

GPUSAT

Generated by Doxygen 1.8.18



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 gpusat Namespace Reference	7
4.1.1 Enumeration Type Documentation	8
4.1.1.1 dataStructure	8
4.1.1.2 nodeType	8
4.1.2 Function Documentation	8
4.1.2.1 compTreedType()	9
4.1.2.2 compVars()	9
4.1.2.3 getCount()	10
4.1.2.4 getTime()	11
4.1.2.5 operator<<() [1/3]	11
4.1.2.6 operator<<() [2/3]	12
4.1.2.7 operator<<() [3/3]	12
4.1.2.8 printbagType()	12
4.1.2.9 printtreedecType()	13
4.1.2.10 printtreeType()	13
4.1.2.11 solJson()	14
4.1.2.12 solutiontable()	14
<b>5 Class Documentation</b>	<b>17</b>
5.1 gpusat::BagMatrix Class Reference	17
5.1.1 Detailed Description	17
5.1.2 Constructor & Destructor Documentation	17
5.1.2.1 BagMatrix() [1/2]	17
5.1.2.2 BagMatrix() [2/2]	18
5.1.3 Member Function Documentation	18
5.1.3.1 columns()	18
5.1.3.2 operator()() [1/2]	18
5.1.3.3 operator()() [2/2]	18
5.1.3.4 rows()	19
5.2 gpusat::bagType Struct Reference	19
5.2.1 Detailed Description	20
5.2.2 Member Data Documentation	20
5.2.2.1 bags	20
5.2.2.2 correction	20

5.2.2.3 edges	20
5.2.2.4 exponent	21
5.2.2.5 id	21
5.2.2.6 maxSize	21
5.2.2.7 solution	21
5.2.2.8 variables	21
5.3 gpusat::CNFParser Class Reference	22
5.3.1 Constructor & Destructor Documentation	22
5.3.1.1 CNFParser()	22
5.3.2 Member Function Documentation	22
5.3.2.1 parseSatFormula()	22
5.4 gpusat::CutSetWidthFitnessFunction Class Reference	23
5.4.1 Detailed Description	24
5.4.2 Constructor & Destructor Documentation	24
5.4.2.1 CutSetWidthFitnessFunction()	24
5.4.2.2 ~CutSetWidthFitnessFunction()	25
5.4.3 Member Function Documentation	25
5.4.3.1 clone()	25
5.4.3.2 fitness()	25
5.4.3.3 getMaxCutSetSize()	26
5.5 gpusat::Decomposer Class Reference	26
5.5.1 Member Function Documentation	26
5.5.1.1 computeDecomposition()	26
5.6 gpusat::Graphoutput Class Reference	27
5.6.1 Constructor & Destructor Documentation	28
5.6.1.1 Graphoutput()	28
5.6.2 Member Function Documentation	28
5.6.2.1 getFilename()	28
5.6.2.2 graphEdgeSet()	28
5.6.2.3 graphEnd()	29
5.6.2.4 graphStart()	30
5.6.2.5 isEnabled()	31
5.6.2.6 neo4jSat()	31
5.6.2.7 neo4jTD()	32
5.6.2.8 nodeBag()	33
5.6.2.9 nodeJoin()	34
5.6.3 Member Data Documentation	35
5.6.3.1 baseIdJoin	35
5.6.3.2 baseIdSol	35
5.6.3.3 connectbag	35
5.6.3.4 dualEdge	35
5.6.3.5 inbag	36

5.6.3.6 incidenceedge	36
5.6.3.7 primaledge	36
5.7 gpusat::JoinSizeFitnessFunction Class Reference	36
5.7.1 Detailed Description	37
5.7.2 Constructor & Destructor Documentation	37
5.7.2.1 JoinSizeFitnessFunction()	37
5.7.2.2 ~JoinSizeFitnessFunction()	38
5.7.3 Member Function Documentation	38
5.7.3.1 clone()	38
5.7.3.2 fitness()	38
5.8 gpusat::NumJoinFitnessFunction Class Reference	39
5.8.1 Detailed Description	39
5.8.2 Constructor & Destructor Documentation	40
5.8.2.1 NumJoinFitnessFunction()	40
5.8.2.2 ~NumJoinFitnessFunction()	40
5.8.3 Member Function Documentation	40
5.8.3.1 clone()	40
5.8.3.2 fitness()	41
5.9 gpusat::Preprocessor Class Reference	41
5.9.1 Member Function Documentation	41
5.9.1.1 preprocessDecomp()	41
5.9.1.2 preprocessFacts()	42
5.10 gpusat::satformulaType Struct Reference	43
5.10.1 Detailed Description	44
5.10.2 Member Data Documentation	44
5.10.2.1 clauses	44
5.10.2.2 facts	44
5.10.2.3 numVars	44
5.10.2.4 numWeights	45
5.10.2.5 unsat	45
5.10.2.6 variableWeights	45
5.11 gpusat::Solver Class Reference	45
5.11.1 Detailed Description	46
5.11.2 Constructor & Destructor Documentation	46
5.11.2.1 Solver()	47
5.11.3 Member Function Documentation	47
5.11.3.1 cleanTree()	47
5.11.3.2 combineTree()	48
5.11.3.3 solveIntroduceForget()	48
5.11.3.4 solveJoin()	51
5.11.3.5 solveProblem()	52
5.11.4 Member Data Documentation	53

---

5.11.4.1 context . . . . .	54
5.11.4.2 graphoutput . . . . .	54
5.11.4.3 isSat . . . . .	54
5.11.4.4 maxBag . . . . .	54
5.11.4.5 maxMemoryBuffer . . . . .	54
5.11.4.6 maxTableSize . . . . .	54
5.11.4.7 memorySize . . . . .	55
5.11.4.8 numIntroduceForget . . . . .	55
5.11.4.9 numJoin . . . . .	55
5.11.4.10 program . . . . .	55
5.11.4.11 queue . . . . .	55
5.11.4.12 solutionType . . . . .	55
5.11.4.13 verbose . . . . .	56
5.11.4.14 visualization . . . . .	56
5.12 gpusat::TableLines Struct Reference . . . . .	56
5.12.1 Detailed Description . . . . .	56
5.12.2 Member Data Documentation . . . . .	57
5.12.2.1 headline . . . . .	57
5.12.2.2 solutions . . . . .	57
5.12.2.3 totalSol . . . . .	57
5.13 gpusat::TDParse Class Reference . . . . .	57
5.13.1 Constructor & Destructor Documentation . . . . .	58
5.13.1.1 TDParse() . . . . .	58
5.13.2 Member Function Documentation . . . . .	58
5.13.2.1 parseTreeDecomp() . . . . .	58
5.13.3 Member Data Documentation . . . . .	59
5.13.3.1 defaultWeight . . . . .	59
5.14 gpusat::treedecType Struct Reference . . . . .	59
5.14.1 Detailed Description . . . . .	60
5.14.2 Member Data Documentation . . . . .	60
5.14.2.1 bags . . . . .	60
5.14.2.2 numb . . . . .	60
5.14.2.3 numVars . . . . .	61
5.14.2.4 width . . . . .	61
5.15 gpusat::treeType Struct Reference . . . . .	61
5.15.1 Detailed Description . . . . .	62
5.15.2 Member Data Documentation . . . . .	62
5.15.2.1 elements . . . . .	62
5.15.2.2 maxId . . . . .	62
5.15.2.3 minId . . . . .	62
5.15.2.4 numSolutions . . . . .	62
5.15.2.5 size . . . . .	63

---

5.16 gpusat::Visualization Class Reference	63
5.16.1 Constructor & Destructor Documentation	63
5.16.1.1 Visualization()	63
5.16.2 Member Function Documentation	64
5.16.2.1 getClausesJson()	64
5.16.2.2 getFilename()	64
5.16.2.3 getTdTimeline()	64
5.16.2.4 getTreeDecJson()	64
5.16.2.5 getWriterBuilder()	64
5.16.2.6 isEnabled()	65
5.16.2.7 tdTimelineAppend() [1/2]	65
5.16.2.8 tdTimelineAppend() [2/2]	65
5.16.2.9 testJson()	66
5.16.2.10 visuClauses()	67
5.16.2.11 visuTreeDec()	67
5.16.2.12 writeJsonFile()	68
5.16.2.13 writeJsonToStdout() [1/2]	68
5.16.2.14 writeJsonToStdout() [2/2]	69
5.17 gpusat::WidthCutSetFitnessFunction Class Reference	70
5.17.1 Detailed Description	70
5.17.2 Constructor & Destructor Documentation	71
5.17.2.1 WidthCutSetFitnessFunction()	71
5.17.2.2 ~WidthCutSetFitnessFunction()	71
5.17.3 Member Function Documentation	71
5.17.3.1 clone()	71
5.17.3.2 fitness()	72
5.17.3.3 getMaxCutSetSize()	72
<b>6 File Documentation</b>	<b>73</b>
6.1 include/decomposer.h File Reference	73
6.2 include/FitnessFunctions/CutSetWidthFitnessFunction.h File Reference	74
6.3 include/FitnessFunctions/JoinSizeFitnessFunction.h File Reference	75
6.4 include/FitnessFunctions/NumJoinFitnessFunction.h File Reference	76
6.5 include/FitnessFunctions/WidthCutSetFitnessFunction.h File Reference	77
6.6 include/gpusatparser.h File Reference	78
6.7 include/gpusatpreprocessor.h File Reference	79
6.8 include/gpusatutils.h File Reference	80
6.9 include/graphoutput.h File Reference	81
6.10 include/kernel.h File Reference	82
6.11 include/solver.h File Reference	82
6.12 include/types.h File Reference	83
6.13 include/visualization.h File Reference	84

6.14 src/decomposer.cpp File Reference . . . . .	86
6.15 src/FitnessFunctions/CutSetWidthFitnessFunction.cpp File Reference . . . . .	86
6.16 src/FitnessFunctions/JoinSizeFitnessFunction.cpp File Reference . . . . .	87
6.17 src/FitnessFunctions/NumJoinFitnessFunction.cpp File Reference . . . . .	87
6.18 src/FitnessFunctions/WidthCutSetFitnessFunction.cpp File Reference . . . . .	88
6.19 src/gpusatparser.cpp File Reference . . . . .	88
6.20 src/gpusatpreprocessor.cpp File Reference . . . . .	89
6.21 src/graphoutput.cpp File Reference . . . . .	89
6.22 src/main.cpp File Reference . . . . .	90
6.22.1 Macro Definition Documentation . . . . .	91
6.22.1.1 __CL_ENABLE_EXCEPTIONS . . . . .	91
6.22.1.2 NELEMS . . . . .	91
6.22.2 Function Documentation . . . . .	91
6.22.2.1 buildKernel() . . . . .	92
6.22.2.2 device_query() . . . . .	92
6.22.2.3 main() . . . . .	93
6.22.2.4 PrintDeviceInfo() . . . . .	94
6.22.3 Variable Documentation . . . . .	95
6.22.3.1 attributeNames . . . . .	95
6.22.3.2 attributeTypes . . . . .	95
6.22.3.3 kernelStr . . . . .	95
6.23 src/solver.cpp File Reference . . . . .	96
6.24 src/visualization.cpp File Reference . . . . .	96
6.24.1 Macro Definition Documentation . . . . .	97
6.24.1.1 LOGGER . . . . .	97
6.24.1.2 LOGGER2 . . . . .	97
<b>Index</b>	<b>99</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

gpusat::BagMatrix . . . . .	17
gpusat::bagType . . . . .	19
gpusat::CNFParser . . . . .	22
gpusat::Decomposer . . . . .	26
gpusat::Graphoutput . . . . .	27
ITreeDecompositionFitnessFunction	
gpusat::CutSetWidthFitnessFunction . . . . .	23
gpusat::JoinSizeFitnessFunction . . . . .	36
gpusat::NumJoinFitnessFunction . . . . .	39
gpusat::WidthCutSetFitnessFunction . . . . .	70
gpusat::Preprocessor . . . . .	41
gpusat::satformulaType . . . . .	43
gpusat::Solver . . . . .	45
gpusat::TableLines . . . . .	56
gpusat::TDParse . . . . .	57
gpusat::treedecType . . . . .	59
gpusat::treeType . . . . .	61
gpusat::Visualization . . . . .	63



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">gpusat::BagMatrix</a>	17
<a href="#">gpusat::bagType</a>	
Type for a bag in the tree decomposition	19
<a href="#">gpusat::CNFParser</a>	22
<a href="#">gpusat::CutSetWidthFitnessFunction</a>	23
<a href="#">gpusat::Decomposer</a>	26
<a href="#">gpusat::Graphoutput</a>	27
<a href="#">gpusat::JoinSizeFitnessFunction</a>	36
<a href="#">gpusat::NumJoinFitnessFunction</a>	39
<a href="#">gpusat::Preprocessor</a>	41
<a href="#">gpusat::satformulaType</a>	
Type for saving the sat formula	43
<a href="#">gpusat::Solver</a>	45
<a href="#">gpusat::TableLines</a>	56
<a href="#">gpusat::TDParser</a>	57
<a href="#">gpusat::treedecType</a>	
Type for saving a tree decomposition	59
<a href="#">gpusat::treeType</a>	
Tree type for storing the models	61
<a href="#">gpusat::Visualization</a>	63
<a href="#">gpusat::WidthCutSetFitnessFunction</a>	70



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

include/decomposer.h	73
include/gpusatparser.h	78
include/gpusatpreprocessor.h	79
include/gpusatutils.h	80
include/graphoutput.h	81
include/kernel.h	82
include/solver.h	82
include/types.h	83
include/visualization.h	84
include/FitnessFunctions/CutSetWidthFitnessFunction.h	74
include/FitnessFunctions/JoinSizeFitnessFunction.h	75
include/FitnessFunctions/NumJoinFitnessFunction.h	76
include/FitnessFunctions/WidthCutSetFitnessFunction.h	77
src/decomposer.cpp	86
src/gpusatparser.cpp	88
src/gpusatpreprocessor.cpp	89
src/graphoutput.cpp	89
src/main.cpp	90
src/solver.cpp	96
src/visualization.cpp	96
src/FitnessFunctions/CutSetWidthFitnessFunction.cpp	86
src/FitnessFunctions/JoinSizeFitnessFunction.cpp	87
src/FitnessFunctions/NumJoinFitnessFunction.cpp	87
src/FitnessFunctions/WidthCutSetFitnessFunction.cpp	88



## Chapter 4

# Namespace Documentation

### 4.1 gpusat Namespace Reference

#### Classes

- class [BagMatrix](#)
- struct [bagType](#)  
*type for a bag in the tree decomposition*
- class [CNFParser](#)
- class [CutSetWidthFitnessFunction](#)
- class [Decomposer](#)
- class [Graphoutput](#)
- class [JoinSizeFitnessFunction](#)
- class [NumJoinFitnessFunction](#)
- class [Preprocessor](#)
- struct [satformulaType](#)  
*type for saving the sat formula*
- class [Solver](#)
- struct [TableLines](#)
- class [TDParser](#)
- struct [treedecType](#)  
*type for saving a tree decomposition*
- struct [treeType](#)  
*tree type for storing the models*
- class [Visualization](#)
- class [WidthCutSetFitnessFunction](#)

#### Enumerations

- enum [dataStructure](#) { [dataStructure::ARRAY](#), [dataStructure::TREE](#) }
- enum [nodeType](#) { [nodeType::JOIN](#), [nodeType::INTRODUCEFORGET](#) }

## Functions

- bool `compTreedType` (const `bagType` \*a, const `bagType` \*b)
- bool `compVars` (const `cl_long` &a, const `cl_long` &b)
- `cl_double` `getCount` (`cl_long` id, `cl_long` \*tree, `cl_long` numVars)
- long long int `getTime` ()
- `std::ostream` & `operator<<` (`std::ostream` &os, const `dataStructure` ds)
- `std::ostream` & `operator<<` (`std::ostream` &os, const `std::vector`< `bagType` \* > vec)
- `std::ostream` & `operator<<` (`std::ostream` &os, const `std::vector`< `cl_long` > vec)
- void `printbagType` (`bagType` \*bag, `std::ostream` &stream, int depth=0)  
*print information for a bag in the tree decomposition*
- void `printtreedecType` (`treedecType` \*dec, `std::ostream` &stream)  
*print a tree decomposition*
- void `printtreeType` (`treeType` \*tree, `std::ostream` &stream, `size_t` size, int depth=0)
- `TableLines` `solJson` (`bagType` node, `dataStructure` solutionType)
- `std::string` `solutiontable` (`bagType` node, `dataStructure` solutionType)  
*Generate a formatted stringoutput for a solved node with solutions.*

## 4.1.1 Enumeration Type Documentation

### 4.1.1.1 dataStructure

```
enum gpusat::dataStructure [strong]
```

#### Enumerator

ARRAY	
TREE	

### 4.1.1.2 nodeType

```
enum gpusat::nodeType [strong]
```

#### Enumerator

JOIN	
INTRODUCEFORGET	

## 4.1.2 Function Documentation



#### 4.1.2.1 compTreedType()

```
bool gpusat::compTreedType (
    const bagType * a,
    const bagType * b ) [inline]
```

Function that compares two tree decompositions by id.

##### Parameters

<i>a</i>	the first tree decomposition
<i>b</i>	the second tree decomposition

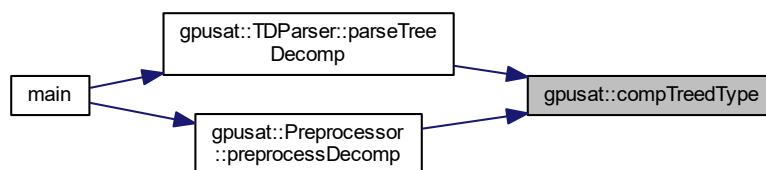
##### Returns

$a < b$

References gpusat::bagType::id.

Referenced by gpusat::TDParser::parseTreeDecomp(), and gpusat::Preprocessor::preprocessDecomp().

Here is the caller graph for this function:



#### 4.1.2.2 compVars()

```
bool gpusat::compVars (
    const cl_long & a,
    const cl_long & b ) [inline]
```

Function that compares two variables.

