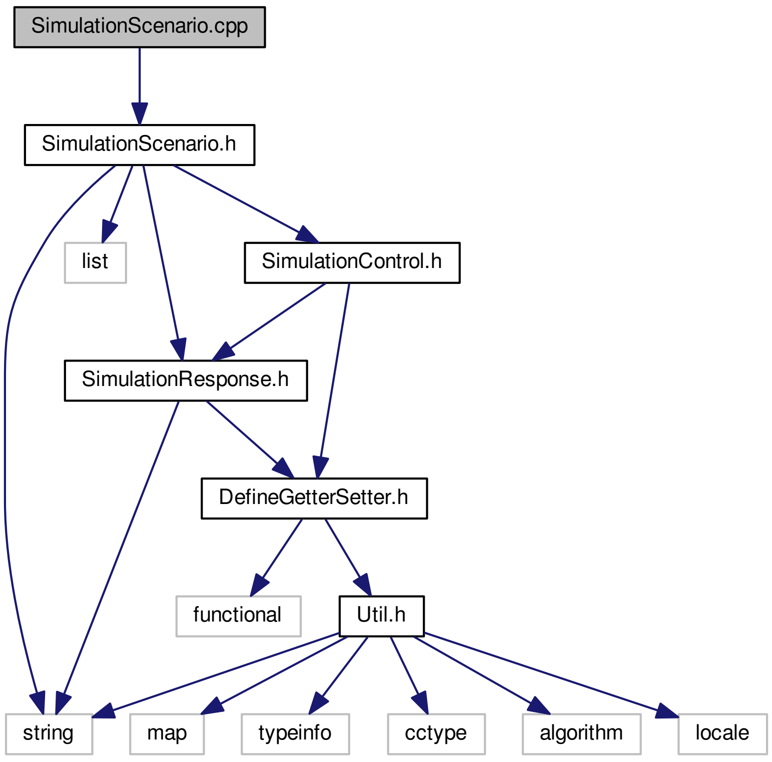
Classes

- class SimulationResponse

6.140 SimulationScenario.cpp File Reference

```
#include "SimulationScenario.h"
```

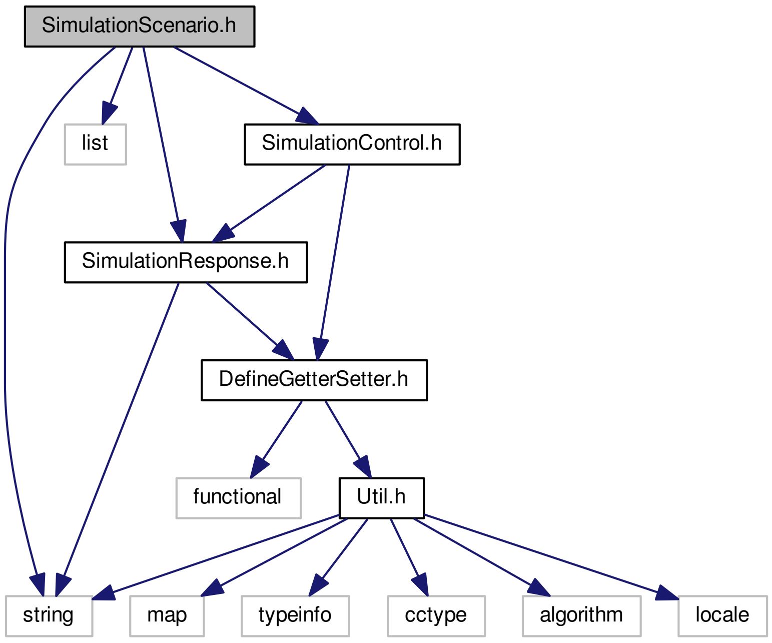
Include dependency graph for SimulationScenario.cpp:



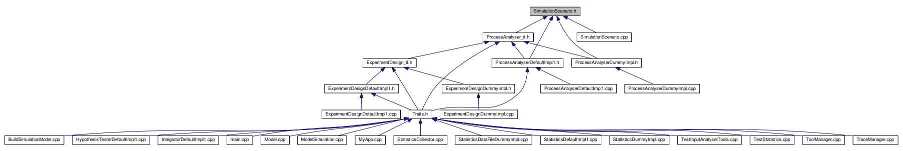
6.141 SimulationScenario.h File Reference

```
#include <string>
#include <list>
#include "SimulationResponse.h"
#include "SimulationControl.h"
```

Include dependency graph for SimulationScenario.h:



This graph shows which files directly or indirectly include this file:


SimulationScenario.h
ProcessAnalyser_if.h
SimulationScenario.cpp
ExperimentDesign_if.h
ProcessAnalyserDefaultImpl1.h
ProcessAnalyserDummyImpl.h
ExperimentDesignDefaultImpl1.h
ExperimentDesignDummyImpl.h
ProcessAnalyserDefaultImpl1.cpp
ProcessAnalyserDummyImpl.cpp
ExperimentDesignDefaultImpl1.cpp
Traits.h
ExperimentDesignDummyImpl.cpp
BuildSimulationModel.cpp
HypothesisTesterDefaultImpl1.cpp
IntegratorDefaultImpl1.cpp
main.cpp
Model.cpp
ModelSimulation.cpp
MyApp.cpp
StatisticsCollector.cpp
StatisticsDataFileDummyImpl.cpp
StatisticsDefaultImpl1.cpp
StatisticsDummyImpl.cpp
TestInputAnalyserTools.cpp
TestStatistics.cpp
ToolManager.cpp
TraceManager.cpp

Classes

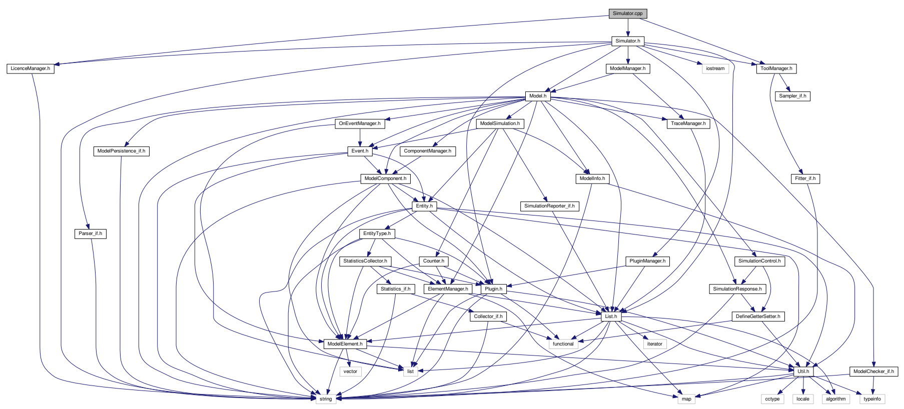
- class SimulationScenario

6.142 Simulator.cpp File Reference

```
#include "Simulator.h"
```

```
#include "LicenceManager.h"
#include "ToolManager.h"
```

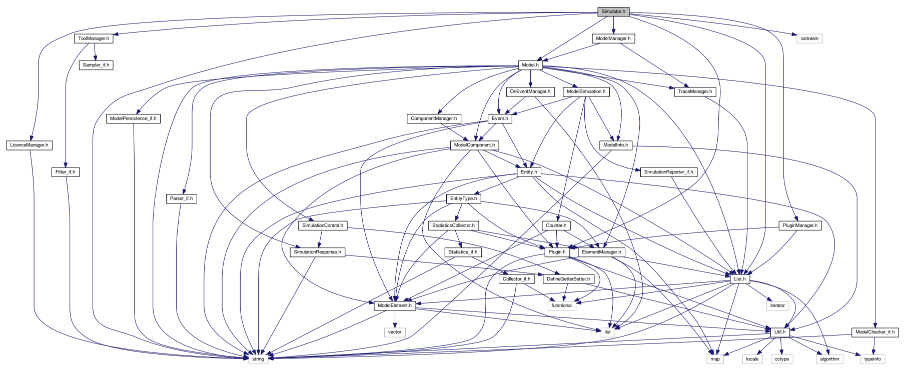
Include dependency graph for Simulator.cpp:



