

opt\_jr\_doc

Generated by Doxygen 1.8.5

Sat Dec 9 2017 12:07:44



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Application Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	6
3.1.1.1	Application . . . . .	6
3.1.2	Member Function Documentation . . . . .	6
3.1.2.1	ObjFunctionComponent . . . . .	7
3.1.3	Member Data Documentation . . . . .	7
3.1.3.1	alpha . . . . .	7
3.1.3.2	app_id . . . . .	7
3.1.3.3	baseFO . . . . .	7
3.1.3.4	beta . . . . .	7
3.1.3.5	bound . . . . .	7
3.1.3.6	boundIterations . . . . .	7
3.1.3.7	chi_0 . . . . .	7
3.1.3.8	chi_C . . . . .	7
3.1.3.9	csi . . . . .	7
3.1.3.10	currentCores_d . . . . .	7
3.1.3.11	datasetSize . . . . .	7
3.1.3.12	Deadline_d . . . . .	7
3.1.3.13	initialBaseFO . . . . .	8
3.1.3.14	m . . . . .	8
3.1.3.15	M . . . . .	8
3.1.3.16	mode . . . . .	8
3.1.3.17	nCores_DB_d . . . . .	8
3.1.3.18	nu_d . . . . .	8
3.1.3.19	R_bound_d . . . . .	8

3.1.3.20	R_d	8
3.1.3.21	sAB	8
3.1.3.22	session_app_id	8
3.1.3.23	stage	8
3.1.3.24	term_i	8
3.1.3.25	V	8
3.1.3.26	v	8
3.1.3.27	vm	8
3.1.3.28	w	8
3.2	Batch Class Reference	8
3.2.1	Constructor & Destructor Documentation	9
3.2.1.1	Batch	9
3.2.2	Member Function Documentation	9
3.2.2.1	approximatedLoop	10
3.2.2.2	calculate_nu	10
3.2.2.3	fixInitialSolution	11
3.2.2.4	initialize	11
3.2.3	Member Data Documentation	12
3.2.3.1	APPs	12
3.3	Bounds Class Reference	12
3.3.1	Member Function Documentation	12
3.3.1.1	calculateBounds	12
3.4	Candidate Class Reference	13
3.4.1	Constructor & Destructor Documentation	14
3.4.1.1	Candidate	14
3.4.2	Member Data Documentation	14
3.4.2.1	app_i	14
3.4.2.2	app_j	14
3.4.2.3	delta_i	14
3.4.2.4	delta_j	14
3.4.2.5	deltaFO	14
3.4.2.6	newCoreAssignment_i	14
3.4.2.7	newCoreAssignment_j	14
3.4.2.8	nodes_i	14
3.4.2.9	nodes_j	14
3.4.2.10	real_i	14
3.4.2.11	real_j	14
3.5	ObjFun Class Reference	14
3.5.1	Member Function Documentation	14
3.5.1.1	ObjFunctionComponent	15

3.5.1.2	ObjFunctionComponentApprox . . . . .	15
3.5.1.3	ObjFunctionGlobal . . . . .	16
3.6	optJrParameters Class Reference . . . . .	16
3.6.1	Constructor & Destructor Documentation . . . . .	17
3.6.1.1	optJrParameters . . . . .	17
3.6.2	Member Function Documentation . . . . .	17
3.6.2.1	get_cache . . . . .	17
3.6.2.2	get_debug . . . . .	18
3.6.2.3	get_filename . . . . .	18
3.6.2.4	get_globalFOcalculation . . . . .	18
3.6.2.5	get_K . . . . .	19
3.6.2.6	get_maxIteration . . . . .	19
3.6.2.7	get_number . . . . .	19
3.6.2.8	get_simulator . . . . .	20
3.7	sAlphaBetaManagement Class Reference . . . . .	20
3.7.1	Member Data Documentation . . . . .	20
3.7.1.1	index . . . . .	20
3.7.1.2	vec . . . . .	20
3.8	Search Class Reference . . . . .	21
3.8.1	Member Function Documentation . . . . .	21
3.8.1.1	localSearch . . . . .	21
3.9	slastSimulatorRun Class Reference . . . . .	22
3.9.1	Member Data Documentation . . . . .	22
3.9.1.1	nCores . . . . .	22
3.9.1.2	R . . . . .	22
<b>4</b>	<b>File Documentation</b>	<b>23</b>
4.1	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/appByWeight.cpp File Reference . . . . .	23
4.1.1	Function Documentation . . . . .	23
4.1.1.1	addApplicationPointer . . . . .	23
4.2	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/appByWeight.hh File Reference . . . . .	24
4.2.1	Typedef Documentation . . . . .	24
4.2.1.1	appByWeight . . . . .	24
4.2.2	Function Documentation . . . . .	24
4.2.2.1	addApplicationPointer . . . . .	25
4.3	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.cpp File Reference . . . . .	25
4.4	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh File Reference . . . . .	25
4.4.1	Macro Definition Documentation . . . . .	26
4.4.1.1	CORES_ALGORITHM . . . . .	26
4.4.1.2	HYP_INTERPOLATION_POINTS . . . . .	26