##### Parameters

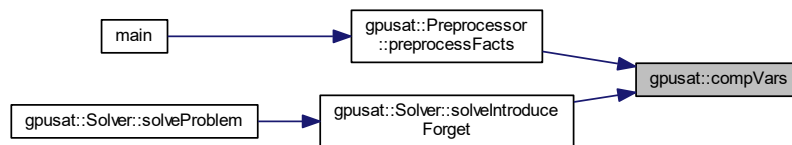
<i>a</i>	the first variable
<i>b</i>	the second variable

**Returns**

true if  $\text{abs } a < b$

Referenced by `gpusat::Preprocessor::preprocessFacts()`, and `gpusat::Solver::solveIntroduceForget()`.

Here is the caller graph for this function:

**4.1.2.3 getCount()**

```

cl_double gpusat::getCount (
    cl_long id,
    cl_long * tree,
    cl_long numVars ) [inline]

```

returns the model count which corresponds to the given id

**Parameters**

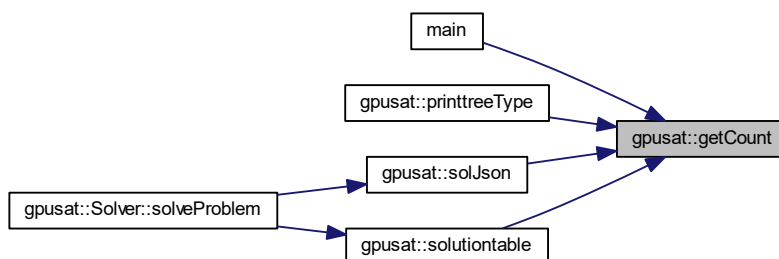
<i>id</i>	the id for which the model count should be returned
<i>tree</i>	a pointer to the tree structure
<i>numVars</i>	the number of variables in the bag

**Returns**

the model count

Referenced by `main()`, `printtreeType()`, `solJson()`, and `solutiontable()`.

Here is the caller graph for this function:



#### 4.1.2.4 getTime()

```
long long int gpusat::getTime ( ) [inline]
```

##### Returns

the time in milliseconds since the epoch

Referenced by `main()`.

Here is the caller graph for this function:



#### 4.1.2.5 operator<<() [1/3]

```
std::ostream& gpusat::operator<< (
    std::ostream & os,
    const dataStructure ds ) [inline]
```

References `ARRAY`, and `TREE`.

#### 4.1.2.6 operator<<() [2/3]

```
std::ostream& gpusat::operator<< (
    std::ostream & os,
    const std::vector< bagType * > vec ) [inline]
```

#### 4.1.2.7 operator<<() [3/3]

```
std::ostream& gpusat::operator<< (
    std::ostream & os,
    const std::vector< cl_long > vec ) [inline]
```

#### 4.1.2.8 printbagType()

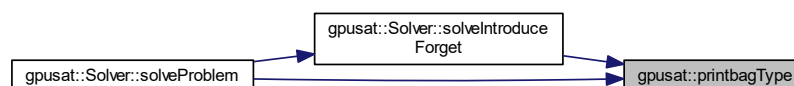
```
void gpusat::printbagType (
    bagType * bag,
    std::ostream & stream,
    int depth = 0 ) [inline]
```

print information for a bag in the tree decomposition

References gpusat::bagType::bags, gpusat::bagType::correction, gpusat::bagType::edges, gpusat::bagType::exponent, gpusat::bagType::id, and gpusat::bagType::variables.

Referenced by gpusat::Solver::solveIntroduceForget(), and gpusat::Solver::solveProblem().

Here is the caller graph for this function:



#### 4.1.2.9 printtreedecType()

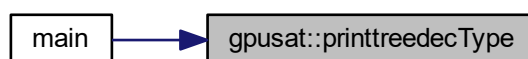
```
void gpusat::printtreedecType (
    treedecType * dec,
    std::ostream & stream ) [inline]
```

print a tree decomposition

References gpusat::treedecType::bags, and gpusat::treedecType::numb.

Referenced by main().

Here is the caller graph for this function:



#### 4.1.2.10 printtreeType()

```
void gpusat::printtreeType (
    treeType * tree,
    std::ostream & stream,
    size_t size,
    int depth = 0 ) [inline]
```

References gpusat::treeType::elements, getCount(), gpusat::treeType::maxId, gpusat::treeType::minId, gpusat::treeType::numSolutions, and gpusat::treeType::size.

Here is the call graph for this function:



#### 4.1.2.11 solJson()

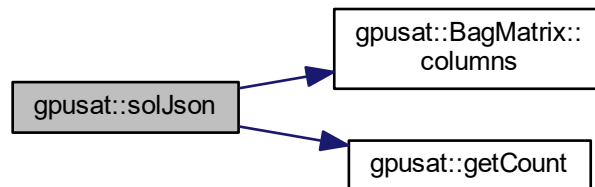
```
TableLines gpusat::solJson (
    bagType node,
    dataStructure solutionType )
```

Construct Tablelines from a solved node of the tree decomposition If no solution is stored in this node return empty grid and totalSol -1.

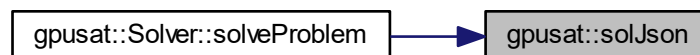
References gpusat::BagMatrix::columns(), gpusat::bagType::correction, gpusat::treeType::elements, getCount(), gpusat::TableLines::headline, gpusat::treeType::minId, gpusat::bagType::solution, gpusat::TableLines::solutions, gpusat::TableLines::totalSol, TREE, and gpusat::bagType::variables.

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.1.2.12 solutiontable()

```
std::string gpusat::solutiontable (
    bagType node,
    dataStructure solutionType ) [inline]
```

Generate a formatted stringoutput for a solved node with solutions.

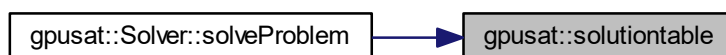
References `gpusat::bagType::correction`, `gpusat::treeType::elements`, `getCount()`, `gpusat::treeType::minId`, `gpusat::bagType::solution`, `TREE`, and `gpusat::bagType::variables`.

Referenced by `gpusat::Solver::solveProblem()`.

Here is the call graph for this function:



Here is the caller graph for this function:







## Chapter 5

# Class Documentation

### 5.1 gpusat::BagMatrix Class Reference

```
#include <visualization.h>
```

#### Public Member Functions

- [BagMatrix](#) ()
- [BagMatrix](#) (size\_t rows, size\_t cols)
- size\_t [columns](#) ()
- cl\_long & [operator\(\)](#) (size\_t i, size\_t j)
- cl\_long [operator\(\)](#) (size\_t i, size\_t j) const
- size\_t [rows](#) ()

#### 5.1.1 Detailed Description

Class for storing 2dim 'ulong' values using vector and a 1dim memory layout. See <https://stackoverflow.com/a/2076668> Example usage:

```
BagMatrix mygrid (5, 5); for (int r = 0; r < mygrid.rows(); r++){ for (int c = 0; c < mygrid.columns(); c++){ mygrid[r][c] = r * c; } } Print the array for (int r = 0; r < mygrid.rows(); r++){ std::cout << std::hex << &(mygrid[r][0]) << std::dec; std::cout << ": "; for (int c = 0; c < mygrid.columns(); c++){ std::cout << mygrid[r][c] << ' '; } std::cout << std::endl; }
```

#### 5.1.2 Constructor & Destructor Documentation

##### 5.1.2.1 BagMatrix() [1/2]

```
gpusat::BagMatrix::BagMatrix (
    size_t rows,
    size_t cols )
```

### 5.1.2.2 BagMatrix() [2/2]

```
gpusat::BagMatrix::BagMatrix ( )
```

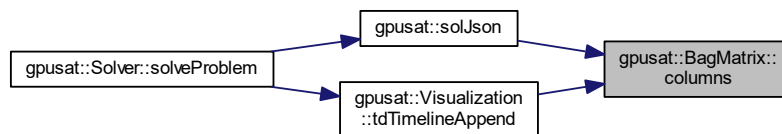
## 5.1.3 Member Function Documentation

### 5.1.3.1 columns()

```
size_t gpusat::BagMatrix::columns ( ) [inline]
```

Referenced by gpusat::solJson(), and gpusat::Visualization::tdTimelineAppend().

Here is the caller graph for this function:



### 5.1.3.2 operator() [1/2]

```
cl_long & gpusat::BagMatrix::operator() (
    size_t i,
    size_t j )
```

### 5.1.3.3 operator() [2/2]

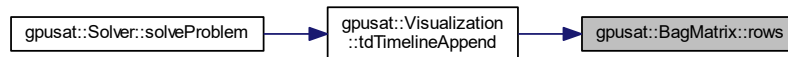
```
cl_long gpusat::BagMatrix::operator() (
    size_t i,
    size_t j ) const
```

#### 5.1.3.4 rows()

```
size_t gpusat::BagMatrix::rows ( ) [inline]
```

Referenced by `gpusat::Visualization::tdTimelineAppend()`.

Here is the caller graph for this function:



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

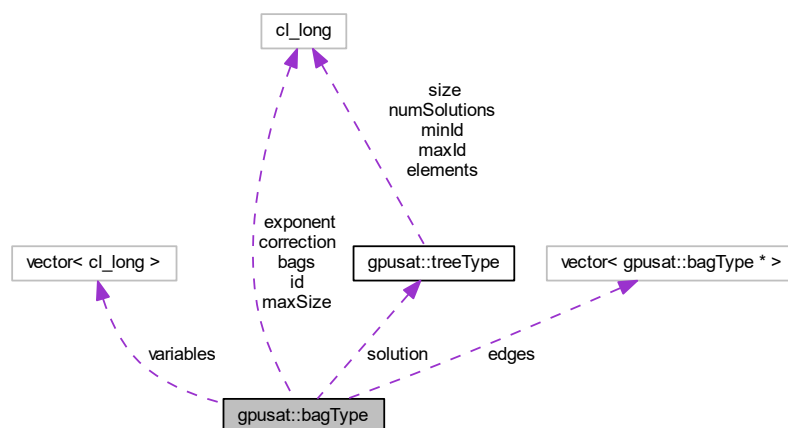
- [include/visualization.h](#)
- [src/visualization.cpp](#)

## 5.2 gpusat::bagType Struct Reference

type for a bag in the tree decomposition

```
#include <types.h>
```

Collaboration diagram for `gpusat::bagType`:



## Public Attributes

- `cl_long` `bags` = 0
- `cl_long` `correction` = 0
- `std::vector< bagType * >` `edges`
- `cl_long` `exponent` = 0
- `cl_long` `id` = 0
- `cl_long` `maxSize` = 0
- `treeType` \* `solution`
- `std::vector< cl_long >` `variables`

## 5.2.1 Detailed Description

type for a bag in the tree decomposition

## 5.2.2 Member Data Documentation

### 5.2.2.1 bags

```
cl_long gpusat::bagType::bags = 0
```

Referenced by `gpusat::printbagType()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

### 5.2.2.2 correction

```
cl_long gpusat::bagType::correction = 0
```

Referenced by `gpusat::printbagType()`, `gpusat::solJson()`, `gpusat::solutiontable()`, `gpusat::Solver::solveIntroduceForget()`, and `gpusat::Solver::solveJoin()`.

### 5.2.2.3 edges

```
std::vector<bagType *> gpusat::bagType::edges
```

Referenced by `gpusat::Preprocessor::preprocessDecomp()`, `gpusat::printbagType()`, `gpusat::Solver::solveProblem()`, and `gpusat::Visualization::visuTreeDec()`.

#### 5.2.2.4 exponent

```
cl_long gpusat::bagType::exponent = 0
```

Referenced by `gpusat::Solver::cleanTree()`, `gpusat::printbagType()`, `gpusat::Solver::solveIntroduceForget()`, and `gpusat::Solver::solveJoin()`.

#### 5.2.2.5 id

```
cl_long gpusat::bagType::id = 0
```

Referenced by `gpusat::compTreeType()`, `gpusat::printbagType()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveProblem()`, and `gpusat::Visualization::visuTreeDec()`.

#### 5.2.2.6 maxSize

```
cl_long gpusat::bagType::maxSize = 0
```

Referenced by `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

#### 5.2.2.7 solution

```
treeType* gpusat::bagType::solution
```

Referenced by `gpusat::solJson()`, `gpusat::solutiontable()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

#### 5.2.2.8 variables

```
std::vector<cl_long> gpusat::bagType::variables
```

Referenced by `main()`, `gpusat::Preprocessor::preprocessDecomp()`, `gpusat::printbagType()`, `gpusat::solJson()`, `gpusat::solutiontable()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

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

- `include/types.h`

## 5.3 gpusat::CNFParser Class Reference

```
#include <gpusatparser.h>
```

### Public Member Functions

- [CNFParser](#) (bool weighted)
- [satformulaType parseSatFormula](#) (std::string formula)

### 5.3.1 Constructor & Destructor Documentation

#### 5.3.1.1 CNFParser()

```
gpusat::CNFParser::CNFParser (
    bool weighted )
```

Constructor for an CNF parser.

#### Parameters

<i>weighted</i>	indicates if weights should be associated with literals
-----------------	---

### 5.3.2 Member Function Documentation

#### 5.3.2.1 parseSatFormula()

```
satformulaType gpusat::CNFParser::parseSatFormula (
    std::string formula )
```

generates a sat formula from a given string

#### Parameters

<i>formula</i>	the string representation of the sat formula
----------------	--

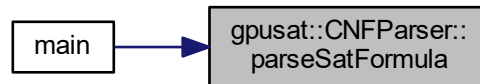
#### Returns

the sat formula

References [gpusat::satformulaType::numVars](#), [gpusat::satformulaType::numWeights](#), and [gpusat::satformulaType::variableWeights](#).

Referenced by `main()`.

Here is the caller graph for this function:



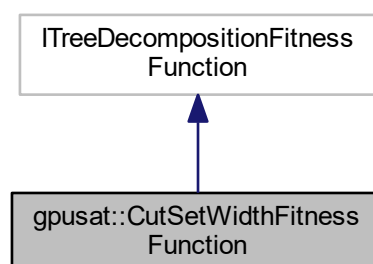
The documentation for this class was generated from the following files:

- [include/gpusatparser.h](#)
- [src/gpusatparser.cpp](#)

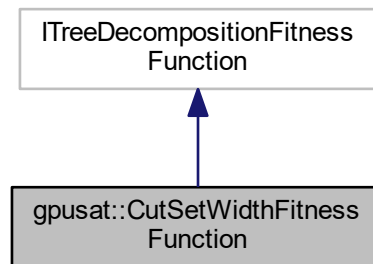
## 5.4 gpusat::CutSetWidthFitnessFunction Class Reference

```
#include <CutSetWidthFitnessFunction.h>
```

Inheritance diagram for `gpusat::CutSetWidthFitnessFunction`:



Collaboration diagram for `gpusat::CutSetWidthFitnessFunction`:



## Public Member Functions

- `CutSetWidthFitnessFunction()`=default
- `~CutSetWidthFitnessFunction()`=default
- `CutSetWidthFitnessFunction * clone` (void) const override
- `htd::FitnessEvaluation * fitness` (const `htd::IMultiHypergraph &graph`, const `htd::ITreeDecomposition &decomposition`) const override
- `double getMaxCutSetSize` (const `htd::ITreeDecomposition &decomposition`) const

### 5.4.1 Detailed Description

Fitness function that first minimizes the cut set size and then the width.

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 CutSetWidthFitnessFunction()

```
gpusat::CutSetWidthFitnessFunction::CutSetWidthFitnessFunction ( ) [default]
```

Referenced by `clone()`.

Here is the caller graph for this function:





### 5.4.2.2 ~CutSetWidthFitnessFunction()

```
gpusat::CutSetWidthFitnessFunction::~~CutSetWidthFitnessFunction ( ) [default]
```

## 5.4.3 Member Function Documentation

### 5.4.3.1 clone()

```
CutSetWidthFitnessFunction * gpusat::CutSetWidthFitnessFunction::clone (
    void ) const [override]
```

References `CutSetWidthFitnessFunction()`.