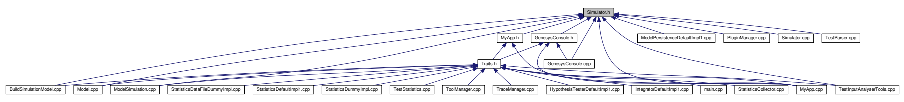
6.143 Simulator.h File Reference

```
#include <string>
#include <iostream>
#include "Model.h"
#include "Plugin.h"
#include "List.h"
#include "LicenceManager.h"
#include "PluginManager.h"
#include "ModelManager.h"
#include "ToolManager.h"
```

Include dependency graph for Simulator.h:



This graph shows which files directly or indirectly include this file:



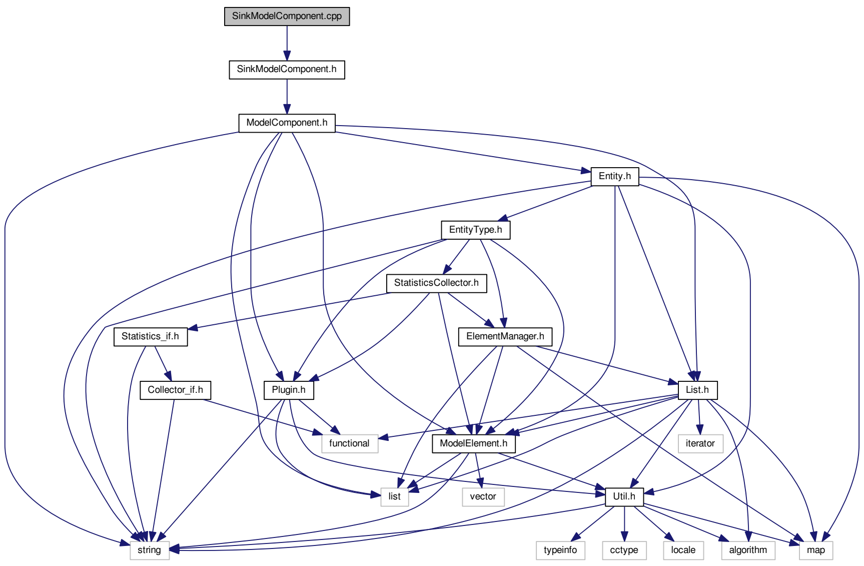
Classes

- class Simulator

6.144 SinkModelComponent.cpp File Reference

```
#include "SinkModelComponent.h"
```

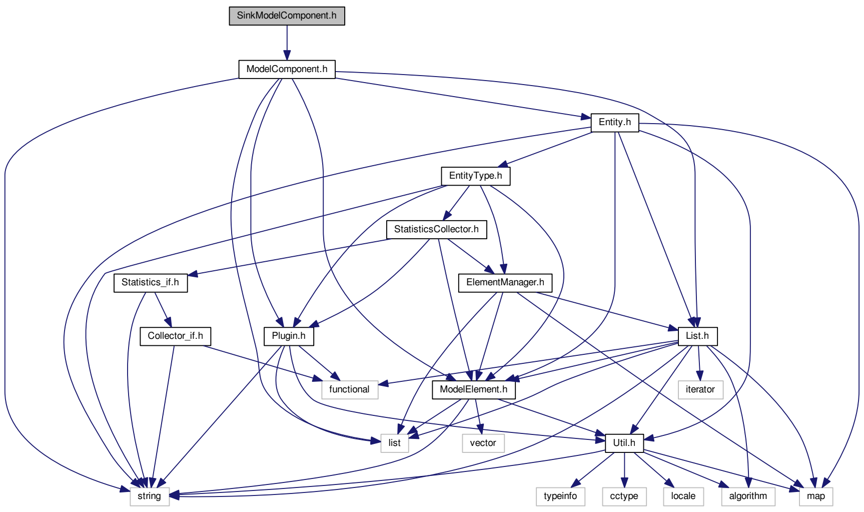
Include dependency graph for SinkModelComponent.cpp:



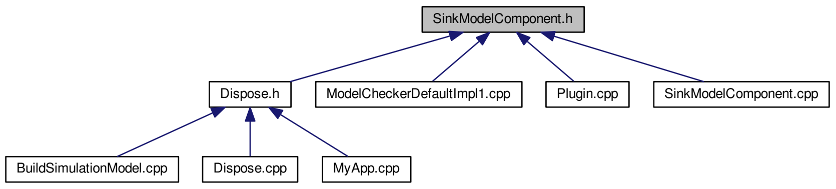
6.145 SinkModelComponent.h File Reference

```
#include "ModelComponent.h"
```

Include dependency graph for SinkModelComponent.h:



This graph shows which files directly or indirectly include this file:



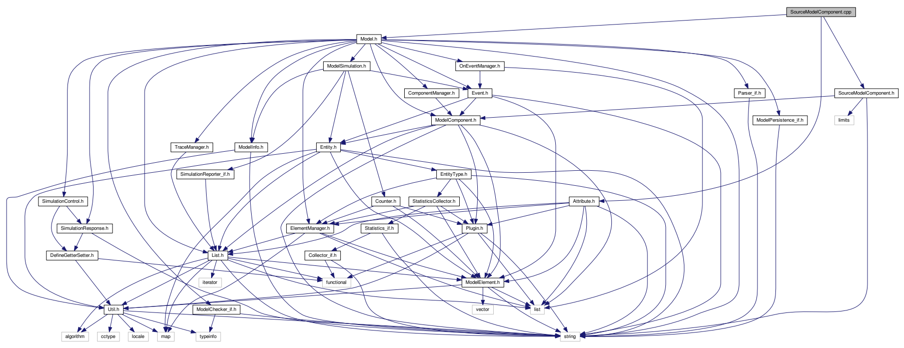
Classes

- class SinkModelComponent

6.146 SourceModelComponent.cpp File Reference

```
#include "SourceModelComponent.h"
#include "Model.h"
#include "Attribute.h"
```

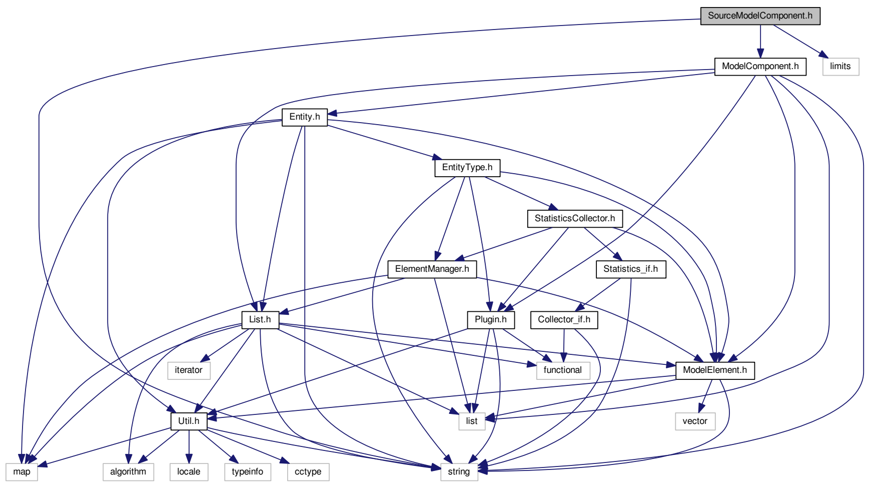
Include dependency graph for SourceModelComponent.cpp:



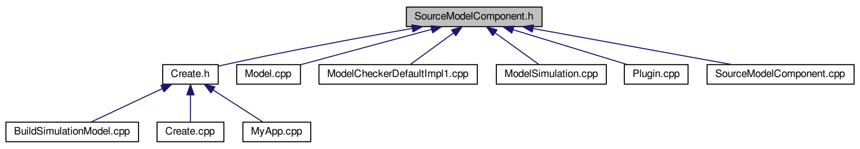
6.147 SourceModelComponent.h File Reference

```
#include "ModelComponent.h"
#include <string>
#include <limits>
```

Include dependency graph for SourceModelComponent.h:



This graph shows which files directly or indirectly include this file:



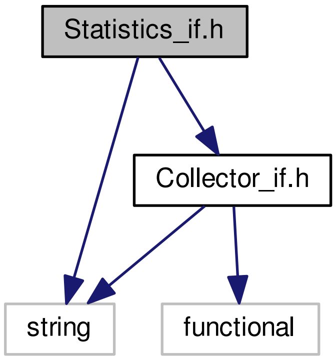
Classes

- class SourceModelComponent

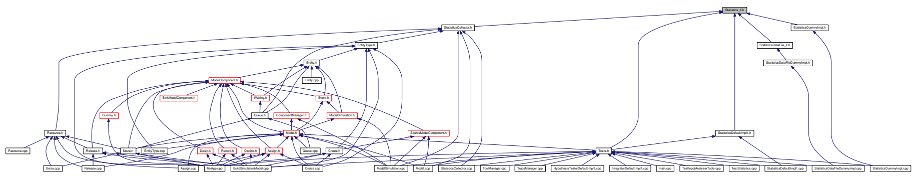
6.148 Statistics_if.h File Reference

```
#include <string>
#include "Collector_if.h"
```

Include dependency graph for Statistics_if.h:



This graph shows which files directly or indirectly include this file:



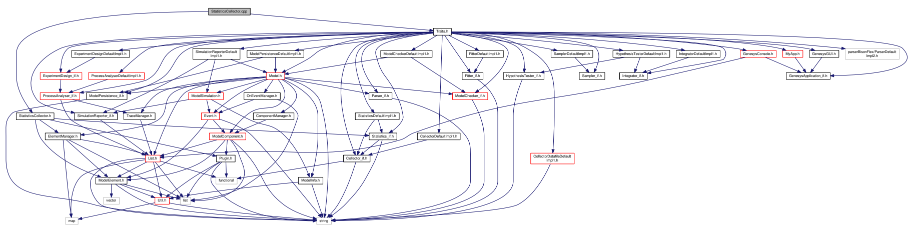
Classes

- class Statistics_if

6.149 StatisticsCollector.cpp File Reference

```
#include "StatisticsCollector.h"
#include "Traits.h"
```

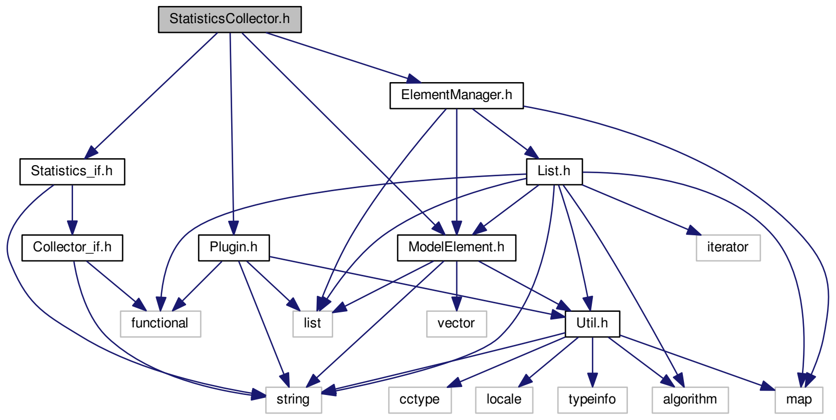
Include dependency graph for StatisticsCollector.cpp:



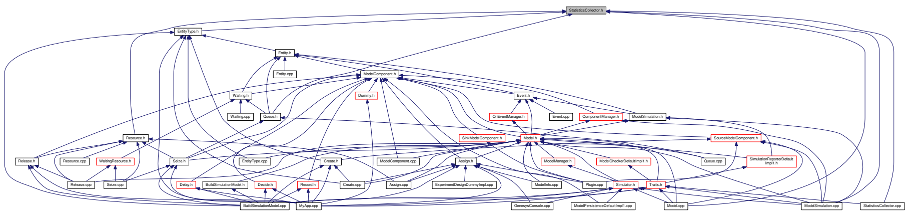
6.150 StatisticsCollector.h File Reference

```
#include "ModelElement.h"
#include "Statistics_if.h"
#include "ElementManager.h"
#include "Plugin.h"
```

Include dependency graph for StatisticsCollector.h:



This graph shows which files directly or indirectly include this file:



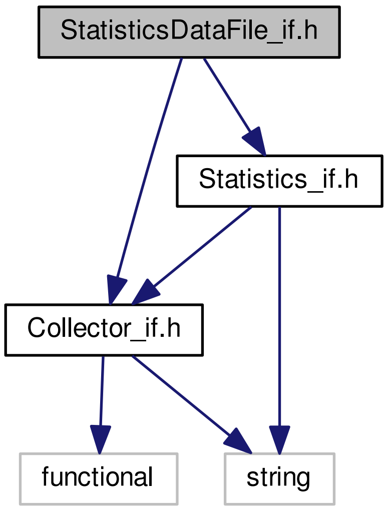
Classes

- class StatisticsCollector

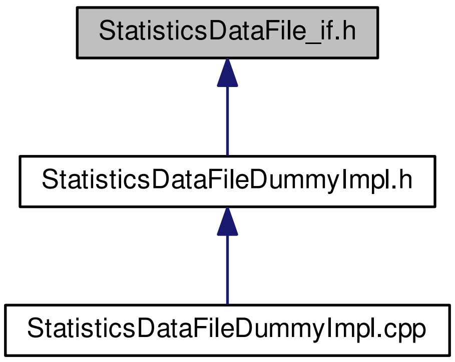
6.151 StatisticsDataFile_if.h File Reference

```
#include "Collector_if.h"
#include "Statistics_if.h"
```

Include dependency graph for StatisticsDataFile_if.h:



This graph shows which files directly or indirectly include this file:


StatisticsDataFile_if.h
StatisticsDataFileDummyImpl.h
StatisticsDataFileDummyImpl.cpp

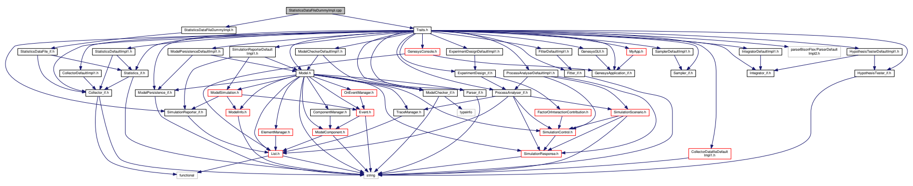
Classes

- class StatisticsDatafile_if

6.152 StatisticsDataFileDummyImpl.cpp File Reference

```
#include "StatisticsDataFileDummyImpl.h"
#include "Traits.h"
```