4.4.1.3	NCORES_ALGORITHM	26
4.4.1.4	R_ALGORITHM	26
4.4.1.5	RESIDUAL_EXECUTION_TIME	26
4.4.1.6	WHOLE_EXECUTION_TIME	26
4.5	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.cpp File Reference	26
4.6	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.hh File Reference	27
4.7	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.cpp File Reference	28
4.8	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.hh File Reference	28
4.9	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.cpp File Reference	29
4.9.1	Function Documentation	29
4.9.1.1	addCandidate	30
4.10	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.hh File Reference	30
4.10.1	Typedef Documentation	31
4.10.1.1	sCandidates	31
4.10.2	Function Documentation	31
4.10.2.1	addCandidate	31
4.11	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/db.cpp File Reference	31
4.11.1	Function Documentation	32
4.11.1.1	DBclose	32
4.11.1.2	DBerror	32
4.11.1.3	DBopen	32
4.11.1.4	executeSQL	33
4.12	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/db.hh File Reference	33
4.12.1	Function Documentation	34
4.12.1.1	DBclose	34
4.12.1.2	DBerror	34
4.12.1.3	DBopen	35
4.12.1.4	executeSQL	35
4.13	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/debugmessage.cpp File Reference	36
4.13.1	Function Documentation	36
4.13.1.1	debugMessage	36
4.14	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/debugmessage.hh File Reference	37
4.14.1	Function Documentation	38
4.14.1.1	debugMessage	38
4.15	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor.cpp File Reference	39
4.15.1	Function Documentation	39
4.15.1.1	invokePredictor	40
4.16	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor.hh File Reference	40
4.16.1	Macro Definition Documentation	41
4.16.1.1	RESIDUAL_DAGSIM	41

4.16.1.2	WHOLE_DAGSIM	41
4.16.2	Function Documentation	41
4.16.2.1	invokePredictor	42
4.17	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor_helper.cpp File Reference	42
4.17.1	Macro Definition Documentation	43
4.17.1.1	BIG_LINE	43
4.17.1.2	BIG_TEXT	43
4.17.2	Function Documentation	43
4.17.2.1	_run	44
4.17.2.2	extractRowMatchingPattern	44
4.17.2.3	extractRowN	44
4.17.2.4	extractWord	45
4.17.2.5	ls	45
4.17.2.6	readFile	45
4.17.2.7	readFolder	46
4.17.2.8	replace	46
4.17.2.9	writeFile	46
4.18	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor_helper.hh File Reference	47
4.18.1	Function Documentation	48
4.18.1.1	_run	48
4.18.1.2	extractRowMatchingPattern	48
4.18.1.3	extractRowN	48
4.18.1.4	extractWord	49
4.18.1.5	ls	49
4.18.1.6	readFile	49
4.18.1.7	readFolder	50
4.18.1.8	replace	50
4.18.1.9	writeFile	50
4.19	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/main.cpp File Reference	51
4.19.1	Function Documentation	51
4.19.1.1	main	52
4.20	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.cpp File Reference	52
4.21	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.hh File Reference	53
4.22	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParam_helper.cpp File Reference	54
4.22.1	Function Documentation	54
4.22.1.1	parseArg	54
4.22.1.2	Usage	55
4.23	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParam_helper.hh File Reference	55
4.23.1	Macro Definition Documentation	56
4.23.1.1	ARGS	56