Here is the call graph for this function:

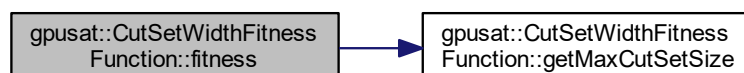


### 5.4.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::CutSetWidthFitnessFunction::fitness (
    const htd::IMultiHypergraph & graph,
    const htd::ITreeDecomposition & decomposition ) const [override]
```

References `getMaxCutSetSize()`.

Here is the call graph for this function:



### 5.4.3.3 getMaxCutSetSize()

```
double gpusat::CutSetWidthFitnessFunction::getMaxCutSetSize (
    const htd::ITreeDecomposition & decomposition ) const
```

Referenced by fitness().

Here is the caller graph for this function:



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

- include/FitnessFunctions/CutSetWidthFitnessFunction.h
- src/FitnessFunctions/CutSetWidthFitnessFunction.cpp

## 5.5 gpusat::Decomposer Class Reference

```
#include <decomposer.h>
```

### Static Public Member Functions

- static std::string [computeDecomposition](#) (std::string formula, htd::ITreeDecompositionFitnessFunction \*fitness, size\_t n)

### 5.5.1 Member Function Documentation

#### 5.5.1.1 computeDecomposition()

```
std::string gpusat::Decomposer::computeDecomposition (
    std::string formula,
    htd::ITreeDecompositionFitnessFunction * fitness,
    size_t n ) [static]
```

computes the decomposition of the primal graph of the given formula

## Parameters

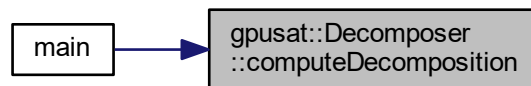
<i>formula</i>	the formula in cnf format
<i>fitness</i>	the fitness function
<i>n</i>	number of iterations for the fitness function

## Returns

the decomposition in td format

Referenced by `main()`.

Here is the caller graph for this function:



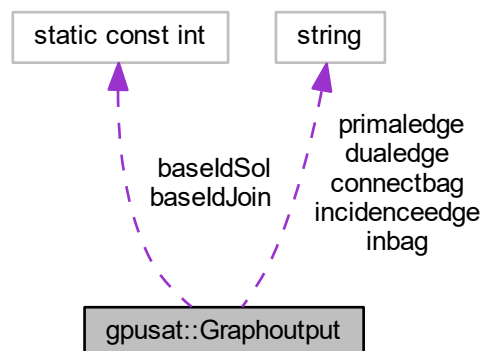
The documentation for this class was generated from the following files:

- [include/decomposer.h](#)
- [src/decomposer.cpp](#)

## 5.6 gpusat::Graphoutput Class Reference

```
#include <graphoutput.h>
```

Collaboration diagram for gpusat::Graphoutput:



## Public Member Functions

- [Graphoutput](#) (std::string filename)
- std::string [getFilename](#) ()
- void [graphEdgeSet](#) ([treedecType](#) \*dec)  
*Addes the edges from the tree-decomposition into the graph.*
- void [graphEnd](#) ()
- void [graphStart](#) ([treedecType](#) \*dec)
- bool [isEnabled](#) ()
- void [neo4jSat](#) ([satformulaType](#) \*satFormula)  
*Output Cypherquery to create the nodes from the SAT formula*
- void [neo4jTD](#) ([treedecType](#) \*treeDec)
- void [nodeBag](#) (unsigned int id, std::string solution)  
*Create one node with the corresponding solution in a connected extra bag.*
- void [nodeJoin](#) (unsigned int id1, unsigned int id2, std::string solution)  
*Creates a node with the solution that joins two bags. Two edges are added to connect the solution.*

## Static Public Attributes

- static const int [baselIdJoin](#) = 4 \* [baselIdSol](#)
- static const int [baselIdSol](#) = 1'000'000
- static const std::string [connectbag](#) = "USES\_BAG"
- static const std::string [dualedge](#) = "SHARE\_VAR"
- static const std::string [inbag](#) = "CONTAINS"
- static const std::string [incidenceedge](#) = "VAR\_IN\_CLAUSE"
- static const std::string [primaledge](#) = "SHARE\_CLAUSE"

## 5.6.1 Constructor & Destructor Documentation

### 5.6.1.1 Graphoutput()

```
gpusat::Graphoutput::Graphoutput (
    std::string filename ) [inline]
```

## 5.6.2 Member Function Documentation

### 5.6.2.1 getFilename()

```
std::string gpusat::Graphoutput::getFilename ( ) [inline]
```

### 5.6.2.2 graphEdgeSet()

```
void gpusat::Graphoutput::graphEdgeSet (
    treedecType * dec )
```

Addes the edges from the tree-decomposition into the graph.

Any outgoing edges from id '0' are not added.

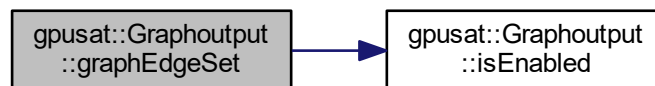
## Parameters

<i>dec</i>	Pointer to the tree-decomposition containing used edges.
------------	--

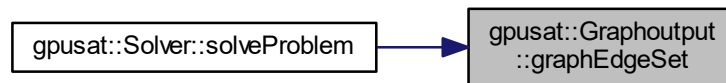
References gpusat::treedecType::bags, and isEnabled().

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



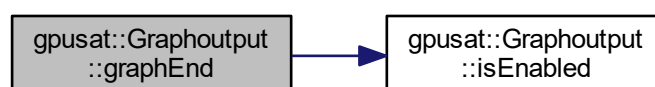
### 5.6.2.3 graphEnd()

```
void gpusat::Graphoutput::graphEnd ( )
```

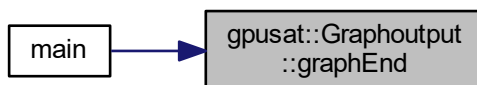
References isEnabled().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



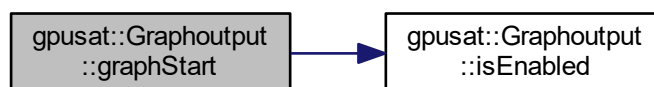
#### 5.6.2.4 graphStart()

```
void gpusat::Graphoutput::graphStart (  
    treedecType * dec )
```

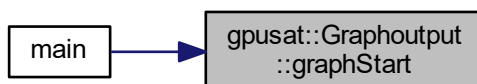
References gpusat::treedecType::bags, and isEnabled().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

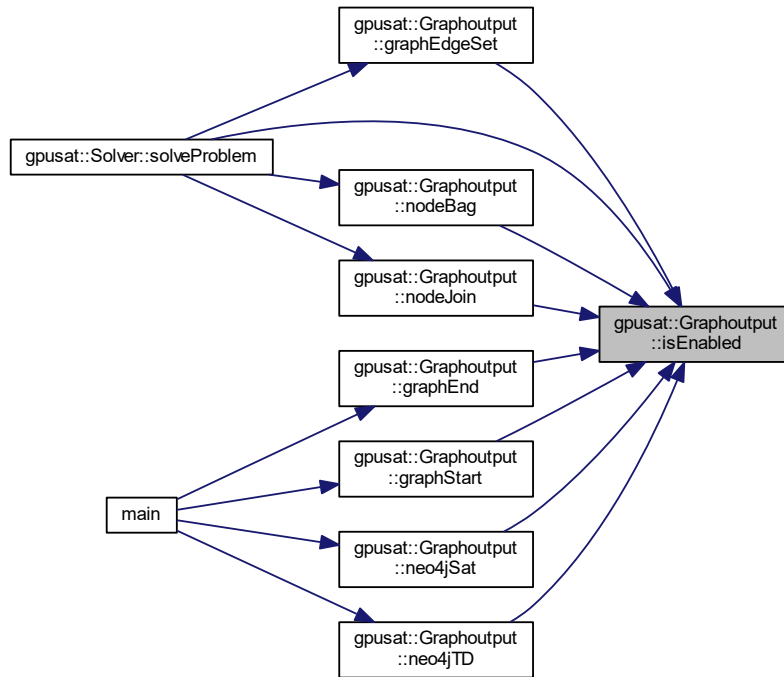


## 5.6.2.5 isEnabled()

```
bool gpusat::Graphoutput::isEnabled ( ) [inline]
```

Referenced by graphEdgeSet(), graphEnd(), graphStart(), neo4jSat(), neo4jTD(), nodeBag(), nodeJoin(), and gpusat::Solver::solveProblem().

Here is the caller graph for this function:



## 5.6.2.6 neo4jSat()

```
void gpusat::Graphoutput::neo4jSat (
    satformulaType * satFormula )
```

Output Cypherquery to create the nodes from the SAT formula

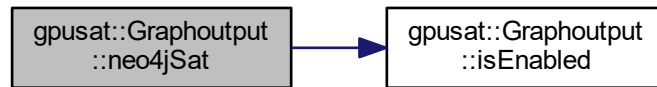
## Parameters

<i>satFormula</i>	The sat formula.
-------------------	------------------

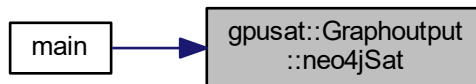
References gpusat::satformulaType::clauses, dualedge, gpusat::satformulaType::facts, incidenceedge, isEnabled(), gpusat::satformulaType::numVars, primaledge, and gpusat::satformulaType::unsat.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



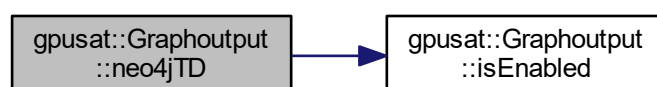
#### 5.6.2.7 neo4jTD()

```
void gpusat::Graphoutput::neo4jTD (
    treedecType * treeDec )
```

References `gpusat::treedecType::bags`, `connectbag`, `inbag`, `isEnabled()`, `gpusat::treedecType::numb`, `gpusat::treedecType::numVars`, and `gpusat::treedecType::width`.

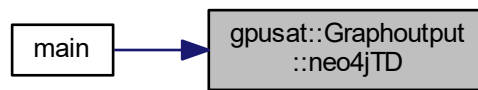
Referenced by `main()`.

Here is the call graph for this function:





Here is the caller graph for this function:



#### 5.6.2.8 nodeBag()

```
void gpusat::Graphoutput::nodeBag (
    unsigned int id,
    std::string solution )
```

Create one node with the corresponding solution in a connected extra bag.

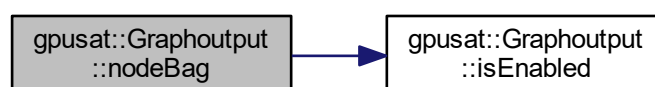
##### Parameters

<i>id</i>	The identifier of the bag.
<i>solution</i>	The solution in string form.

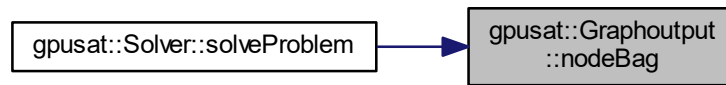
References isEnabled().

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.6.2.9 nodeJoin()

```

void gpusat::Graphoutput::nodeJoin (
    unsigned int id1,
    unsigned int id2,
    std::string solution )
  
```

Creates a node with the solution that joins two bags. Two edges are added to connect the solution.

##### Parameters

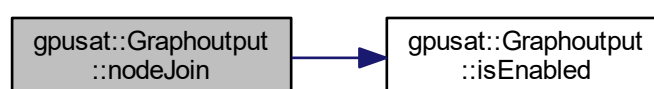
<i>id1</i>	The id1.
<i>id2</i>	The id2.
<i>solution</i>	The solution.

TODO Edit XML Comment Template for nodeJoin

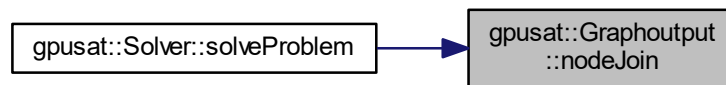
References isEnabled().

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



## 5.6.3 Member Data Documentation

### 5.6.3.1 baseIdJoin

```
const int gpusat::Graphoutput::baseIdJoin = 4 * baseIdSol [static]
```

### 5.6.3.2 baseIdSol

```
const int gpusat::Graphoutput::baseIdSol = 1'000'000 [static]
```

### 5.6.3.3 connectbag

```
const std::string gpusat::Graphoutput::connectbag = "USES_BAG" [static]
```

Referenced by `neo4jTD()`.

### 5.6.3.4 dualedge

```
const std::string gpusat::Graphoutput::dualedge = "SHARE_VAR" [static]
```

Referenced by `neo4jSat()`.

#### 5.6.3.5 inbag

```
const std::string gpusat::Graphoutput::inbag = "CONTAINS" [static]
```

Referenced by neo4jTD().

#### 5.6.3.6 incidenceedge

```
const std::string gpusat::Graphoutput::incidenceedge = "VAR_IN_CLAUSE" [static]
```

Referenced by neo4jSat().

#### 5.6.3.7 primaledge

```
const std::string gpusat::Graphoutput::primaledge = "SHARE_CLAUSE" [static]
```

Referenced by neo4jSat().

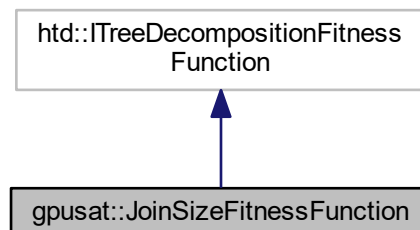
The documentation for this class was generated from the following files:

- [include/graphoutput.h](#)
- [src/graphoutput.cpp](#)

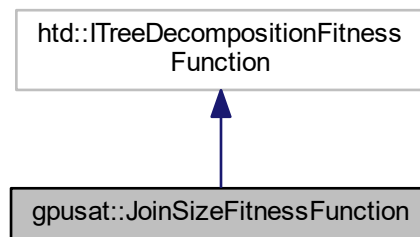
## 5.7 gpusat::JoinSizeFitnessFunction Class Reference

```
#include <JoinSizeFitnessFunction.h>
```

Inheritance diagram for gpusat::JoinSizeFitnessFunction:



Collaboration diagram for gpusat::JoinSizeFitnessFunction:



## Public Member Functions

- [JoinSizeFitnessFunction](#) ()=default
- [~JoinSizeFitnessFunction](#) ()=default
- [JoinSizeFitnessFunction](#) \* [clone](#) (void) const override
- htd::FitnessEvaluation \* [fitness](#) (const htd::IMultiHypergraph &graph, const htd::ITreeDecomposition &decomposition) const override

### 5.7.1 Detailed Description

Fitness function that minimizes the number of vertices in a join node and then the width.

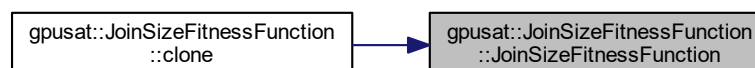
### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 JoinSizeFitnessFunction()

```
gpusat::JoinSizeFitnessFunction::JoinSizeFitnessFunction ( ) [default]
```

Referenced by [clone\(\)](#).

Here is the caller graph for this function:



### 5.7.2.2 ~JoinSizeFitnessFunction()

```
gpusat::JoinSizeFitnessFunction::~~JoinSizeFitnessFunction ( ) [default]
```

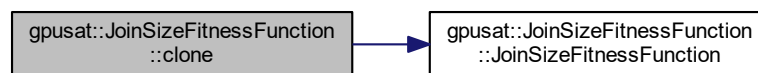
## 5.7.3 Member Function Documentation

### 5.7.3.1 clone()

```
JoinSizeFitnessFunction * gpusat::JoinSizeFitnessFunction::clone (
    void ) const [override]
```

References [JoinSizeFitnessFunction\(\)](#).

Here is the call graph for this function:



### 5.7.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::JoinSizeFitnessFunction::fitness (
    const htd::IMultiHypergraph & graph,
    const htd::ITreeDecomposition & decomposition ) const [override]
```

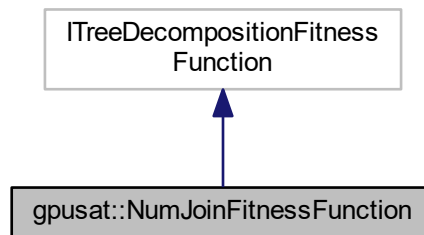
The documentation for this class was generated from the following files:

- [include/FitnessFunctions/JoinSizeFitnessFunction.h](#)
- [src/FitnessFunctions/JoinSizeFitnessFunction.cpp](#)

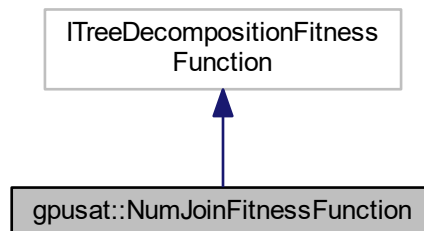
## 5.8 gpusat::NumJoinFitnessFunction Class Reference

```
#include <NumJoinFitnessFunction.h>
```

Inheritance diagram for gpusat::NumJoinFitnessFunction:



Collaboration diagram for gpusat::NumJoinFitnessFunction:



### Public Member Functions

- [NumJoinFitnessFunction](#) ()=default
- [~NumJoinFitnessFunction](#) ()=default
- [NumJoinFitnessFunction \\* clone](#) (void) const override
- `htd::FitnessEvaluation * fitness` (const `htd::IMultiHypergraph &graph`, const `htd::ITreeDecomposition &decomposition`) const override

#### 5.8.1 Detailed Description

Fitness function that minimizes the number of join nodes and then the width.

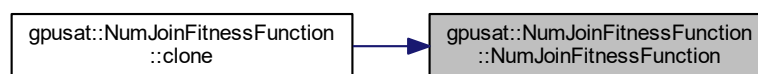
## 5.8.2 Constructor & Destructor Documentation

### 5.8.2.1 NumJoinFitnessFunction()

```
gpusat::NumJoinFitnessFunction::NumJoinFitnessFunction ( ) [default]
```

Referenced by clone().

Here is the caller graph for this function:



### 5.8.2.2 ~NumJoinFitnessFunction()

```
gpusat::NumJoinFitnessFunction::~~NumJoinFitnessFunction ( ) [default]
```

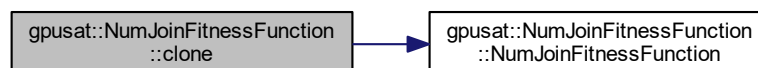
## 5.8.3 Member Function Documentation

### 5.8.3.1 clone()

```
NumJoinFitnessFunction * gpusat::NumJoinFitnessFunction::clone (
    void ) const [override]
```

References NumJoinFitnessFunction().