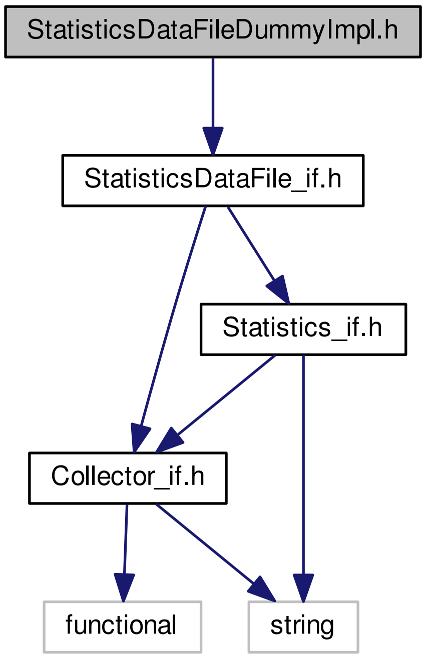
Include dependency graph for StatisticsDataFileDummyImpl.cpp:



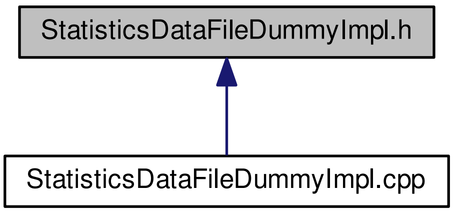
6.153 StatisticsDataFileDummyImpl.h File Reference

```
#include "StatisticsDataFile_if.h"
```

Include dependency graph for StatisticsDataFileDummyImpl.h:



This graph shows which files directly or indirectly include this file:


StatisticsDataFileDummyImpl.h
StatisticsDataFileDummyImpl.cpp

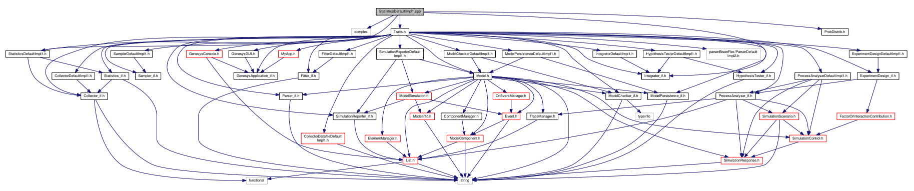
Classes

- class StatisticsDataFileDummyImpl

6.154 StatisticsDefaultImpl1.cpp File Reference

```
#include <complex>
#include "StatisticsDefaultImpl1.h"
#include "Traits.h"
#include "Integrator_if.h"
#include "ProbDistrib.h"
```

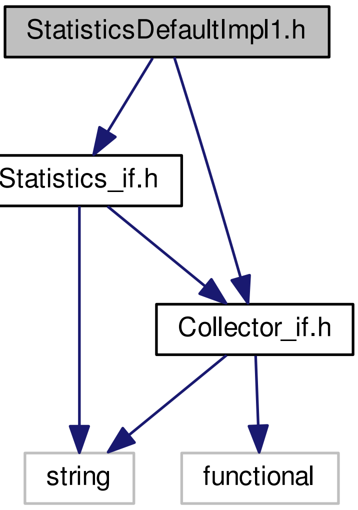
Include dependency graph for StatisticsDefaultImpl1.cpp:



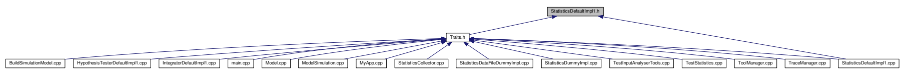
6.155 StatisticsDefaultImpl1.h File Reference

```
#include "Statistics_if.h"
#include "Collector_if.h"
```

Include dependency graph for StatisticsDefaultImpl1.h:



This graph shows which files directly or indirectly include this file:



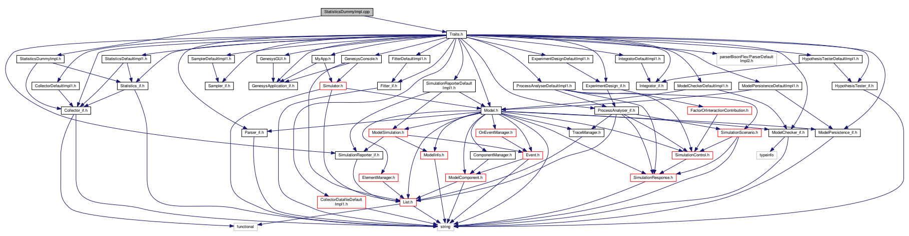
Classes

- class StatisticsDefaultImpl1

6.156 StatisticsDummyImpl.cpp File Reference

```
#include "StatisticsDummyImpl.h"
#include "Traits.h"
```

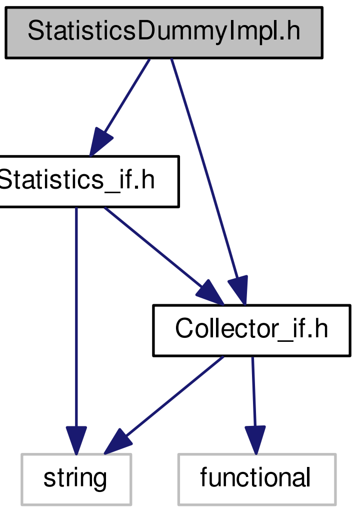
Include dependency graph for StatisticsDummyImpl.cpp:



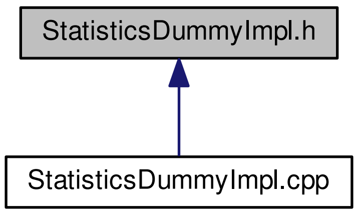
6.157 StatisticsDummyImpl.h File Reference

```
#include "Statistics_if.h"
#include "Collector_if.h"
```

Include dependency graph for StatisticsDummyImpl.h:



This graph shows which files directly or indirectly include this file:



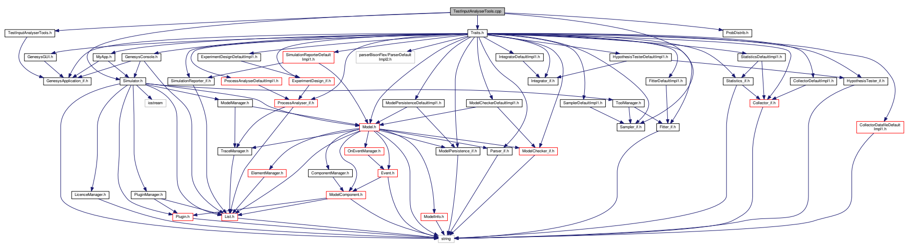
Classes

- class StatisticsDummyImpl

6.158 TestInputAnalyserTools.cpp File Reference

```
#include "TestInputAnalyserTools.h"
#include "Simulator.h"
#include "Sampler_if.h"
#include "ProbDistrib.h"
#include "Traits.h"
```

Include dependency graph for TestInputAnalyserTools.cpp:



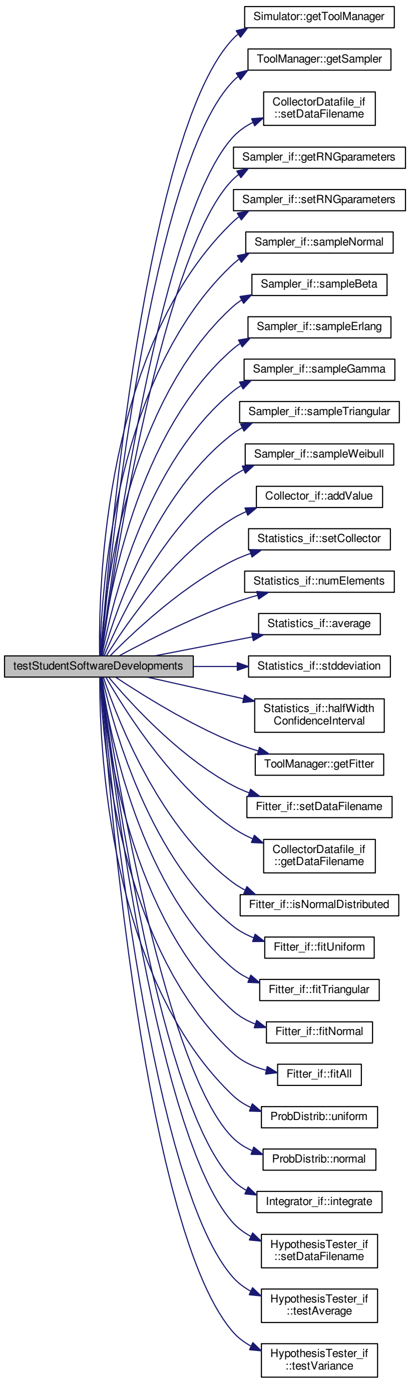
Functions

- void testStudentSoftwareDevelopments ()

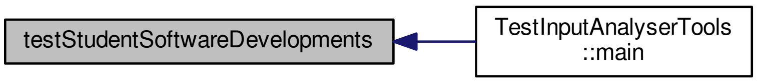
6.158.1 Function Documentation

6.158.1.1 void testStudentSoftwareDevelopments ( )

Here is the call graph for this function:


testStudentSoftwareDevelopments
Simulator::getToolManager
ToolManager::getSampler
CollectorDatafile_if ::setDataFilename
Sampler_if::getRNGparameters
Sampler_if::setRNGparameters
Sampler_if::sampleNormal
Sampler_if::sampleBeta
Sampler_if::sampleErlang
Sampler_if::sampleGamma
Sampler_if::sampleTriangular
Sampler_if::sampleWeibull
Collector_if::addValue
Statistics_if::setCollector
Statistics_if::numElements
Statistics_if::average
Statistics_if::stddeviation
Statistics_if::halfWidth ConfidenceInterval
ToolManager::getFitter
Fitter_if::setDataFilename
CollectorDatafile_if ::getDataFilename
Fitter_if::isNormalDistributed
Fitter_if::fitUniform
Fitter_if::fitTriangular
Fitter_if::fitNormal
Fitter_if::fitAll
ProbDistrib::uniform
ProbDistrib::normal
Integrator_if::integrate
HypothesisTester_if ::setDataFilename
HypothesisTester_if ::testAverage
HypothesisTester_if ::testVariance

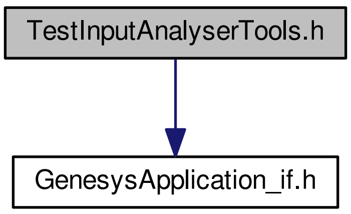
Here is the caller graph for this function:


testStudentSoftwareDevelopments
TestInputAnalyserTools ::main

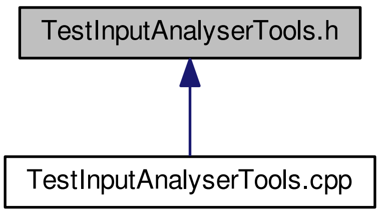
6.159 TestInputAnalyserTools.h File Reference

```
#include "GenesysApplication_if.h"
```

Include dependency graph for TestInputAnalyserTools.h:



This graph shows which files directly or indirectly include this file:



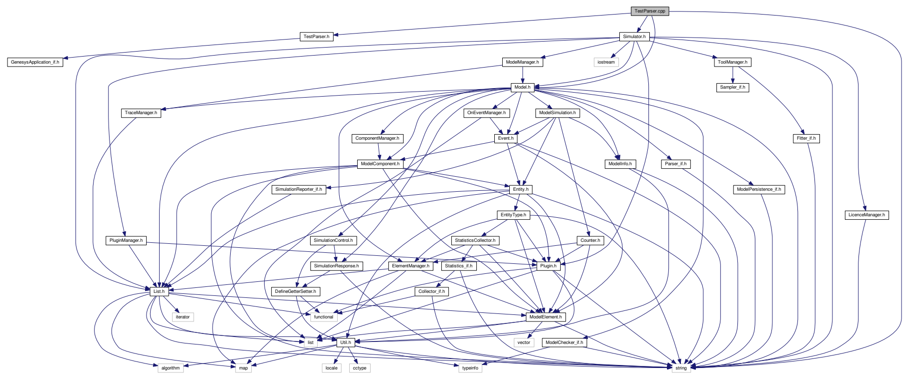
Classes

- class TestInputAnalyserTools

6.160 TestParser.cpp File Reference

```
#include <string>
#include "TestParser.h"
#include "Model.h"
#include "Simulator.h"
```

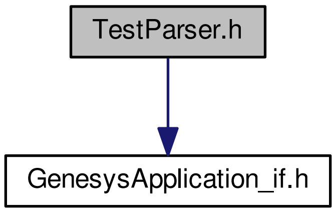
Include dependency graph for TestParser.cpp:



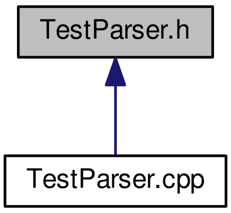
6.161 TestParser.h File Reference

```
#include "GenesysApplication_if.h"
```

Include dependency graph for TestParser.h:



This graph shows which files directly or indirectly include this file:



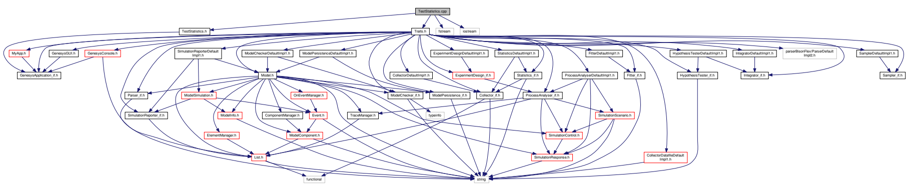
Classes

- class TestParser

6.162 TestStatistics.cpp File Reference

```
#include "TestStatistics.h"
#include <fstream>
#include <iostream>
#include "Traits.h"
```

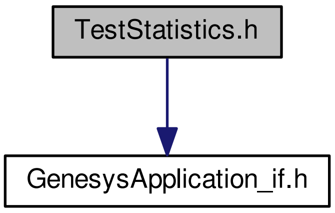
Include dependency graph for TestStatistics.cpp:



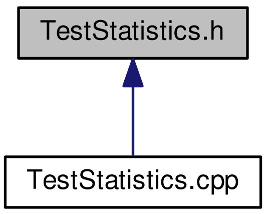
6.163 TestStatistics.h File Reference

```
#include "GenesysApplication_if.h"
```

Include dependency graph for TestStatistics.h:



This graph shows which files directly or indirectly include this file:



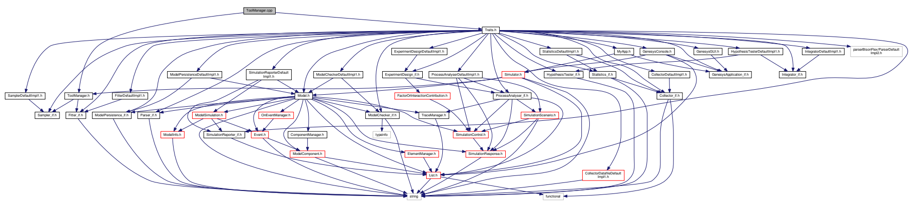
Classes

- class TestStatistics

6.164 ToolManager.cpp File Reference

```
#include "ToolManager.h"
#include "Traits.h"
```

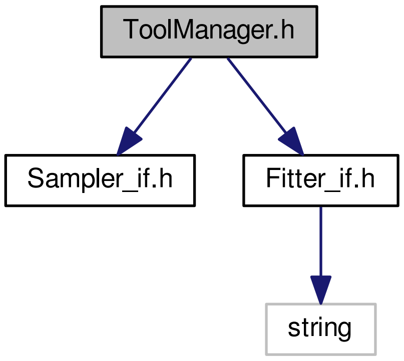
Include dependency graph for ToolManager.cpp:



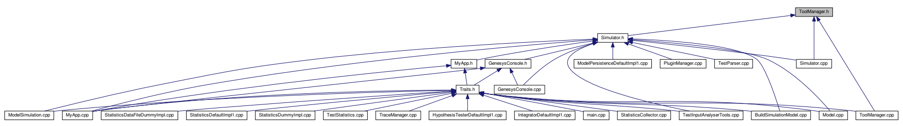
6.165 ToolManager.h File Reference

```
#include "Sampler_if.h"
#include "Fitter_if.h"
```

Include dependency graph for ToolManager.h:



This graph shows which files directly or indirectly include this file:



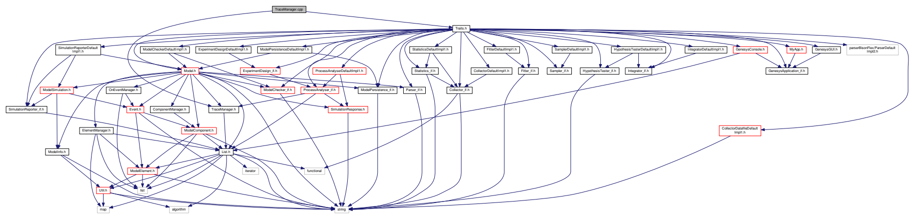
Classes

- class ToolManager

6.166 TraceManager.cpp File Reference

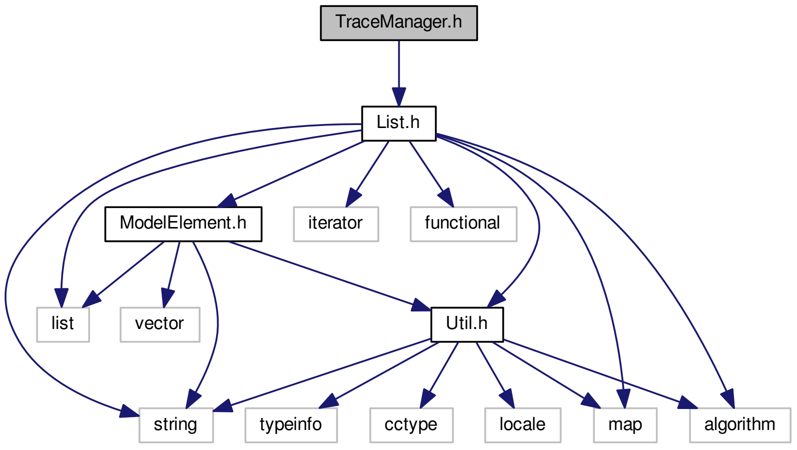
```
#include "TraceManager.h"
#include "Traits.h"
```

Include dependency graph for TraceManager.cpp:

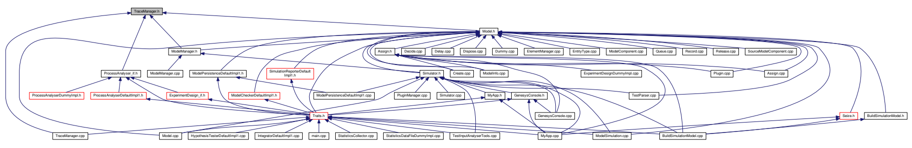


6.167 TraceManager.h File Reference

```
#include "List.h"
```
Include dependency graph for TraceManager.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TraceEvent
- class TraceErrorEvent
- class TraceSimulationEvent
- class TraceSimulationProcess
- class TraceManager

Typedefs

- typedef void(∗ traceListener) (TraceEvent)
- typedef void(∗ traceErrorListener) (TraceErrorEvent)
- typedef void(∗ traceSimulationListener) (TraceSimulationEvent)
- typedef void(∗ traceSimulationProcessListener) (TraceSimulationProcess)

6.167.1 Typedef Documentation

6.167.1.1 typedef void(∗ traceErrorListener) (**TraceErrorEvent**)

6.167.1.2 typedef void(∗ traceListener) (**TraceEvent**)

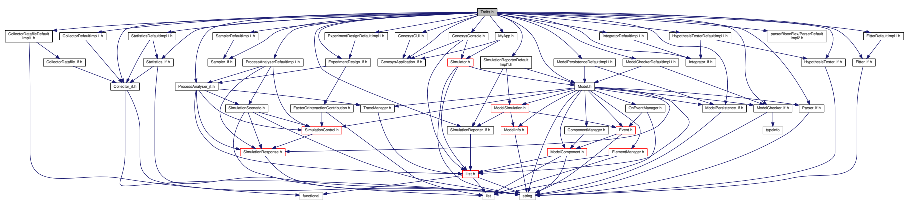
6.167.1.3 typedef void(∗ traceSimulationListener) (**TraceSimulationEvent**)

6.167.1.4 typedef void(∗ traceSimulationProcessListener) (**TraceSimulationProcess**)

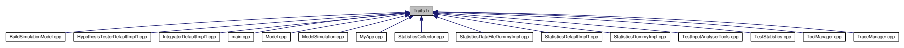
6.168 Traits.h File Reference

```
#include "Model.h"
#include "Collector_if.h"
#include "Sampler_if.h"
#include "Fitter_if.h"
#include "ModelChecker_if.h"
#include "Parser_if.h"
#include "Statistics_if.h"
#include "Integrator_if.h"
#include "HypothesisTester_if.h"
#include "ModelPersistence_if.h"
#include "GenesysApplication_if.h"
#include "ProcessAnalyser_if.h"
#include "ExperimentDesign_if.h"
#include "SimulationReporter_if.h"
#include "MyApp.h"
#include "GenesysGUI.h"
#include "GenesysConsole.h"
#include "CollectorDefaultImpl1.h"
#include "CollectorDatafileDefaultImpl1.h"
#include "StatisticsDefaultImpl1.h"
#include "IntegratorDefaultImpl1.h"
#include "HypothesisTesterDefaultImpl1.h"
#include "SamplerDefaultImpl1.h"
#include "SimulationReporterDefaultImpl1.h"
#include "ModelCheckerDefaultImpl1.h"
#include "ModelPersistenceDefaultImpl1.h"
#include "parserBisonFlex/ParserDefaultImpl2.h"
#include "ExperimentDesignDefaultImpl1.h"
#include "ProcessAnalyserDefaultImpl1.h"
#include "FitterDefaultImpl1.h"
```

Include dependency graph for Traits.h:



This graph shows which files directly or indirectly include this file:



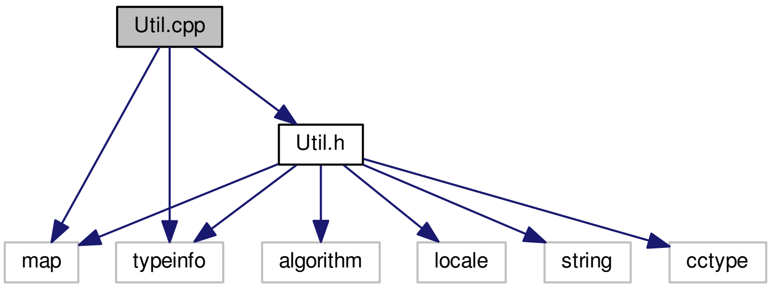
Classes

- struct Traits< T >
- struct Traits< GenesysApplication_if >
- struct Traits< Model >
- struct Traits< ModelPersistence_if >
- struct Traits< SimulationReporter_if >
- struct Traits< ModelComponent >
- struct Traits< ModelChecker_if >
- struct Traits< Parser_if >
- struct Traits< Collector_if >
- struct Traits< Statistics_if >
- struct Traits< Integrator_if >
- struct Traits< Sampler_if >
- struct Traits< Fitter_if >
- struct Traits< HypothesisTester_if >
- struct Traits< ExperimentDesign_if >
- struct Traits< ProcessAnalyser_if >

6.169 Util.cpp File Reference

```
#include "Util.h"
#include <typeinfo>
#include <map>
```

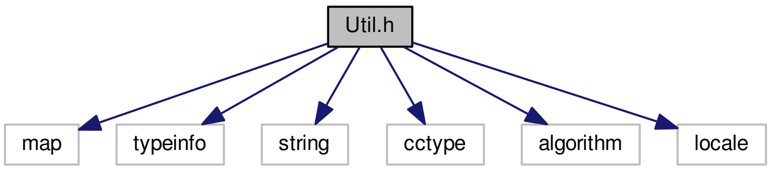
Include dependency graph for Util.cpp:



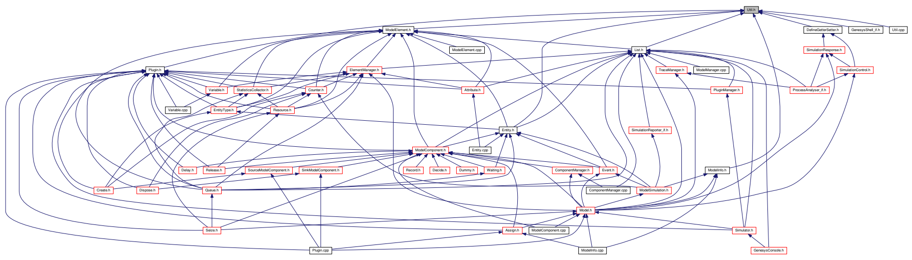
6.170 Util.h File Reference

```
#include <map>
#include <typeinfo>
#include <string>
#include <cctype>
#include <algorithm>
#include <locale>
```

Include dependency graph for Util.h:



This graph shows which files directly or indirectly include this file:



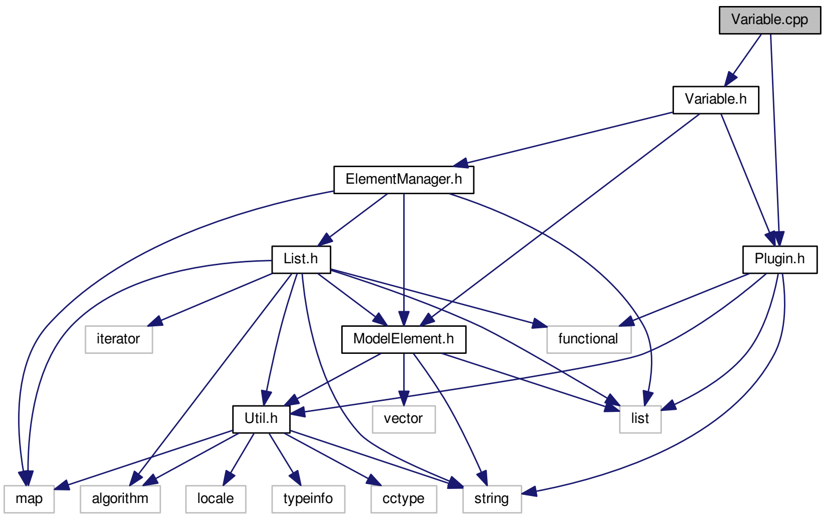
Classes

- class Util

6.171 Variable.cpp File Reference

```
#include "Variable.h"
#include "Plugin.h"
```

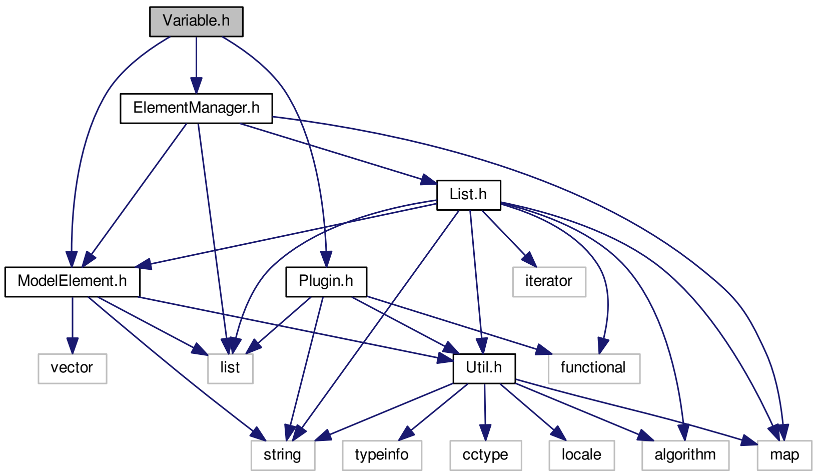
Include dependency graph for Variable.cpp:



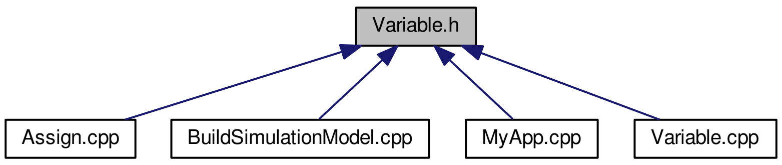
6.172 Variable.h File Reference

```
#include "ModelElement.h"
#include "ElementManager.h"
#include "Plugin.h"
```

Include dependency graph for Variable.h:



This graph shows which files directly or indirectly include this file:



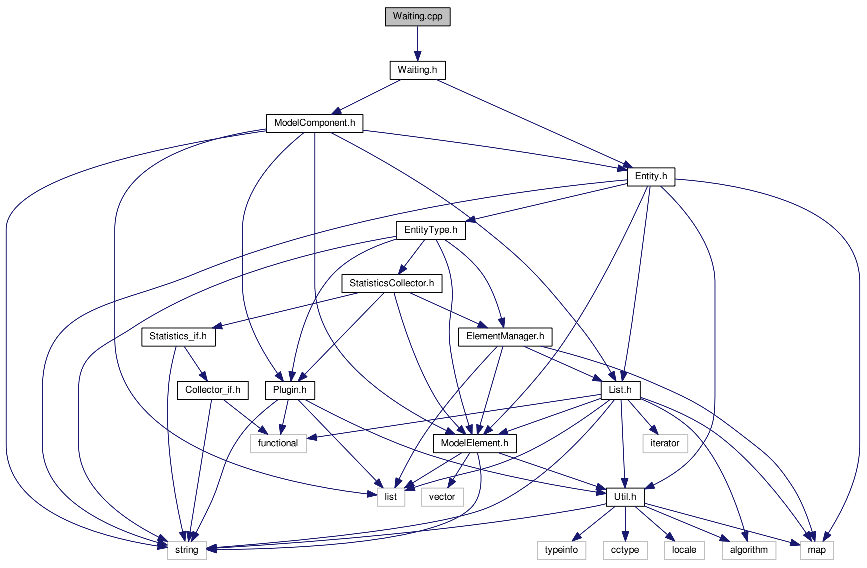
Classes

- class Variable

6.173 Waiting.cpp File Reference

```
#include "Waiting.h"
```

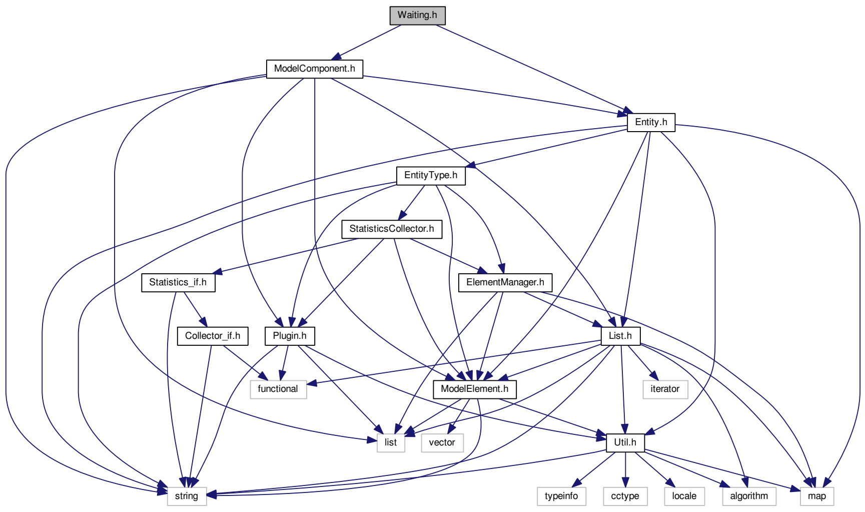
Include dependency graph for Waiting.cpp:



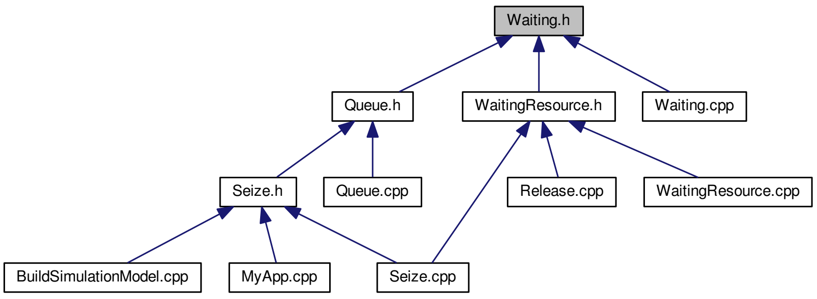
6.174 Waiting.h File Reference

```
#include "Entity.h"
#include "ModelComponent.h"
```

Include dependency graph for Waiting.h:



This graph shows which files directly or indirectly include this file:



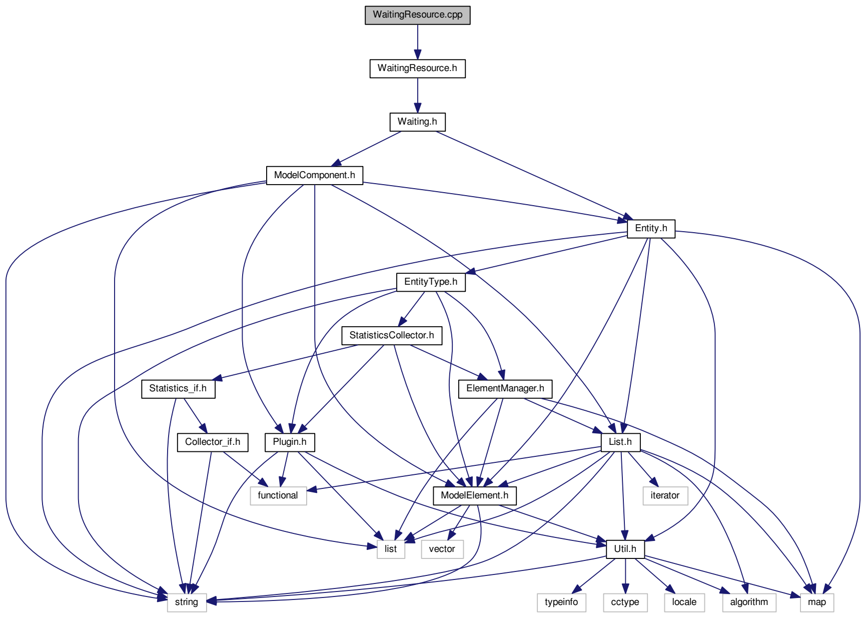
Classes

- class Waiting

6.175 WaitingResource.cpp File Reference

```
#include "WaitingResource.h"
```

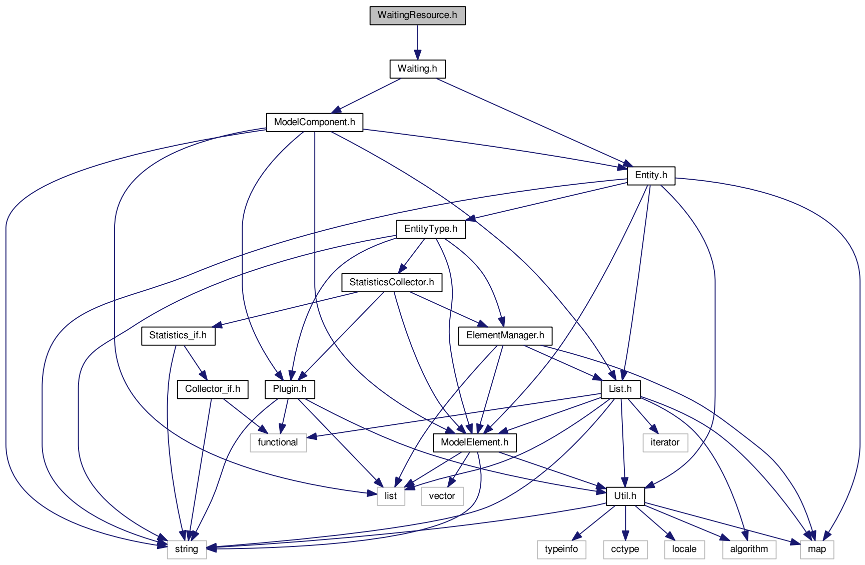
Include dependency graph for WaitingResource.cpp:



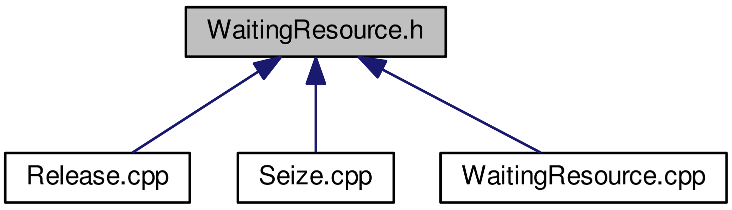
6.176 WaitingResource.h File Reference

```
#include "Waiting.h"
```

Include dependency graph for WaitingResource.h:



This graph shows which files directly or indirectly include this file:



Classes

- class WaitingResource