4.23.1.2	DEBUG	56
4.23.1.3	FILENAME	56
4.23.1.4	GLOBAL_FO_CALCULATION	56
4.23.1.5	LIST_LIMIT	56
4.23.1.6	MAX_ITERATIONS	56
4.23.1.7	NO	57
4.23.1.8	NUM_N	57
4.23.1.9	NUMBER	57
4.23.1.10	SIMULATOR	57
4.23.1.11	STRING	57
4.23.1.12	YES	57
4.23.1.13	YES_NO	57
4.23.2	Function Documentation	57
4.23.2.1	parseArg	57
4.23.2.2	Usage	57
4.24	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrparameters.cpp File Reference	58
4.25	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParameters.hh File Reference	58
4.25.1	Macro Definition Documentation	59
4.25.1.1	DAGSIM	59
4.25.1.2	LUNDSTROM	59
4.26	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/read_app_file.cpp File Reference	59
4.26.1	Macro Definition Documentation	60
4.26.1.1	MAX_APP_LENGTH	60
4.26.2	Function Documentation	60
4.26.2.1	getfield	60
4.26.2.2	readAppFile	61
4.27	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/read_app_file.hh File Reference	61
4.27.1	Function Documentation	62
4.27.1.1	getfield	62
4.27.1.2	readAppFile	63
4.27.2	Variable Documentation	63
4.27.2.1	_APP_ID	63
4.27.2.2	_CHI_0	63
4.27.2.3	_CHI_C	63
4.27.2.4	_D	63
4.27.2.5	_Dsz	63
4.27.2.6	_M	63
4.27.2.7	_m	63
4.27.2.8	_SESSION_APP_ID	63
4.27.2.9	_St	63



4.27.2.10	_V	63
4.27.2.11	_v	63
4.27.2.12	_W	63
4.27.2.13	MAX_LINE_LENGTH	63
4.27.2.14	PARAMETERS	63
4.28	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/readConfigurationFile.cpp File Reference	64
4.28.1	Function Documentation	64
4.28.1.1	extractItem	64
4.28.1.2	readConfigurationFile	65
4.29	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/readConfigurationFile.hh File Reference	65
4.29.1	Typedef Documentation	66
4.29.1.1	sConfiguration	66
4.29.2	Function Documentation	66
4.29.2.1	readConfigurationFile	66
4.30	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.cpp File Reference	67
4.31	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.hh File Reference	67
4.32	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/utility.cpp File Reference	68
4.32.1	Function Documentation	69
4.32.1.1	doubleCompare	69
4.33	/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/utility.hh File Reference	70
4.33.1	Function Documentation	70
4.33.1.1	doubleCompare	70
4.33.2	Variable Documentation	70
4.33.2.1	epsilon	70



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Application</a>	5
<a href="#">Batch</a>	8
<a href="#">Bounds</a>	12
<a href="#">Candidate</a>	13
<a href="#">ObjFun</a>	14
<a href="#">optJrParameters</a>	16
<a href="#">sAlphaBetaManagement</a>	20
<a href="#">Search</a>	21
<a href="#">slastSimulatorRun</a>	22



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/appByWeight.cpp	23
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/appByWeight.hh	24
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.cpp	25
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh	25
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.cpp	26
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/batch.hh	27
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.cpp	28
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/bounds.hh	28
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.cpp	29
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.hh	30
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/db.cpp	31
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/db.hh	33
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/debugmessage.cpp	36
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/debugmessage.hh	37
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor.cpp	39
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor.hh	40
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor_helper.cpp	42
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/invokePredictor_helper.hh	47
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/main.cpp	51
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.cpp	52
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.hh	53
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParam_helper.cpp	54
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParam_helper.hh	55
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParameters.cpp	58
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParameters.hh	58
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/read_app_file.cpp	59
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/read_app_file.hh	61
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/readConfigurationFile.cpp	64
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/readConfigurationFile.hh	65
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.cpp	67
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/search.hh	67
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/utility.cpp	68
/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/utility.hh	70