Here is the call graph for this function:





### 5.8.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::NumJoinFitnessFunction::fitness (
    const htd::IMultiHypergraph & graph,
    const htd::ITreeDecomposition & decomposition ) const [override]
```

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

- include/FitnessFunctions/[NumJoinFitnessFunction.h](#)
- src/FitnessFunctions/[NumJoinFitnessFunction.cpp](#)

## 5.9 gpusat::Preprocessor Class Reference

```
#include <gpusatpreprocessor.h>
```

### Static Public Member Functions

- static void [preprocessDecomp](#) ([bagType](#) \*decomp, cl\_long combineWidth)
- static void [preprocessFacts](#) ([treedecType](#) &decomp, [satformulaType](#) &formula, cl\_double &defaultWeight)

### 5.9.1 Member Function Documentation

#### 5.9.1.1 preprocessDecomp()

```
void gpusat::Preprocessor::preprocessDecomp (
    bagType * decomp,
    cl_long combineWidth ) [static]
```

preprocess the tree decomposition

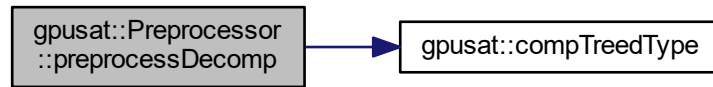
#### Parameters

<i>decomp</i>	the tree decomposition
<i>combineWidth</i>	max width to combine bags

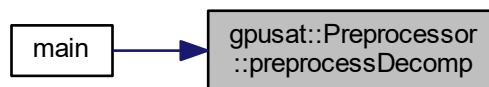
References [gpusat::compTreedListType\(\)](#), [gpusat::bagType::edges](#), and [gpusat::bagType::variables](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.9.1.2 preprocessFacts()

```

void gpusat::Preprocessor::preprocessFacts (
    treedecType & decomp,
    satformulaType & formula,
    cl_double & defaultWeight ) [static]
  
```

removes facts from the sat formula

#### Parameters

<i>decomp</i>	the tree decomposition
<i>formula</i>	the sat formula
<i>defaultWeight</i>	for WMC the product of the weights of the removed literals

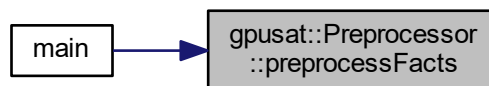
References `gpusat::treedecType::bags`, `gpusat::satformulaType::clauses`, `gpusat::compVars()`, `gpusat::satformulaType::facts`, `gpusat::treedecType::numVars`, `gpusat::satformulaType::numVars`, `gpusat::satformulaType::numWeights`, `gpusat::satformulaType::unsat`, and `gpusat::satformulaType::variableWeights`.

Referenced by `main()`.

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:

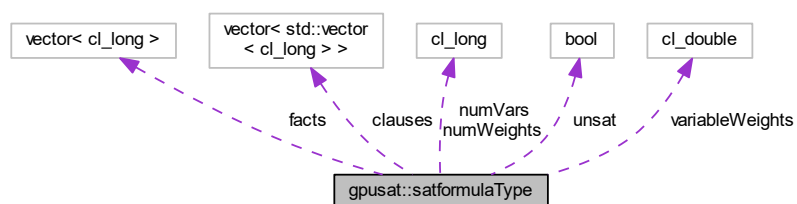
- [include/gpusatpreprocessor.h](#)
- [src/gpusatpreprocessor.cpp](#)

## 5.10 gpusat::satformulaType Struct Reference

type for saving the sat formula

```
#include <types.h>
```

Collaboration diagram for gpusat::satformulaType:



## Public Attributes

- `std::vector< std::vector< cl_long > >` `clauses`
- `std::vector< cl_long >` `facts`
- `cl_long` `numVars` = 0
- `cl_long` `numWeights` = 0
- `bool` `unsat` = false
- `cl_double *` `variableWeights` = nullptr

### 5.10.1 Detailed Description

type for saving the sat formula

### 5.10.2 Member Data Documentation

#### 5.10.2.1 clauses

```
std::vector<std::vector<cl_long> > gpusat::satformulaType::clauses
```

Referenced by `gpusat::Graphoutput::neo4jSat()`, `gpusat::Preprocessor::preprocessFacts()`, `gpusat::Solver::solve↔` `IntroduceForget()`, and `gpusat::Visualization::visuClauses()`.

#### 5.10.2.2 facts

```
std::vector<cl_long> gpusat::satformulaType::facts
```

Referenced by `gpusat::Graphoutput::neo4jSat()`, and `gpusat::Preprocessor::preprocessFacts()`.

#### 5.10.2.3 numVars

```
cl_long gpusat::satformulaType::numVars = 0
```

Referenced by `gpusat::Graphoutput::neo4jSat()`, `gpusat::CNFParser::parseSatFormula()`, and `gpusat::↔` `Preprocessor::preprocessFacts()`.

### 5.10.2.4 numWeights

```
cl_long gpusat::satformulaType::numWeights = 0
```

Referenced by `gpusat::CNFParser::parseSatFormula()`, `gpusat::Preprocessor::preprocessFacts()`, `gpusat::Solver::solveIntroduceForget()`, and `gpusat::Solver::solveJoin()`.

### 5.10.2.5 unsat

```
bool gpusat::satformulaType::unsat = false
```

Referenced by `main()`, `gpusat::Graphoutput::neo4jSat()`, and `gpusat::Preprocessor::preprocessFacts()`.

### 5.10.2.6 variableWeights

```
cl_double* gpusat::satformulaType::variableWeights = nullptr
```

Referenced by `gpusat::CNFParser::parseSatFormula()`, `gpusat::Preprocessor::preprocessFacts()`, `gpusat::Solver::solveIntroduceForget()`, and `gpusat::Solver::solveJoin()`.

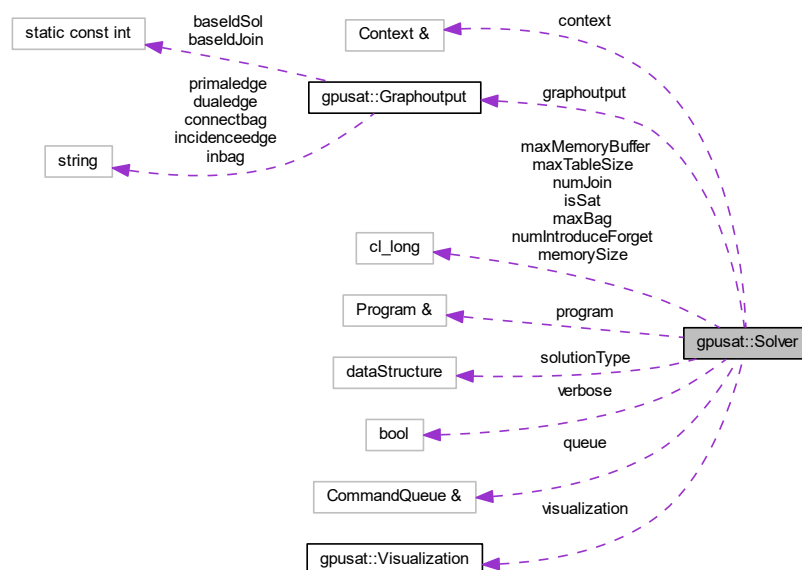
The documentation for this struct was generated from the following file:

- [include/types.h](#)

## 5.11 gpusat::Solver Class Reference

```
#include <solver.h>
```

Collaboration diagram for `gpusat::Solver`:



## Public Member Functions

- [Solver](#) (cl::Context &context\_, cl::CommandQueue &queue\_, cl::Program &program\_, cl\_long memorySize\_, cl\_long maxMemoryBuffer\_, [dataStructure](#) solutionType\_, cl\_long maxBag\_, bool verbose\_, [Graphoutput](#) \*graphoutput\_, [Visualization](#) \*visualization\_)
- void [solveProblem](#) ([treeDecType](#) &decomp, [satformulaType](#) &formula, [bagType](#) &node, [bagType](#) &pnode, [nodeType](#) lastNode)

*Solves the sat formula with a given decomposition.*

## Public Attributes

- [Graphoutput](#) \* [graphoutput](#)
- cl\_long [isSat](#) = 1
- cl\_long [maxBag](#) = 0
- cl\_long [maxMemoryBuffer](#) = 0
- cl\_long [maxTableSize](#) = 0
- cl\_long [numIntroduceForget](#) = 0
- cl\_long [numJoin](#) = 0
- [dataStructure](#) [solutionType](#) = [dataStructure::TREE](#)
- bool [verbose](#) = false
- [Visualization](#) \* [visualization](#)

## Protected Member Functions

- void [cleanTree](#) ([treeType](#) &table, cl\_long size, cl\_long numVars, [bagType](#) &node, cl\_long nextSize)
- void [combineTree](#) ([treeType](#) &to, [treeType](#) &from, cl\_long numVars)
- void [solveIntroduceForget](#) ([satformulaType](#) &formula, [bagType](#) &pnode, [bagType](#) &node, [bagType](#) &cnode, bool leaf, [nodeType](#) nextNode)
- void [solveJoin](#) ([bagType](#) &node, [bagType](#) &edge1, [bagType](#) &edge2, [satformulaType](#) &formula, [nodeType](#) nextNode)

## Protected Attributes

- cl::Context & [context](#)
- cl\_long [memorySize](#)
- cl::Program & [program](#)
- cl::CommandQueue & [queue](#)

### 5.11.1 Detailed Description

Organizes the process of solving a sat problem in the given context.

### 5.11.2 Constructor & Destructor Documentation

### 5.11.2.1 Solver()

```
gpusat::Solver::Solver (
    cl::Context & context_,
    cl::CommandQueue & queue_,
    cl::Program & program_,
    cl_long memorySize_,
    cl_long maxMemoryBuffer_,
    dataStructure solutionType_,
    cl_long maxBag_,
    bool verbose_,
    Graphoutput * graphoutput_,
    Visualization * visualization_ ) [inline]
```

#### Parameters

<i>context_</i>	the given context as hardware
<i>queue_</i>	
<i>program_</i>	
<i>memorySize_</i>	
<i>maxMemory↵ Buffer_</i>	
<i>solutionType_</i>	
<i>maxBag_</i>	
<i>verbose_</i>	
<i>graphoutput_</i>	
<i>visualization_</i>	

## 5.11.3 Member Function Documentation

### 5.11.3.1 cleanTree()

```
void gpusat::Solver::cleanTree (
    treeType & table,
    cl_long size,
    cl_long numVars,
    bagType & node,
    cl_long nextSize ) [protected]
```

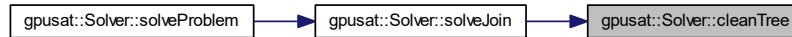
#### Parameters

<i>table</i>	
<i>size</i>	
<i>numVars</i>	
<i>node</i>	

References `context`, `gpusat::treeType::elements`, `gpusat::bagType::exponent`, `gpusat::treeType::maxId`, `gpusat↵  
::treeType::minId`, `gpusat::treeType::numSolutions`, `program`, `queue`, and `gpusat::treeType::size`.

Referenced by solveJoin().

Here is the caller graph for this function:



### 5.11.3.2 combineTree()

```

void gpusat::Solver::combineTree (
    treeType & to,
    treeType & from,
    cl_long numVars ) [protected]
  
```

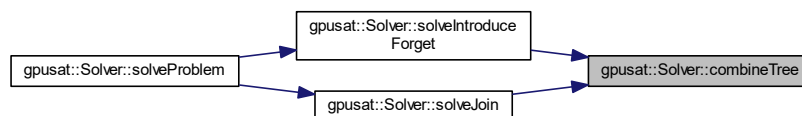
#### Parameters

<i>to</i>	
<i>from</i>	
<i>numVars</i>	

References context, gpusat::treeType::elements, gpusat::treeType::maxId, gpusat::treeType::minId, gpusat::treeType::numSolutions, program, queue, and gpusat::treeType::size.

Referenced by solveIntroduceForget(), and solveJoin().

Here is the caller graph for this function:



### 5.11.3.3 solveIntroduceForget()

```

void gpusat::Solver::solveIntroduceForget (
    satformulaType & formula,
    bagType & pnode,
    bagType & node,
  
```



```
    bagType & cnode,  
    bool leaf,  
    nodeType nextNode ) [protected]
```

function to solve an introduce forget node

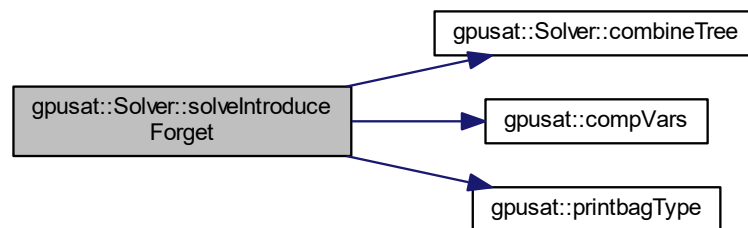
## Parameters

<i>formula</i>	the sat formula
<i>pnode</i>	the parent of the current node
<i>node</i>	the current node
<i>cnode</i>	the child of the current node
<i>leaf</i>	indicates that the current node is a leaf node

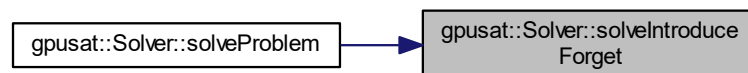
References `gpusat::ARRAY`, `gpusat::bagType::bags`, `gpusat::satformulaType::clauses`, `combineTree()`, `gpusat::compVars()`, `context`, `gpusat::bagType::correction`, `gpusat::treeType::elements`, `gpusat::bagType::exponent`, `gpusat::bagType::id`, `gpusat::INTRODUCEFORGET`, `isSat`, `gpusat::JOIN`, `maxBag`, `gpusat::treeType::maxId`, `maxMemoryBuffer`, `gpusat::bagType::maxSize`, `maxTableSize`, `memorySize`, `gpusat::treeType::minId`, `numIntroduceForget`, `gpusat::treeType::numSolutions`, `gpusat::satformulaType::numWeights`, `gpusat::printbagType()`, `program`, `queue`, `gpusat::treeType::size`, `gpusat::bagType::solution`, `solutionType`, `gpusat::TREE`, `gpusat::bagType::variables`, `gpusat::satformulaType::variableWeights`, and `verbose`.

Referenced by `solveProblem()`.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.11.3.4 solveJoin()

```
void gpusat::Solver::solveJoin (
    bagType & node,
    bagType & edge1,
    bagType & edge2,
    satformulaType & formula,
    nodeType nextNode ) [protected]
```

function to solve a join node

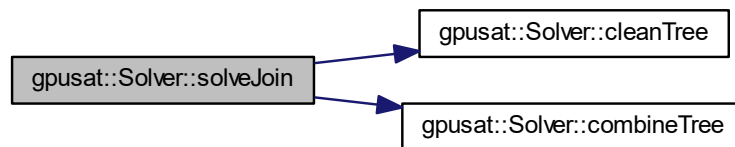
## Parameters

<i>node</i>	the node to save the solutions in
<i>edge1</i>	the first child node
<i>edge2</i>	the second child node
<i>formula</i>	the sat formula

References `gpusat::ARRAY`, `gpusat::bagType::bags`, `cleanTree()`, `combineTree()`, `context`, `gpusat::bagType::correction`, `gpusat::treeType::elements`, `gpusat::bagType::exponent`, `gpusat::INTRODUCEFORGET`, `isSat`, `gpusat::JOIN`, `maxBag`, `gpusat::treeType::maxId`, `maxMemoryBuffer`, `gpusat::bagType::maxSize`, `maxTableSize`, `memorySize`, `gpusat::treeType::minId`, `numJoin`, `gpusat::treeType::numSolutions`, `gpusat::satformulaType::numWeights`, `program`, `queue`, `gpusat::treeType::size`, `gpusat::bagType::solution`, `solutionType`, `gpusat::TREE`, `gpusat::bagType::variables`, and `gpusat::satformulaType::variableWeights`.

Referenced by `solveProblem()`.

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.11.3.5 solveProblem()

```

void gpusat::Solver::solveProblem (
    treedecType & decomp,
    satformulaType & formula,
    bagType & node,
    bagType & pnode,
    nodeType lastNode )
  
```

Solves the sat formula with a given decomposition.

function to solve the sat problem

## Parameters

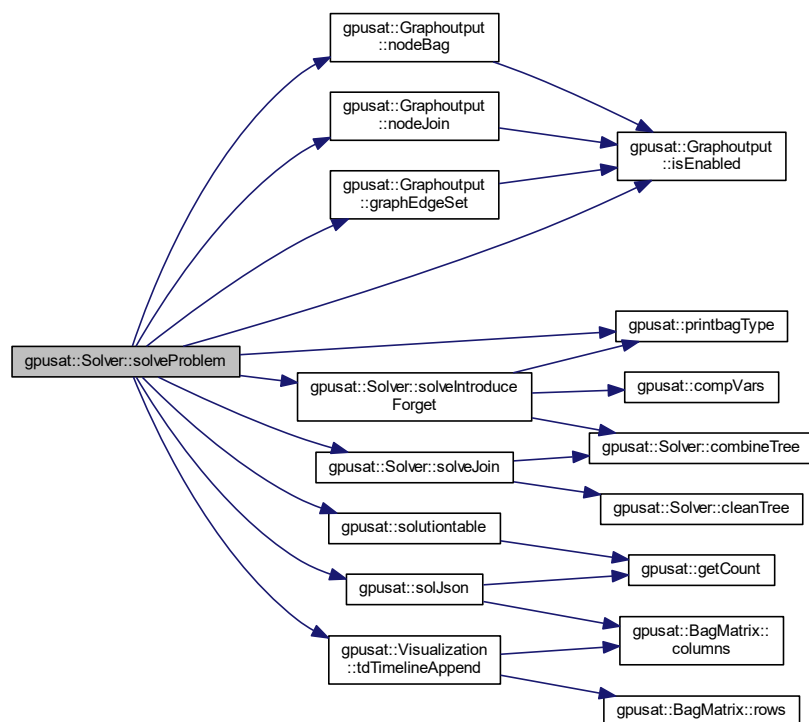
<i>decomp</i>	the tree decomposition
<i>formula</i>	the sat formula
<i>node</i>	the node to start from in the tree decomposition

## Parameters

<i>decomp</i>	The decomposition of the
<i>formula</i>	The formula.
<i>node</i>	The node.
<i>pnode</i>	The pnode.
<i>lastNode</i>	The last node.

References gpusat::bagType::bags, gpusat::bagType::edges, gpusat::treeType::elements, gpusat::Graphoutput↵  
::graphEdgeSet(), graphoutput, gpusat::bagType::id, gpusat::INTRODUCEFORGET, gpusat::Graphoutput::is↵  
Enabled(), isSat, gpusat::JOIN, gpusat::treeType::maxId, gpusat::bagType::maxSize, gpusat::treeType::minId,  
gpusat::Graphoutput::nodeBag(), gpusat::Graphoutput::nodeJoin(), gpusat::treeType::numSolutions, gpusat↵  
::printbagType(), gpusat::treeType::size, gpusat::solJson(), gpusat::bagType::solution, gpusat::solutiontable(),  
solutionType, solveIntroduceForget(), solveJoin(), gpusat::Visualization::tdTimelineAppend(), gpusat::bagType↵  
::variables, verbose, and visualization.

Here is the call graph for this function:



## 5.11.4 Member Data Documentation

#### 5.11.4.1 context

```
cl::Context& gpusat::Solver::context [protected]
```

Referenced by `cleanTree()`, `combineTree()`, `solveIntroduceForget()`, and `solveJoin()`.

#### 5.11.4.2 graphoutput

```
Graphoutput* gpusat::Solver::graphoutput
```

Referenced by `solveProblem()`.

#### 5.11.4.3 isSat

```
cl_long gpusat::Solver::isSat = 1
```

Referenced by `solveIntroduceForget()`, `solveJoin()`, and `solveProblem()`.

#### 5.11.4.4 maxBag

```
cl_long gpusat::Solver::maxBag = 0
```

Referenced by `solveIntroduceForget()`, and `solveJoin()`.

#### 5.11.4.5 maxMemoryBuffer

```
cl_long gpusat::Solver::maxMemoryBuffer = 0
```

Referenced by `solveIntroduceForget()`, and `solveJoin()`.

#### 5.11.4.6 maxTableSize

```
cl_long gpusat::Solver::maxTableSize = 0
```

Referenced by `main()`, `solveIntroduceForget()`, and `solveJoin()`.

#### 5.11.4.7 memorySize

```
cl_long gpusat::Solver::memorySize [protected]
```

Referenced by solveIntroduceForget(), and solveJoin().

#### 5.11.4.8 numIntroduceForget

```
cl_long gpusat::Solver::numIntroduceForget = 0
```

Referenced by main(), and solveIntroduceForget().

#### 5.11.4.9 numJoin

```
cl_long gpusat::Solver::numJoin = 0
```

Referenced by main(), and solveJoin().

#### 5.11.4.10 program

```
cl::Program& gpusat::Solver::program [protected]
```

Referenced by cleanTree(), combineTree(), solveIntroduceForget(), and solveJoin().

#### 5.11.4.11 queue

```
cl::CommandQueue& gpusat::Solver::queue [protected]
```

Referenced by cleanTree(), combineTree(), solveIntroduceForget(), and solveJoin().

#### 5.11.4.12 solutionType

```
dataStructure gpusat::Solver::solutionType = dataStructure::TREE
```

Referenced by solveIntroduceForget(), solveJoin(), and solveProblem().

#### 5.11.4.13 verbose

```
bool gpusat::Solver::verbose = false
```

Referenced by solveIntroduceForget(), and solveProblem().

#### 5.11.4.14 visualization

```
Visualization* gpusat::Solver::visualization
```

Referenced by solveProblem().

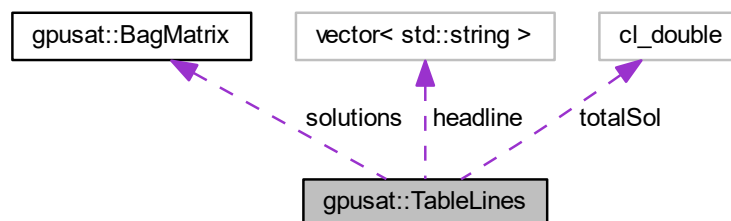
The documentation for this class was generated from the following files:

- include/solver.h
- src/solver.cpp

## 5.12 gpusat::TableLines Struct Reference

```
#include <visualization.h>
```

Collaboration diagram for gpusat::TableLines:



### Public Attributes

- std::vector< std::string > [headline](#)
- [BagMatrix](#) [solutions](#)
- cl\_double [totalSol](#)

#### 5.12.1 Detailed Description

Solutions in Timeline. Example: `[["id", "v2", "v4", "n Sol"], [0, 0, 0, 1], [1, 1, 0, 2], [2, 0, 1, 2], [3, 1, 1, 3]]`



## 5.12.2 Member Data Documentation

### 5.12.2.1 headline

`std::vector<std::string> gpusat::TableLines::headline`

Referenced by `gpusat::solJson()`, and `gpusat::Visualization::tdTimelineAppend()`.

### 5.12.2.2 solutions

`BagMatrix gpusat::TableLines::solutions`

Referenced by `gpusat::solJson()`, and `gpusat::Visualization::tdTimelineAppend()`.

### 5.12.2.3 totalSol

`cl_double gpusat::TableLines::totalSol`

Referenced by `gpusat::solJson()`, and `gpusat::Visualization::tdTimelineAppend()`.

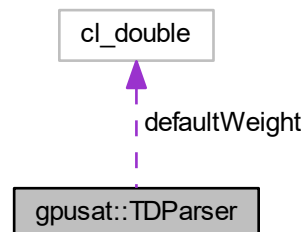
The documentation for this struct was generated from the following file:

- `include/visualization.h`

## 5.13 gpusat::TDParse Class Reference

```
#include <gpusatparser.h>
```

Collaboration diagram for `gpusat::TDParse`:



## Public Member Functions

- [TDParser](#) ()
- [treedecType](#) [parseTreeDecomp](#) (std::string *graph*, [satformulaType](#) &*formula*)

## Public Attributes

- cl\_double [defaultWeight](#) = 1.0

### 5.13.1 Constructor & Destructor Documentation

#### 5.13.1.1 TDParser()

```
gpusat::TDParser::TDParser ( ) [inline]
```

### 5.13.2 Member Function Documentation

#### 5.13.2.1 parseTreeDecomp()

```
treedecType gpusat::TDParser::parseTreeDecomp (
    std::string graph,
    satformulaType & formula )
```

generates a treedec from a given string

#### Parameters

<i>graph</i>	the string representation of the tree decomposition
--------------	---

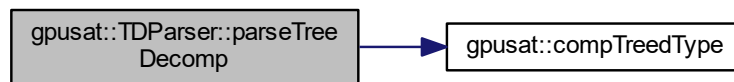
#### Returns

the tree decomposition

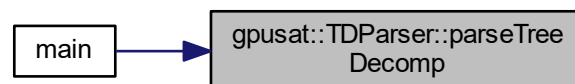
References [gpusat::treedecType::bags](#), [gpusat::compTreedType\(\)](#), and [gpusat::treedecType::numb](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.13.3 Member Data Documentation

#### 5.13.3.1 defaultWeight

```
cl_double gpusat::TDParseTree::defaultWeight = 1.0
```

Referenced by `main()`.

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

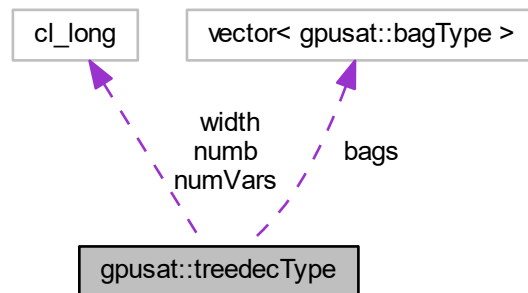
- [include/gpusatparser.h](#)
- [src/gpusatparser.cpp](#)

## 5.14 gpusat::treedecType Struct Reference

type for saving a tree decomposition

```
#include <types.h>
```

Collaboration diagram for `gpusat::treedecType`:



## Public Attributes

- `std::vector< bagType > bags`
- `cl_long numb = 0`
- `cl_long numVars = 0`
- `cl_long width = 0`

### 5.14.1 Detailed Description

type for saving a tree decomposition

### 5.14.2 Member Data Documentation

#### 5.14.2.1 bags

`std::vector<bagType> gpusat::treedecType::bags`

Referenced by `gpusat::Graphoutput::graphEdgeSet()`, `gpusat::Graphoutput::graphStart()`, `main()`, `gpusat::Graphoutput::neo4jTD()`, `gpusat::TDParser::parseTreeDecomp()`, `gpusat::Preprocessor::preprocessFacts()`, `gpusat::printtreedecType()`, and `gpusat::Visualization::visuTreeDec()`.

#### 5.14.2.2 numb

`cl_long gpusat::treedecType::numb = 0`

Referenced by `gpusat::Graphoutput::neo4jTD()`, `gpusat::TDParser::parseTreeDecomp()`, and `gpusat::printtreedecType()`.

### 5.14.2.3 numVars

```
cl_long gpusat::treedecType::numVars = 0
```

Referenced by `gpusat::Graphoutput::neo4jTD()`, `gpusat::Preprocessor::preprocessFacts()`, and `gpusat::Visualization::visuTreeDec()`.

### 5.14.2.4 width

```
cl_long gpusat::treedecType::width = 0
```

Referenced by `main()`, and `gpusat::Graphoutput::neo4jTD()`.

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

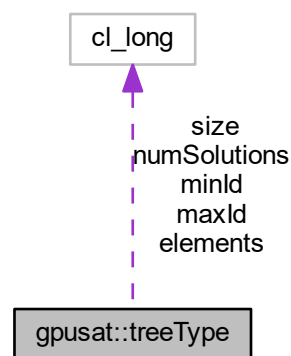
- [include/types.h](#)

## 5.15 gpusat::treeType Struct Reference

tree type for storing the models

```
#include <types.h>
```

Collaboration diagram for `gpusat::treeType`:



### Public Attributes

- `cl_long * elements = nullptr`
- `cl_long maxId = 0`
- `cl_long minId = 0`
- `cl_long numSolutions = 0`
- `cl_long size = 0`

### 5.15.1 Detailed Description

tree type for storing the models

### 5.15.2 Member Data Documentation

#### 5.15.2.1 elements

```
cl_long* gpusat::treeType::elements = nullptr
```

Referenced by `gpusat::Solver::cleanTree()`, `gpusat::Solver::combineTree()`, `main()`, `gpusat::printtreeType()`, `gpusat::solJson()`, `gpusat::solutiontable()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

#### 5.15.2.2 maxId

```
cl_long gpusat::treeType::maxId = 0
```

Referenced by `gpusat::Solver::cleanTree()`, `gpusat::Solver::combineTree()`, `main()`, `gpusat::printtreeType()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

#### 5.15.2.3 minId

```
cl_long gpusat::treeType::minId = 0
```

Referenced by `gpusat::Solver::cleanTree()`, `gpusat::Solver::combineTree()`, `main()`, `gpusat::printtreeType()`, `gpusat::solJson()`, `gpusat::solutiontable()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

#### 5.15.2.4 numSolutions

```
cl_long gpusat::treeType::numSolutions = 0
```

Referenced by `gpusat::Solver::cleanTree()`, `gpusat::Solver::combineTree()`, `gpusat::printtreeType()`, `gpusat::↔ Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

## 5.15.2.5 size

```
cl_long gpusat::treeType::size = 0
```

Referenced by `gpusat::Solver::cleanTree()`, `gpusat::Solver::combineTree()`, `gpusat::printtreeType()`, `gpusat::Solver::solveIntroduceForget()`, `gpusat::Solver::solveJoin()`, and `gpusat::Solver::solveProblem()`.

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

- `include/types.h`

## 5.16 gpusat::Visualization Class Reference

```
#include <visualization.h>
```

## Public Member Functions

- [Visualization](#) (std::string filename)
- Json::Value [getClausesJson](#) ()
- std::string [getFilename](#) ()
- Json::Value [getTdTimeline](#) ()
- Json::Value [getTreeDecJson](#) ()
- Json::StreamWriterBuilder \* [getWriterBuilder](#) ()
- bool [isEnabled](#) ()
- void [tdTimelineAppend](#) (std::vector< cl\_long > bag\_ids)
- void [tdTimelineAppend](#) (std::vector< cl\_long > bag\_ids, [TableLines](#) tablelines, std::string const toplabel="", std::string const bottomlabel="", bool transpose=true)
- void [testJson](#) ()
- void [visuClauses](#) (satformulaType \*sat)
- void [visuTreeDec](#) (treedecType \*treeDec)
- void [writeJsonFile](#) (bool append=false)
- void [writeJsonToStdout](#) (Json::StreamWriter::Factory const &factory, Json::Value const &value, Json::OStream \*sout=&std::cout)
- void [writeJsonToStdout](#) (Json::Value const &value, Json::OStream \*sout=&std::cout)

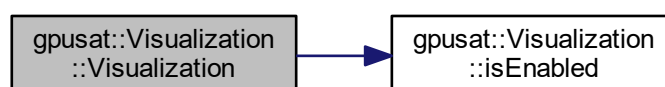
## 5.16.1 Constructor &amp; Destructor Documentation

## 5.16.1.1 Visualization()

```
gpusat::Visualization::Visualization (
    std::string filename ) [inline]
```

References `isEnabled()`.

Here is the call graph for this function:



## 5.16.2 Member Function Documentation

### 5.16.2.1 getClausesJson()

```
Json::Value gpusat::Visualization::getClausesJson ( ) [inline]
```

### 5.16.2.2 getFilename()

```
std::string gpusat::Visualization::getFilename ( ) [inline]
```

### 5.16.2.3 getTdTimeline()

```
Json::Value gpusat::Visualization::getTdTimeline ( ) [inline]
```

### 5.16.2.4 getTreeDecJson()

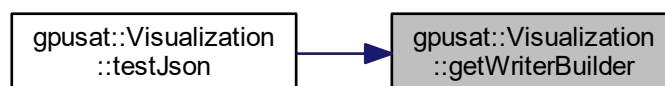
```
Json::Value gpusat::Visualization::getTreeDecJson ( ) [inline]
```

### 5.16.2.5 getWriterBuilder()

```
Json::StreamWriterBuilder * gpusat::Visualization::getWriterBuilder ( )
```

Referenced by testJson().

Here is the caller graph for this function:



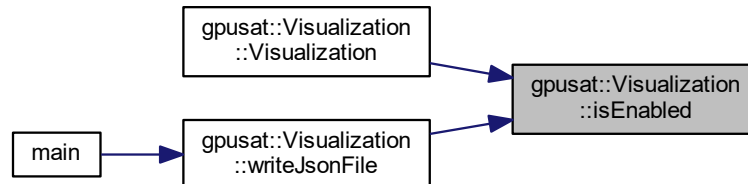


### 5.16.2.6 isEnabled()

```
bool gpusat::Visualization::isEnabled ( ) [inline]
```

Referenced by Visualization(), and writeJsonFile().

Here is the caller graph for this function:



### 5.16.2.7 tdTimelineAppend() [1/2]

```
void gpusat::Visualization::tdTimelineAppend (
    std::vector< cl_long > bag_ids )
```

Only append those bags as one array. Might even be only one bag. The behaviour with zero length is not implemented (yet).

References `LOGGER2`.

### 5.16.2.8 tdTimelineAppend() [2/2]

```
void gpusat::Visualization::tdTimelineAppend (
    std::vector< cl_long > bag_ids,
    TableLines tablelines,
    std::string const toplabel = "",
    std::string const bottomlabel = "",
    bool transpose = true )
```

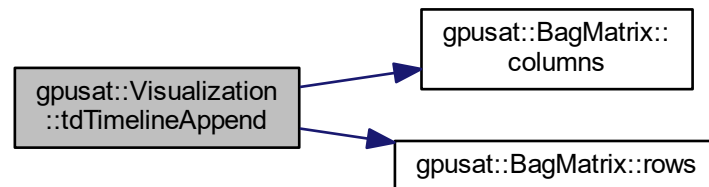
`bag_ids`: asserts not empty

- if one: one (IF) solution found
- more: join solution found tablelines: all lines for the table toplabel: string to label the whole solution  
bottomlabel: string to summarize the solution transpose: whether the tablelines are rowfirst (true) or not.

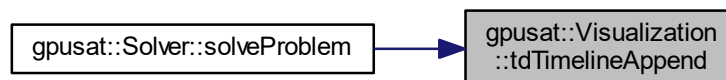
References `gpusat::BagMatrix::columns()`, `gpusat::TableLines::headline`, `LOGGER2`, `gpusat::BagMatrix::rows()`, `gpusat::TableLines::solutions`, and `gpusat::TableLines::totalSol`.

Referenced by `gpusat::Solver::solveProblem()`.

Here is the call graph for this function:



Here is the caller graph for this function:

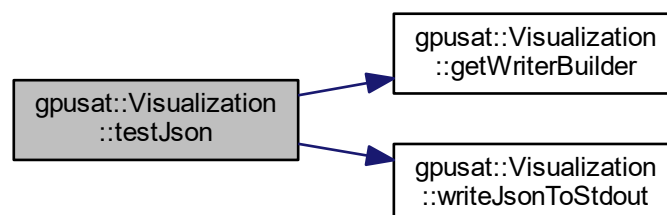


#### 5.16.2.9 testJson()

```
void gpusat::Visualization::testJson ( )
```

References `getWriterBuilder()`, and `writeJsonToStdout()`.

Here is the call graph for this function:



### 5.16.2.10 visuClauses()

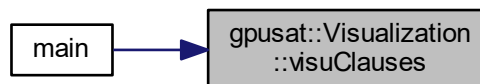
```
void gpusat::Visualization::visuClauses (
    satformulaType * sat )
```

Save the clauses into the Visualization::clausesJson. Form: List[Dict{INT: List[Int]]} Error if: sat formula is null or empty.

References gpusat::satformulaType::clauses.

Referenced by main().

Here is the caller graph for this function:



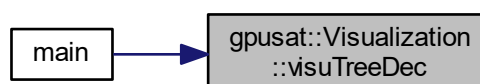
### 5.16.2.11 visuTreeDec()

```
void gpusat::Visualization::visuTreeDec (
    treedecType * treeDec )
```

References gpusat::treedecType::bags, gpusat::bagType::edges, gpusat::bagType::id, and gpusat::treedecType↔::numVars.

Referenced by main().

Here is the caller graph for this function:



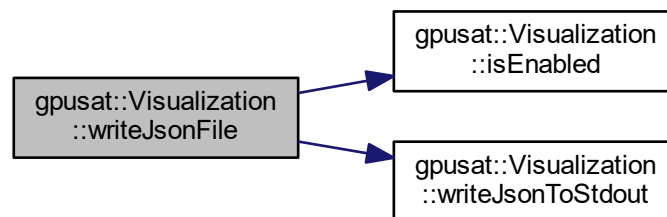
### 5.16.2.12 writeJsonFile()

```
void gpusat::Visualization::writeJsonFile (
    bool append = false )
```

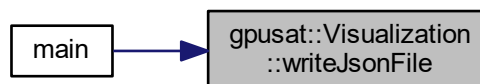
References isEnabled(), LOGGER, and writeJsonToStdout().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

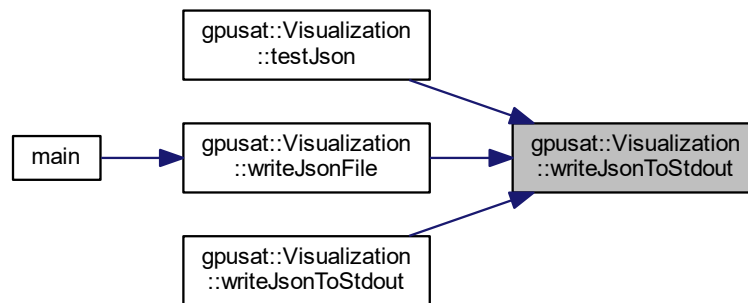


### 5.16.2.13 writeJsonToStdout() [1/2]

```
void gpusat::Visualization::writeJsonToStdout (
    Json::StreamWriter::Factory const & factory,
    Json::Value const & value,
    Json::OStream * sout = &std::cout )
```

Referenced by testJson(), writeJsonFile(), and writeJsonToStdout().

Here is the caller graph for this function:



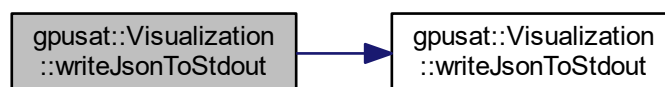
#### 5.16.2.14 writeJsonToStdout() [2/2]

```

void gpusat::Visualization::writeJsonToStdout (
    Json::Value const & value,
    Json::OStream * sout = &std::cout )
  
```

References `writeJsonToStdout()`.

Here is the call graph for this function:



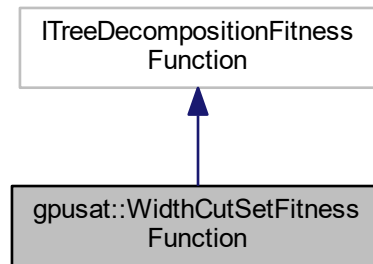
The documentation for this class was generated from the following files:

- [include/visualization.h](#)
- [src/visualization.cpp](#)

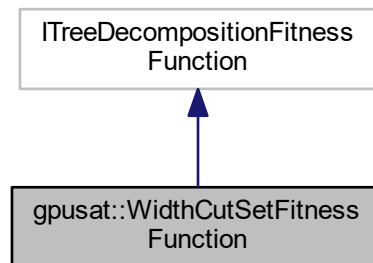
## 5.17 gpusat::WidthCutSetFitnessFunction Class Reference

```
#include <WidthCutSetFitnessFunction.h>
```

Inheritance diagram for gpusat::WidthCutSetFitnessFunction:



Collaboration diagram for gpusat::WidthCutSetFitnessFunction:



### Public Member Functions

- [WidthCutSetFitnessFunction](#) ()=default
- [~WidthCutSetFitnessFunction](#) ()=default
- [WidthCutSetFitnessFunction](#) \* [clone](#) (void) const override
- htd::FitnessEvaluation \* [fitness](#) (const htd::IMultiHypergraph &graph, const htd::ITreeDecomposition &decomposition) const override
- double [getMaxCutSetSize](#) (const htd::ITreeDecomposition &decomposition) const

#### 5.17.1 Detailed Description

Fitness function that first minimizes the width and then the cut set size.

## 5.17.2 Constructor & Destructor Documentation

### 5.17.2.1 WidthCutSetFitnessFunction()

```
gpusat::WidthCutSetFitnessFunction::WidthCutSetFitnessFunction ( ) [default]
```

Referenced by clone().

Here is the caller graph for this function:



### 5.17.2.2 ~WidthCutSetFitnessFunction()

```
gpusat::WidthCutSetFitnessFunction::~~WidthCutSetFitnessFunction ( ) [default]
```

## 5.17.3 Member Function Documentation

### 5.17.3.1 clone()

```
WidthCutSetFitnessFunction * gpusat::WidthCutSetFitnessFunction::clone (
    void ) const [override]
```

References WidthCutSetFitnessFunction().

Here is the call graph for this function:



### 5.17.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::WidthCutSetFitnessFunction::fitness (
    const htd::IMultiHypergraph & graph,
    const htd::ITreeDecomposition & decomposition ) const [override]
```

References getMaxCutSetSize().

Here is the call graph for this function:

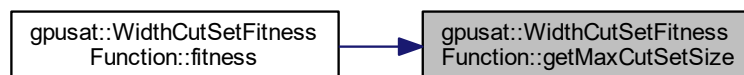


### 5.17.3.3 getMaxCutSetSize()

```
double gpusat::WidthCutSetFitnessFunction::getMaxCutSetSize (
    const htd::ITreeDecomposition & decomposition ) const
```

Referenced by `fitness()`.

Here is the caller graph for this function:



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

- [include/FitnessFunctions/WidthCutSetFitnessFunction.h](#)
- [src/FitnessFunctions/WidthCutSetFitnessFunction.cpp](#)



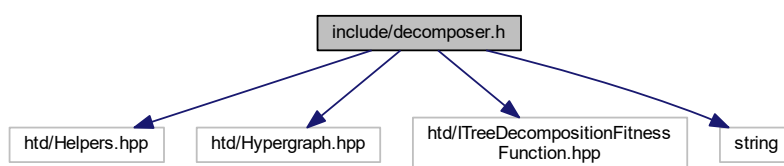
## Chapter 6

# File Documentation

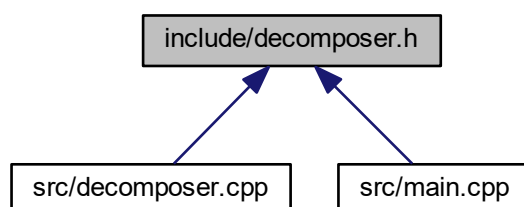
### 6.1 include/decomposer.h File Reference

```
#include <htd/Helpers.hpp>
#include <htd/Hypergraph.hpp>
#include <htd/ITreeDecompositionFitnessFunction.hpp>
#include <string>
```

Include dependency graph for decomposer.h:



This graph shows which files directly or indirectly include this file:



### Classes

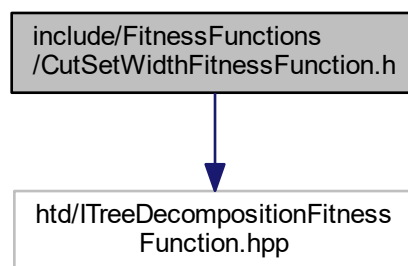
- class `gpusat::Decomposer`

## Namespaces

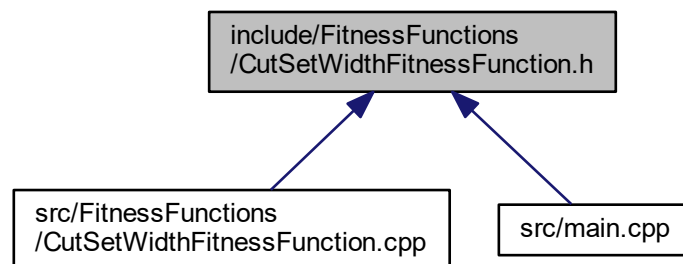
- [gpusat](#)

## 6.2 include/FitnessFunctions/CutSetWidthFitnessFunction.h File Reference

`#include <htd/ITreeDecompositionFitnessFunction.hpp>`  
Include dependency graph for CutSetWidthFitnessFunction.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gpusat::CutSetWidthFitnessFunction](#)

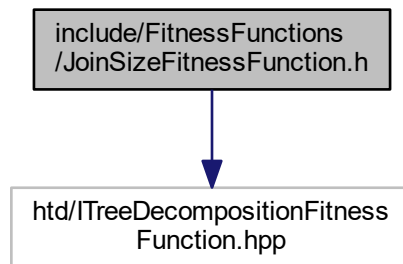
## Namespaces

- [gpusat](#)

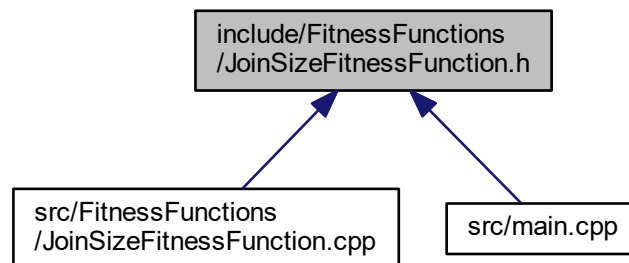
## 6.3 include/FitnessFunctions/JoinSizeFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for JoinSizeFitnessFunction.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gpusat::JoinSizeFitnessFunction](#)

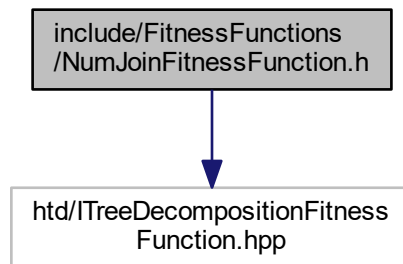
### Namespaces

- [gpusat](#)

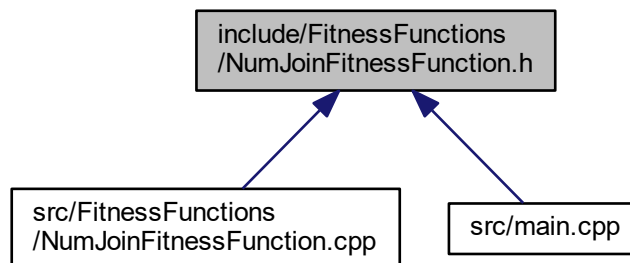
## 6.4 include/FitnessFunctions/NumJoinFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for NumJoinFitnessFunction.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gpusat::NumJoinFitnessFunction](#)

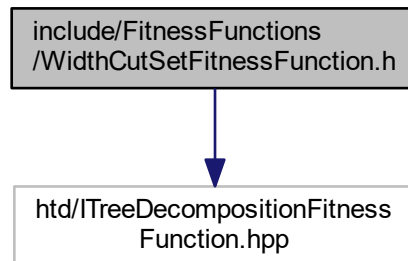
### Namespaces

- [gpusat](#)

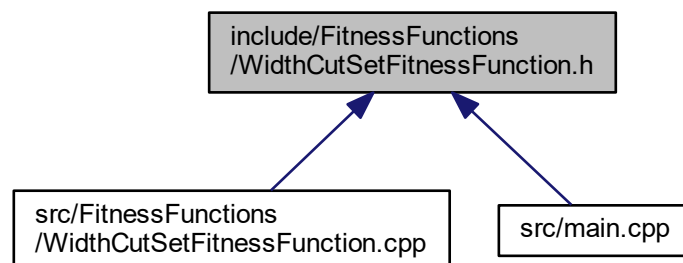
## 6.5 include/FitnessFunctions/WidthCutSetFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for WidthCutSetFitnessFunction.h:



This graph shows which files directly or indirectly include this file:



### Classes

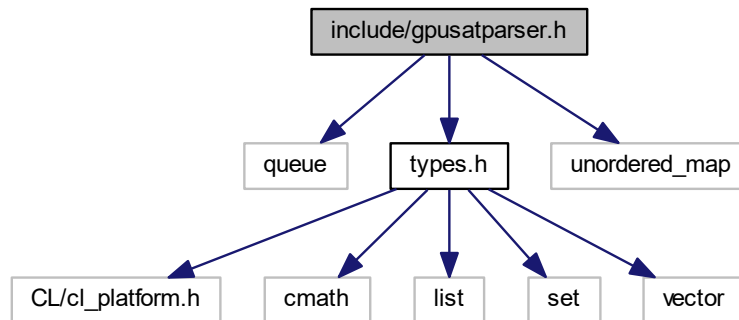
- class [gpusat::WidthCutSetFitnessFunction](#)

### Namespaces

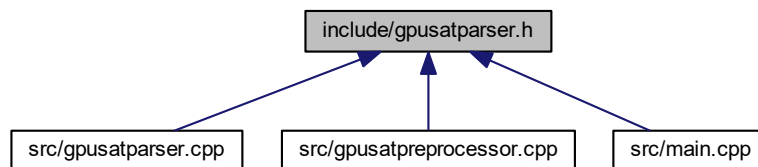
- [gpusat](#)

## 6.6 include/gpusatparser.h File Reference

```
#include <queue>
#include <types.h>
#include <unordered_map>
Include dependency graph for gpusatparser.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class `gpusat::CNFParser`
- class `gpusat::TDParser`

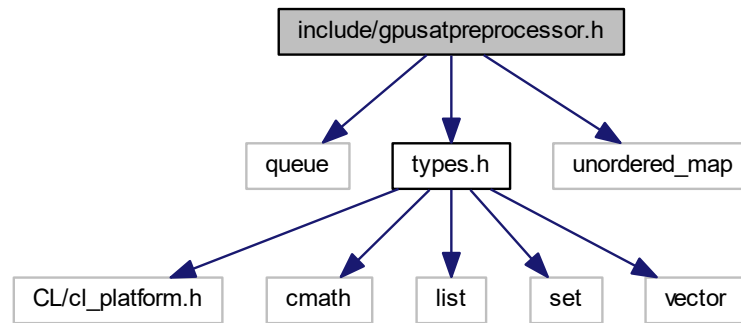
### Namespaces

- `gpusat`

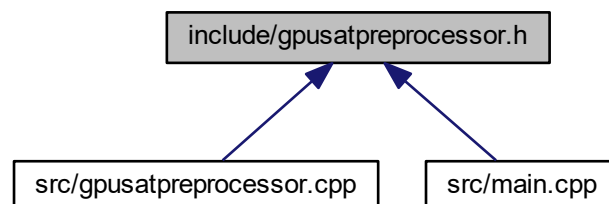
## 6.7 include/gpusatpreprocessor.h File Reference

```
#include <queue>
#include <types.h>
#include <unordered_map>
```

Include dependency graph for gpusatpreprocessor.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gpusat::Preprocessor](#)

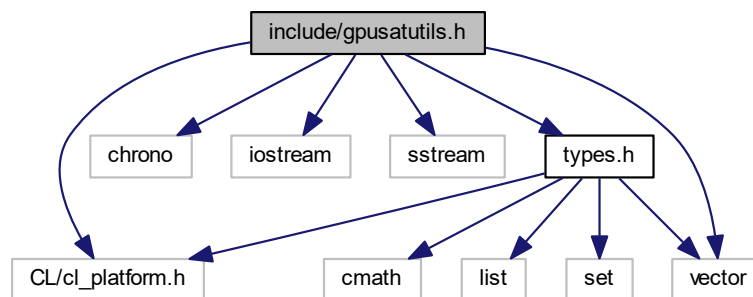
### Namespaces

- [gpusat](#)

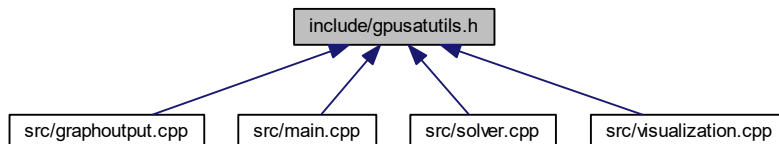
## 6.8 include/gpusatutils.h File Reference

```
#include <CL/cl_platform.h>
#include <chrono>
#include <iostream>
#include <sstream>
#include <types.h>
#include <vector>
```

Include dependency graph for gpusatutils.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [gpusat](#)

## Functions

- `cl_double` [gpusat::getCount](#) (`cl_long` id, `cl_long` \*tree, `cl_long` numVars)
- `long long int` [gpusat::getTime](#) ()
- `std::ostream &` [gpusat::operator<<](#) (`std::ostream &os`, `const dataStructure ds`)
- `std::ostream &` [gpusat::operator<<](#) (`std::ostream &os`, `const std::vector< bagType * > vec`)
- `std::ostream &` [gpusat::operator<<](#) (`std::ostream &os`, `const std::vector< cl_long > vec`)
- `void` [gpusat::printbagType](#) (`bagType *bag`, `std::ostream &stream`, `int depth=0`)

*print information for a bag in the tree decomposition*

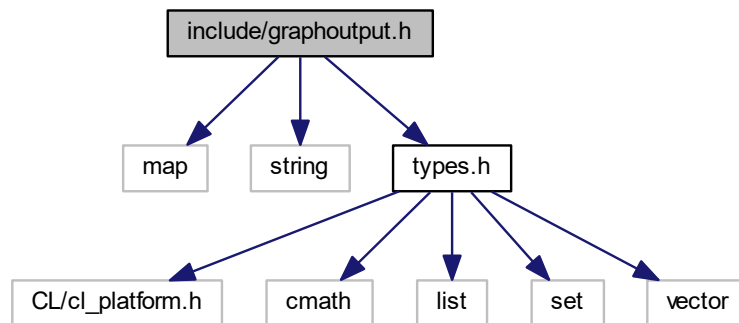


- void [gpusat::printtreeDecType](#) (treeDecType \*dec, std::ostream &stream)  
*print a tree decomposition*
- void [gpusat::printtreeType](#) (treeType \*tree, std::ostream &stream, size\_t size, int depth=0)
- std::string [gpusat::solutiontable](#) (bagType node, dataStructure solutionType)  
*Generate a formatted stringoutput for a solved node with solutions.*

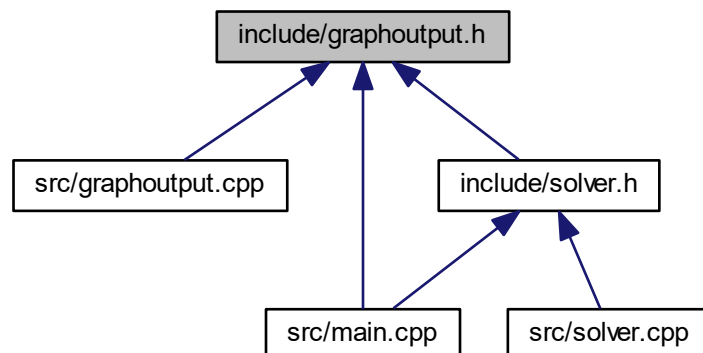
## 6.9 include/graphoutput.h File Reference

```
#include <map>
#include <string>
#include <types.h>
```

Include dependency graph for graphoutput.h:



This graph shows which files directly or indirectly include this file:



## Classes

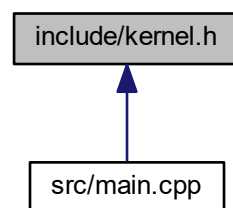
- class [gpusat::Graphoutput](#)

## Namespaces

- [gpusat](#)

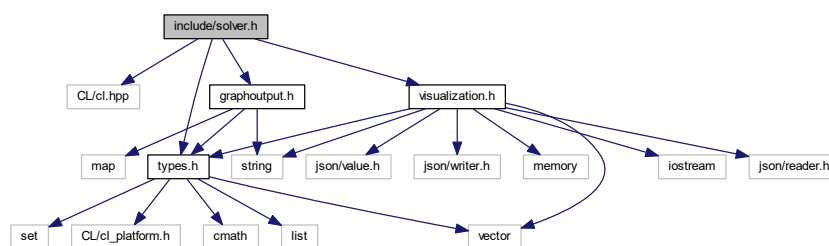
## 6.10 include/kernel.h File Reference

This graph shows which files directly or indirectly include this file:

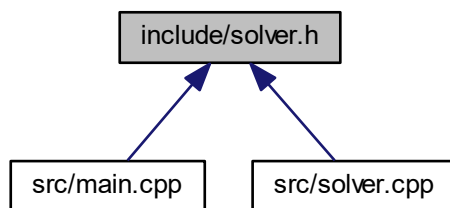


## 6.11 include/solver.h File Reference

```
#include <CL/cl.hpp>
#include <graphoutput.h>
#include <types.h>
#include <visualization.h>
Include dependency graph for solver.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

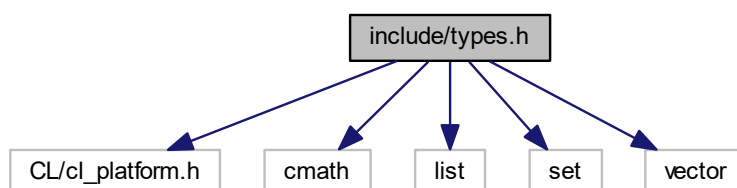
- class [gpusat::Solver](#)

## Namespaces

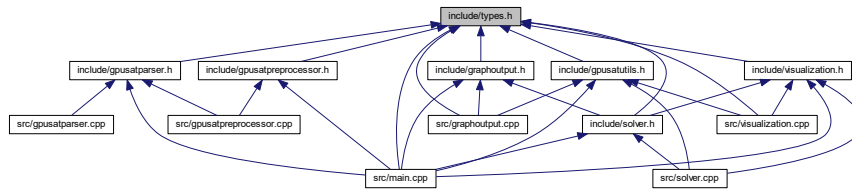
- [gpusat](#)

## 6.12 include/types.h File Reference

```
#include <CL/cl_platform.h>
#include <cmath>
#include <list>
#include <set>
#include <vector>
Include dependency graph for types.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- struct [gpusat::bagType](#)  
*type for a bag in the tree decomposition*
- struct [gpusat::satformulaType](#)  
*type for saving the sat formula*
- struct [gpusat::treedecType](#)  
*type for saving a tree decomposition*
- struct [gpusat::treeType](#)  
*tree type for storing the models*

## Namespaces

- [gpusat](#)

## Enumerations

- enum [gpusat::dataStructure](#) { [gpusat::dataStructure::ARRAY](#), [gpusat::dataStructure::TREE](#) }
- enum [gpusat::nodeType](#) { [gpusat::nodeType::JOIN](#), [gpusat::nodeType::INTRODUCEFORGET](#) }

## Functions

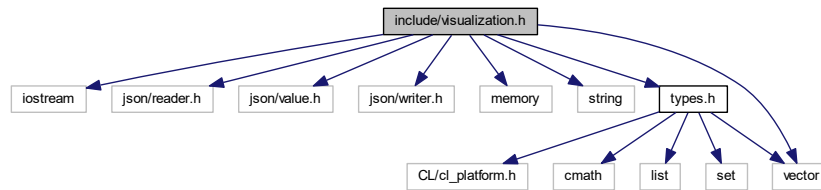
- bool [gpusat::compTreedType](#) (const bagType \*a, const bagType \*b)
- bool [gpusat::compVars](#) (const cl\_long &a, const cl\_long &b)

## 6.13 include/visualization.h File Reference

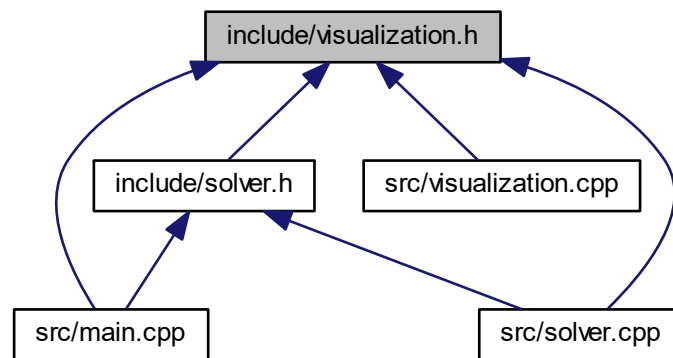
```
#include <iostream>
#include <json/reader.h>
#include <json/value.h>
#include <json/writer.h>
#include <memory>
#include <string>
#include <types.h>
```

```
#include <vector>
```

Include dependency graph for visualization.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gpusat::BagMatrix`
- struct `gpusat::TableLines`
- class `gpusat::Visualization`

## Namespaces

- `gpusat`

## Functions

- TableLines `gpusat::solJson` (bagType node, dataStructure solutionType)

## 6.14 src/decomposer.cpp File Reference

```
#include <decomposer.h>
#include <htd/BucketEliminationTreeDecompositionAlgorithm.hpp>
#include <htd/GraphPreprocessor.hpp>
#include <htd/ITreeDecompositionAlgorithm.hpp>
#include <htd/IterativeImprovementTreeDecompositionAlgorithm.hpp>
#include <htd/TreeDecompositionOptimizationOperation.hpp>
#include <htd_io/TdFormatExporter.hpp>
#include <sstream>
```

Include dependency graph for decomposer.cpp:



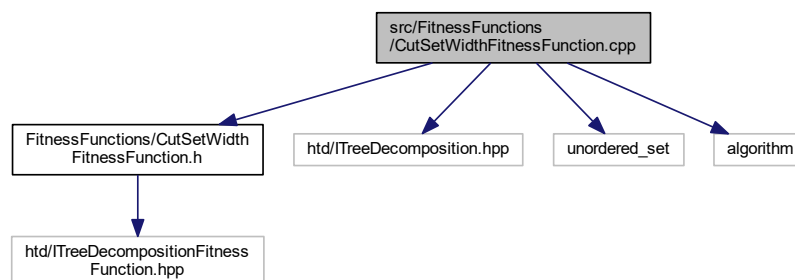
## Namespaces

- [gpusat](#)

## 6.15 src/FitnessFunctions/CutSetWidthFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/CutSetWidthFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for CutSetWidthFitnessFunction.cpp:



## Namespaces

- [gpusat](#)

## 6.16 src/FitnessFunctions/JoinSizeFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/JoinSizeFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for JoinSizeFitnessFunction.cpp:



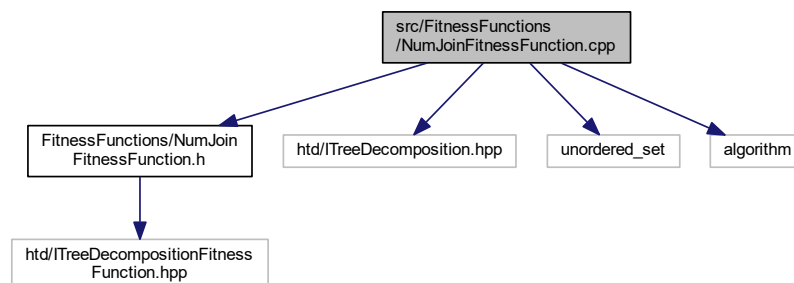
### Namespaces

- [gpusat](#)

## 6.17 src/FitnessFunctions/NumJoinFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/NumJoinFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for NumJoinFitnessFunction.cpp:



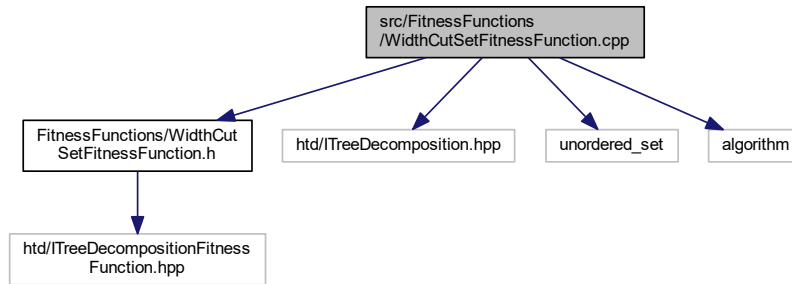
### Namespaces

- [gpusat](#)

## 6.18 src/FitnessFunctions/WidthCutSetFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/WidthCutSetFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for WidthCutSetFitnessFunction.cpp:



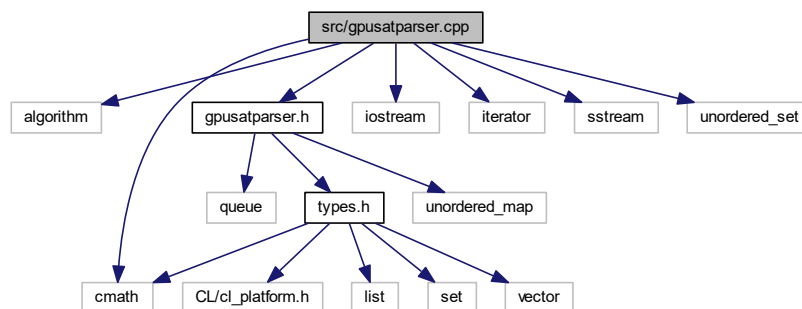
### Namespaces

- [gpusat](#)

## 6.19 src/gpusatparser.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <gpusatparser.h>
#include <iostream>
#include <iterator>
#include <sstream>
#include <unordered_set>
```

Include dependency graph for gpusatparser.cpp:





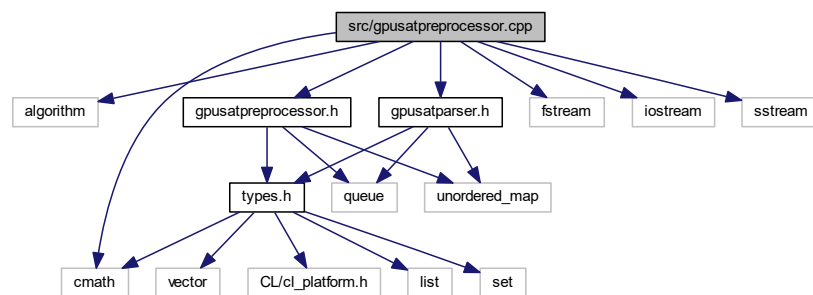
## Namespaces

- [gpusat](#)

## 6.20 src/gpusatpreprocessor.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <fstream>
#include <gpusatparser.h>
#include <gpusatpreprocessor.h>
#include <iostream>
#include <sstream>
```

Include dependency graph for gpusatpreprocessor.cpp:



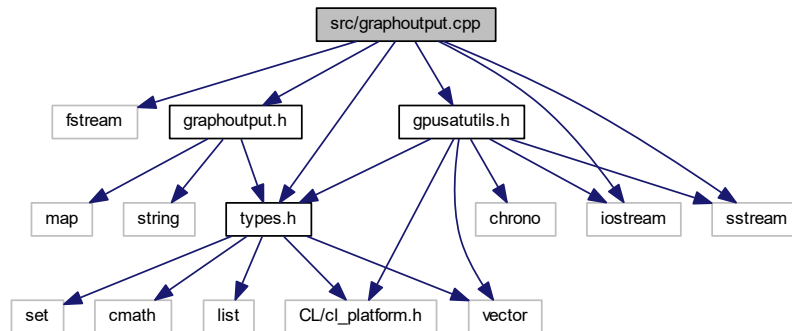
## Namespaces

- [gpusat](#)

## 6.21 src/graphoutput.cpp File Reference

```
#include <fstream>
#include <gpusatutils.h>
#include <graphoutput.h>
#include <iostream>
#include <sstream>
```

Include dependency graph for graphoutput.cpp:



- gpusat

```
#include <fstream>
#include <iostream>
#include <sstream>
#include <CLI11.hpp>
#include <FitnessFunctions/CutSetWidthFitnessFunction.h>
#include <FitnessFunctions/JoinSizeFitnessFunction.h>
#include <FitnessFunctions/NumJoinFitnessFunction.h>
#include <FitnessFunctions/WidthCutSetFitnessFunction.h>
#include <boost/multiprecision/cpp_bin_float.hpp>
#include <chrono>
#include <decomposer.h>
#include <gpusatparser.h>
#include <gpusatpreprocessor.h>
#include <gpusatutils.h>
#include <graphoutput.h>
#include <math.h>
#include <numeric>
#include <solver.h>
#include <sys/stat.h>
#include <types.h>
#include <visualization.h>
#include <kernel.h>
```

[illegible]

## Macros

- `#define __CL_ENABLE_EXCEPTIONS`
- `#define NELEMS(x) (sizeof(x) / sizeof((x)[0]))`

## Functions

- void `buildKernel` (cl::Context &context, std::vector< cl::Device > &devices, cl::CommandQueue &queue, cl::Program &program, cl\_long &memorySize, cl\_long &maxMemoryBuffer, bool nvidia, bool amd, bool cpu, long &combineWidth)
- void `device_query` ()
- int `main` (int argc, char \*argv[])
- void `PrintDeviceInfo` (cl\_device\_id device)

## Variables

- const char \*const `attributeNames` []
- const cl\_platform\_info `attributeTypes` [5]
- std::string `kernelStr`

### 6.22.1 Macro Definition Documentation

#### 6.22.1.1 \_\_CL\_ENABLE\_EXCEPTIONS

```
#define __CL_ENABLE_EXCEPTIONS
```

#### 6.22.1.2 NELEMS

```
#define NELEMS(  
    x ) (sizeof(x) / sizeof((x)[0]))
```

### 6.22.2 Function Documentation

### 6.22.2.1 buildKernel()

```
void buildKernel (
    cl::Context & context,
    std::vector< cl::Device > & devices,
    cl::CommandQueue & queue,
    cl::Program & program,
    cl_long & memorySize,
    cl_long & maxMemoryBuffer,
    bool nvidia,
    bool amd,
    bool cpu,
    long & combineWidth )
```

References kernelStr.

Referenced by main().

Here is the caller graph for this function:



### 6.22.2.2 device\_query()

```
void device_query ( )
```

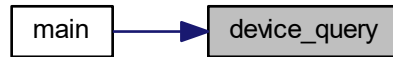
References attributeNames, attributeTypes, NELEMS, and PrintDeviceInfo().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.22.2.3 main()

```

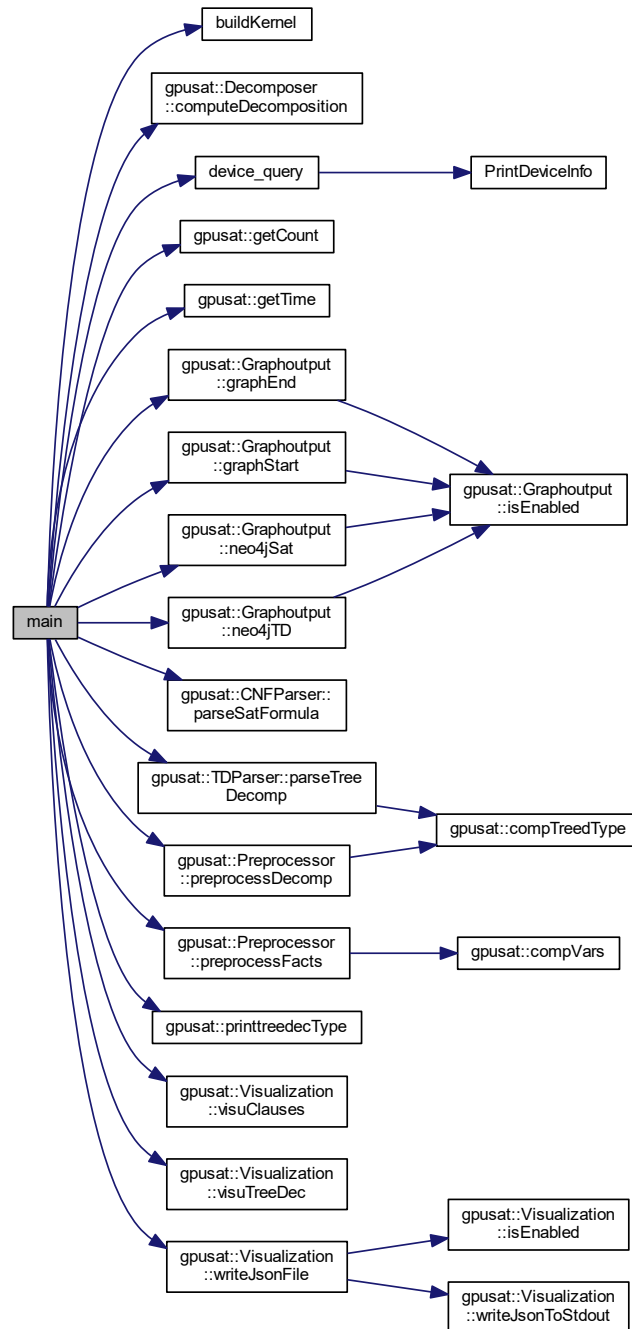
int main (
    int argc,
    char * argv[] )

```

solution visualisation

References `gpusat::ARRAY`, `gpusat::treedecType::bags`, `buildKernel()`, `gpusat::Decomposer::computeDecomposition()`, `gpusat::TDParser::defaultWeight`, `device_query()`, `gpusat::treeType::elements`, `gpusat::getCount()`, `gpusat::getTime()`, `gpusat::Graphoutput::graphEnd()`, `gpusat::Graphoutput::graphStart()`, `gpusat::INTRODUCEFORGET`, `kernelStr`, `gpusat::treeType::maxId`, `gpusat::Solver::maxTableSize`, `gpusat::treeType::minId`, `gpusat::Graphoutput::neo4jSat()`, `gpusat::Graphoutput::neo4jTD()`, `gpusat::Solver::numIntroduceForget`, `gpusat::Solver::numJoin`, `gpusat::CNFParser::parseSatFormula()`, `gpusat::TDParser::parseTreeDecomp()`, `gpusat::Preprocessor::preprocessDecomp()`, `gpusat::Preprocessor::preprocessFacts()`, `gpusat::printtreedecType()`, `gpusat::TREE`, `gpusat::satformulaType::unsat`, `gpusat::bagType::variables`, `gpusat::Visualization::visuClauses()`, `gpusat::Visualization::visuTreeDec()`, `gpusat::treedecType::width`, and `gpusat::Visualization::writeJsonFile()`.

Here is the call graph for this function:



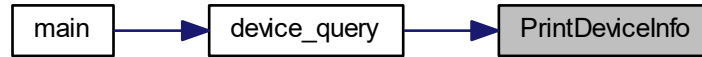
#### 6.22.2.4 PrintDeviceInfo()

```

void PrintDeviceInfo (
    cl_device_id device )
  
```

Referenced by device\_query().

Here is the caller graph for this function:



### 6.22.3 Variable Documentation

#### 6.22.3.1 attributeNames

```
const char* const attributeNames[]
```

**Initial value:**

```
= {
    "CL_PLATFORM_NAME",
    "CL_PLATFORM_VENDOR",
    "CL_PLATFORM_VERSION",
    "CL_PLATFORM_PROFILE",
    "CL_PLATFORM_EXTENSIONS" }
```

Referenced by device\_query().

#### 6.22.3.2 attributeTypes

```
const cl_platform_info attributeTypes[5]
```

**Initial value:**

```
= {
    CL_PLATFORM_NAME,
    CL_PLATFORM_VENDOR,
    CL_PLATFORM_VERSION,
    CL_PLATFORM_PROFILE,
    CL_PLATFORM_EXTENSIONS }
```

Referenced by device\_query().

#### 6.22.3.3 kernelStr

```
std::string kernelStr
```

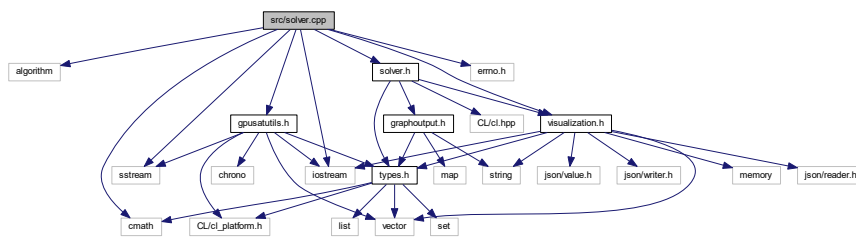
**Initial value:**

```
=
using namespace gpustat
```

Referenced by buildKernel(), and main().

## 6.23 src/solver.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <errno.h>
#include <gpusatutils.h>
#include <iostream>
#include <solver.h>
#include <sstream>
#include <visualization.h>
Include dependency graph for solver.cpp:
```

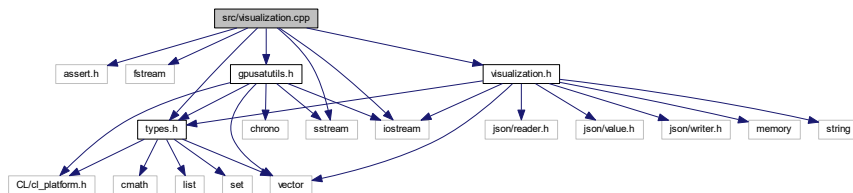


### Namespaces

- [gpusat](#)

## 6.24 src/visualization.cpp File Reference

```
#include <assert.h>
#include <fstream>
#include <gpusatutils.h>
#include <iostream>
#include <sstream>
#include <types.h>
#include <visualization.h>
Include dependency graph for visualization.cpp:
```



### Namespaces

- [gpusat](#)



## Macros

- `#define LOGGER(x)` (`std::cout << "LOGGING: " << x << std::endl`)
- `#define LOGGER2(x)`

## Functions

- TableLines [gpusat::solJson](#) (bagType node, dataStructure solutionType)

### 6.24.1 Macro Definition Documentation

#### 6.24.1.1 [LOGGER](#)

```
#define LOGGER(  
    x ) (std::cout << "LOGGING: " << x << std::endl)
```

#### 6.24.1.2 [LOGGER2](#)

```
#define LOGGER2(  
    x )
```



# Index

- `__CL_ENABLE_EXCEPTIONS`
  - `main.cpp`, 91
- `~CutSetWidthFitnessFunction`
  - `gpusat::CutSetWidthFitnessFunction`, 24
- `~JoinSizeFitnessFunction`
  - `gpusat::JoinSizeFitnessFunction`, 37
- `~NumJoinFitnessFunction`
  - `gpusat::NumJoinFitnessFunction`, 40
- `~WidthCutSetFitnessFunction`
  - `gpusat::WidthCutSetFitnessFunction`, 71
- ARRAY
  - `gpusat`, 8
- attributeNames
  - `main.cpp`, 95
- attributeTypes
  - `main.cpp`, 95
- BagMatrix
  - `gpusat::BagMatrix`, 17
- bags
  - `gpusat::bagType`, 20
  - `gpusat::treedecType`, 60
- baseIdJoin
  - `gpusat::Graphoutput`, 35
- baseIdSol
  - `gpusat::Graphoutput`, 35
- buildKernel
  - `main.cpp`, 91
- clauses
  - `gpusat::satformulaType`, 44
- cleanTree
  - `gpusat::Solver`, 47
- clone
  - `gpusat::CutSetWidthFitnessFunction`, 25
  - `gpusat::JoinSizeFitnessFunction`, 38
  - `gpusat::NumJoinFitnessFunction`, 40
  - `gpusat::WidthCutSetFitnessFunction`, 71
- CNFParser
  - `gpusat::CNFParser`, 22
- columns
  - `gpusat::BagMatrix`, 18
- combineTree
  - `gpusat::Solver`, 48
- compTreedType
  - `gpusat`, 8
- computeDecomposition
  - `gpusat::Decomposer`, 26
- compVars
  - `gpusat`, 9
- connectbag
  - `gpusat::Graphoutput`, 35
- context
  - `gpusat::Solver`, 53
- correction
  - `gpusat::bagType`, 20
- CutSetWidthFitnessFunction
  - `gpusat::CutSetWidthFitnessFunction`, 24
- dataStructure
  - `gpusat`, 8
- defaultWeight
  - `gpusat::TDParser`, 59
- device\_query
  - `main.cpp`, 92
- dualedge
  - `gpusat::Graphoutput`, 35
- edges
  - `gpusat::bagType`, 20
- elements
  - `gpusat::treeType`, 62
- exponent
  - `gpusat::bagType`, 20
- facts
  - `gpusat::satformulaType`, 44
- fitness
  - `gpusat::CutSetWidthFitnessFunction`, 25
  - `gpusat::JoinSizeFitnessFunction`, 38
  - `gpusat::NumJoinFitnessFunction`, 40
  - `gpusat::WidthCutSetFitnessFunction`, 71
- getClausesJson
  - `gpusat::Visualization`, 64
- getCount
  - `gpusat`, 10
- getFilename
  - `gpusat::Graphoutput`, 28
  - `gpusat::Visualization`, 64
- getMaxCutSetSize
  - `gpusat::CutSetWidthFitnessFunction`, 25
  - `gpusat::WidthCutSetFitnessFunction`, 72
- getTdTimeline
  - `gpusat::Visualization`, 64
- getTime
  - `gpusat`, 11
- getTreeDecJson
  - `gpusat::Visualization`, 64

- getWriterBuilder
  - gpusat::Visualization, 64
- gpusat, 7
  - ARRAY, 8
  - compTreedType, 8
  - compVars, 9
  - dataStructure, 8
  - getCount, 10
  - getTime, 11
  - INTRODUCEFORGET, 8
  - JOIN, 8
  - nodeType, 8
  - operator<<, 11, 12
  - printbagType, 12
  - printtreedecType, 12
  - printtreeType, 13
  - solJson, 13
  - solutiontable, 14
  - TREE, 8
- gpusat::BagMatrix, 17
  - BagMatrix, 17
  - columns, 18
  - operator(), 18
  - rows, 18
- gpusat::bagType, 19
  - bags, 20
  - correction, 20
  - edges, 20
  - exponent, 20
  - id, 21
  - maxSize, 21
  - solution, 21
  - variables, 21
- gpusat::CNFParser, 22
  - CNFParser, 22
  - parseSatFormula, 22
- gpusat::CutSetWidthFitnessFunction, 23
  - ~CutSetWidthFitnessFunction, 24
  - clone, 25
  - CutSetWidthFitnessFunction, 24
  - fitness, 25
  - getMaxCutSetSize, 25
- gpusat::Decomposer, 26
  - computeDecomposition, 26
- gpusat::Graphoutput, 27
  - baselIdJoin, 35
  - baselIdSol, 35
  - connectbag, 35
  - dualedge, 35
  - getFilename, 28
  - graphEdgeSet, 28
  - graphEnd, 29
  - Graphoutput, 28
  - graphStart, 30
  - inbag, 35
  - incidenceedge, 36
  - isEnabled, 30
  - neo4jSat, 31
  - neo4jTD, 32
  - nodeBag, 33
  - nodeJoin, 34
  - primaledge, 36
- gpusat::JoinSizeFitnessFunction, 36
  - ~JoinSizeFitnessFunction, 37
  - clone, 38
  - fitness, 38
  - JoinSizeFitnessFunction, 37
- gpusat::NumJoinFitnessFunction, 39
  - ~NumJoinFitnessFunction, 40
  - clone, 40
  - fitness, 40
  - NumJoinFitnessFunction, 40
- gpusat::Preprocessor, 41
  - preprocessDecomp, 41
  - preprocessFacts, 42
- gpusat::satformulaType, 43
  - clauses, 44
  - facts, 44
  - numVars, 44
  - numWeights, 44
  - unsat, 45
  - variableWeights, 45
- gpusat::Solver, 45
  - cleanTree, 47
  - combineTree, 48
  - context, 53
  - graphoutput, 54
  - isSat, 54
  - maxBag, 54
  - maxMemoryBuffer, 54
  - maxTableSize, 54
  - memorySize, 54
  - numIntroduceForget, 55
  - numJoin, 55
  - program, 55
  - queue, 55
  - solutionType, 55
  - solveIntroduceForget, 48
  - solveJoin, 50
  - solveProblem, 52
  - Solver, 46
  - verbose, 55
  - visualization, 56
- gpusat::TableLines, 56
  - headline, 57
  - solutions, 57
  - totalSol, 57
- gpusat::TDParse, 57
  - defaultWeight, 59
  - parseTreeDecomp, 58
  - TDParse, 58
- gpusat::treedecType, 59
  - bags, 60
  - numb, 60
  - numVars, 60
  - width, 61

- gpusat::treeType, 61
  - elements, 62
  - maxId, 62
  - minId, 62
  - numSolutions, 62
  - size, 62
- gpusat::Visualization, 63
  - getClausesJson, 64
  - getFilename, 64
  - getTdTimeline, 64
  - getTreeDecJson, 64
  - getWriterBuilder, 64
  - isEnabled, 64
  - tdTimelineAppend, 65
  - testJson, 66
  - Visualization, 63
  - visuClauses, 67
  - visuTreeDec, 67
  - writeJsonFile, 67
  - writeJsonToStdout, 68, 69
- gpusat::WidthCutSetFitnessFunction, 70
  - ~WidthCutSetFitnessFunction, 71
  - clone, 71
  - fitness, 71
  - getMaxCutSetSize, 72
  - WidthCutSetFitnessFunction, 71
- graphEdgeSet
  - gpusat::Graphoutput, 28
- graphEnd
  - gpusat::Graphoutput, 29
- Graphoutput
  - gpusat::Graphoutput, 28
- graphoutput
  - gpusat::Solver, 54
- graphStart
  - gpusat::Graphoutput, 30
- headline
  - gpusat::TableLines, 57
- id
  - gpusat::bagType, 21
- inbag
  - gpusat::Graphoutput, 35
- incidenceedge
  - gpusat::Graphoutput, 36
- include/decomposer.h, 73
- include/FitnessFunctions/CutSetWidthFitnessFunction.h, 74
- include/FitnessFunctions/JoinSizeFitnessFunction.h, 75
- include/FitnessFunctions/NumJoinFitnessFunction.h, 76
- include/FitnessFunctions/WidthCutSetFitnessFunction.h, 77
- include/gpusatparser.h, 78
- include/gpusatpreprocessor.h, 79
- include/gpusatutils.h, 80
- include/graphoutput.h, 81
- include/kernel.h, 82
- include/solver.h, 82
- include/types.h, 83
- include/visualization.h, 84
- INTRODUCEFORGET
  - gpusat, 8
- isEnabled
  - gpusat::Graphoutput, 30
  - gpusat::Visualization, 64
- isSat
  - gpusat::Solver, 54
- JOIN
  - gpusat, 8
- JoinSizeFitnessFunction
  - gpusat::JoinSizeFitnessFunction, 37
- kernelStr
  - main.cpp, 95
- LOGGER
  - visualization.cpp, 97
- LOGGER2
  - visualization.cpp, 97
- main
  - main.cpp, 93
- main.cpp
  - \_\_CL\_ENABLE\_EXCEPTIONS, 91
  - attributeNames, 95
  - attributeTypes, 95
  - buildKernel, 91
  - device\_query, 92
  - kernelStr, 95
  - main, 93
  - NELEMS, 91
  - PrintDeviceInfo, 94
- maxBag
  - gpusat::Solver, 54
- maxId
  - gpusat::treeType, 62
- maxMemoryBuffer
  - gpusat::Solver, 54
- maxSize
  - gpusat::bagType, 21
- maxTableSize
  - gpusat::Solver, 54
- memorySize
  - gpusat::Solver, 54
- minId
  - gpusat::treeType, 62
- NELEMS
  - main.cpp, 91
- neo4jSat
  - gpusat::Graphoutput, 31
- neo4jTD
  - gpusat::Graphoutput, 32
- nodeBag
  - gpusat::Graphoutput, 33
- nodeJoin

- gpusat::Graphoutput, 34
- nodeType
  - gpusat, 8
- numb
  - gpusat::treedecType, 60
- numIntroduceForget
  - gpusat::Solver, 55
- numJoin
  - gpusat::Solver, 55
- NumJoinFitnessFunction
  - gpusat::NumJoinFitnessFunction, 40
- numSolutions
  - gpusat::treeType, 62
- numVars
  - gpusat::satformulaType, 44
  - gpusat::treedecType, 60
- numWeights
  - gpusat::satformulaType, 44
- operator<<
  - gpusat, 11, 12
- operator()
  - gpusat::BagMatrix, 18
- parseSatFormula
  - gpusat::CNFParser, 22
- parseTreeDecomp
  - gpusat::TDParser, 58
- preprocessDecomp
  - gpusat::Preprocessor, 41
- preprocessFacts
  - gpusat::Preprocessor, 42
- primaledge
  - gpusat::Graphoutput, 36
- printbagType
  - gpusat, 12
- PrintDeviceInfo
  - main.cpp, 94
- printtreedecType
  - gpusat, 12
- printtreeType
  - gpusat, 13
- program
  - gpusat::Solver, 55
- queue
  - gpusat::Solver, 55
- rows
  - gpusat::BagMatrix, 18
- size
  - gpusat::treeType, 62
- solJson
  - gpusat, 13
- solution
  - gpusat::bagType, 21
- solutions
  - gpusat::TableLines, 57
- solutiontable
  - gpusat, 14
- solutionType
  - gpusat::Solver, 55
- solveIntroduceForget
  - gpusat::Solver, 48
- solveJoin
  - gpusat::Solver, 50
- solveProblem
  - gpusat::Solver, 52
- Solver
  - gpusat::Solver, 46
- src/decomposer.cpp, 86
- src/FitnessFunctions/CutSetWidthFitnessFunction.cpp, 86
- src/FitnessFunctions/JoinSizeFitnessFunction.cpp, 87
- src/FitnessFunctions/NumJoinFitnessFunction.cpp, 87
- src/FitnessFunctions/WidthCutSetFitnessFunction.cpp, 88
- src/gpusatparser.cpp, 88
- src/gpusatpreprocessor.cpp, 89
- src/graphoutput.cpp, 89
- src/main.cpp, 90
- src/solver.cpp, 96
- src/visualization.cpp, 96
- TDParser
  - gpusat::TDParser, 58
- tdTimelineAppend
  - gpusat::Visualization, 65
- testJson
  - gpusat::Visualization, 66
- totalSol
  - gpusat::TableLines, 57
- TREE
  - gpusat, 8
- unsat
  - gpusat::satformulaType, 45
- variables
  - gpusat::bagType, 21
- variableWeights
  - gpusat::satformulaType, 45
- verbose
  - gpusat::Solver, 55
- Visualization
  - gpusat::Visualization, 63
- visualization
  - gpusat::Solver, 56
- visualization.cpp
  - LOGGER, 97
  - LOGGER2, 97
- visuClauses
  - gpusat::Visualization, 67
- visuTreeDec
  - gpusat::Visualization, 67
- width

---

- gpusat::treedecType, [61](#)
- WidthCutSetFitnessFunction
  - gpusat::WidthCutSetFitnessFunction, [71](#)
- writeJsonFile
  - gpusat::Visualization, [67](#)
- writeJsonToStdout
  - gpusat::Visualization, [68](#), [69](#)