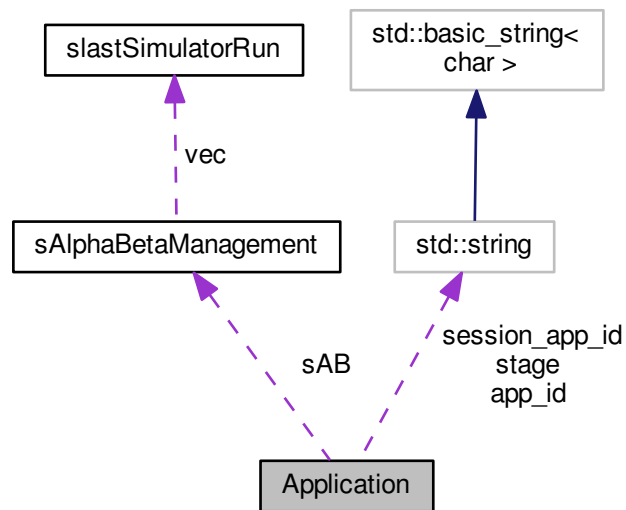
## Chapter 3

# Class Documentation

### 3.1 Application Class Reference

```
#include <application.hh>
```

Collaboration diagram for Application:



#### Public Member Functions

- `Application` (`std::string session_app_id`, `std::string app_id`, `double w`, `double chi_0`, `double chi_C`, `double m`, `double M`, `double V`, `double v`, `double D`, `double csi`, `std::string St`, `int DatasetSize`)
- `double ObjFunctionComponent` (`sConfiguration &configuration`, `MYSQL *conn`, `optJrParameters &par`)

#### Public Attributes

- `int mode`
- `std::string session_app_id`

- `std::string app_id`
- `double w`
- `double term_i`
- `double chi_0`
- `double chi_C`
- `double m`
- `double M`
- `double V`
- `double v`
- `double Deadline_d`
- `double csi`
- `std::string stage`
- `int datasetSize`
- `double nu_d`
- `int currentCores_d`
- `int nCores_DB_d`
- `int bound`
- `double R_d`
- `double R_bound_d`
- `double baseFO`
- `double initialBaseFO`
- `float alpha`
- `float beta`
- `sAlphaBetaManagement sAB`
- `int boundIterations`
- `int vm`

### 3.1.1 Constructor & Destructor Documentation

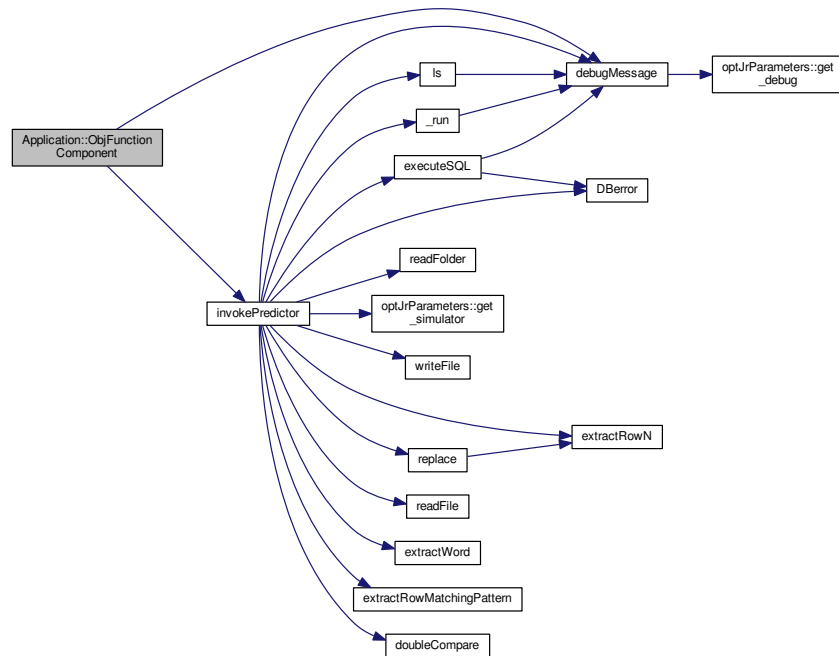
- 3.1.1.1 `Application::Application ( std::string session_app_id, std::string app_id, double w, double chi_0, double chi_C, double m, double M, double V, double v, double D, double csi, std::string St, int DatasetSize )`

### 3.1.2 Member Function Documentation



### 3.1.2.1 double Application::ObjFunctionComponent ( sConfiguration & configuration, MYSQL \* conn, optJrParameters & par )

Here is the call graph for this function:



### 3.1.3 Member Data Documentation

3.1.3.1 float Application::alpha

3.1.3.2 std::string Application::app\_id

3.1.3.3 double Application::baseFO

3.1.3.4 float Application::beta

3.1.3.5 int Application::bound

3.1.3.6 int Application::boundIterations

3.1.3.7 double Application::chi\_0

3.1.3.8 double Application::chi\_C

3.1.3.9 double Application::csi

3.1.3.10 int Application::currentCores\_d

3.1.3.11 int Application::datasetSize

3.1.3.12 double Application::Deadline\_d

- 3.1.3.13 double Application::initialBaseFO
- 3.1.3.14 double Application::m
- 3.1.3.15 double Application::M
- 3.1.3.16 int Application::mode
- 3.1.3.17 int Application::nCores\_DB\_d
- 3.1.3.18 double Application::nu\_d
- 3.1.3.19 double Application::R\_bound\_d
- 3.1.3.20 double Application::R\_d
- 3.1.3.21 sAlphaBetaManagement Application::sAB
- 3.1.3.22 std::string Application::session\_app\_id
- 3.1.3.23 std::string Application::stage
- 3.1.3.24 double Application::term\_i
- 3.1.3.25 double Application::V
- 3.1.3.26 double Application::v
- 3.1.3.27 int Application::vm
- 3.1.3.28 double Application::w

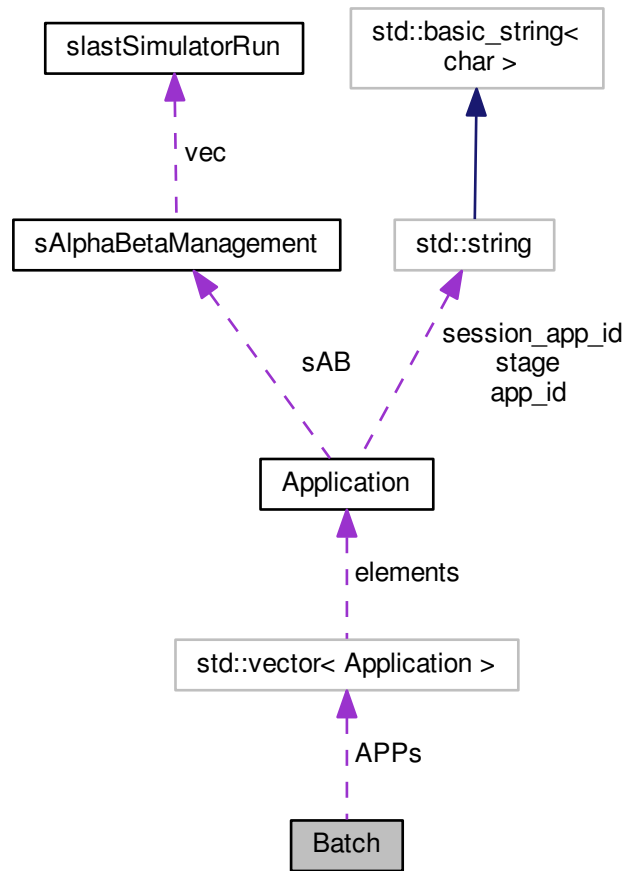
The documentation for this class was generated from the following files:

- /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/[application.hh](#)
- /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/[application.cpp](#)

## 3.2 Batch Class Reference

```
#include <batch.hh>
```

Collaboration diagram for Batch:



### Public Member Functions

- [Batch](#) (std::vector< [Application](#) > apps)
- void [calculate\\_nu](#) (optJrParameters &par)
- void [initialize](#) (sConfiguration &configuration, MYSQL \*conn, optJrParameters &par)
- void [fixInitialSolution](#) (optJrParameters &par)
- [sCandidates approximatedLoop](#) (int &iteration, optJrParameters &par)

### Public Attributes

- std::vector< [Application](#) > [APPs](#)

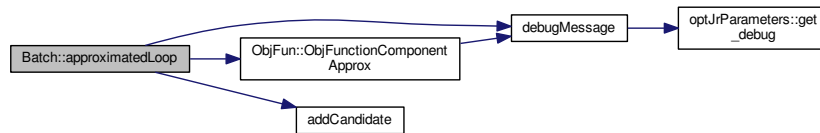
### 3.2.1 Constructor & Destructor Documentation

3.2.1.1 **Batch::Batch** ( std::vector< **Application** > *apps* ) [inline]

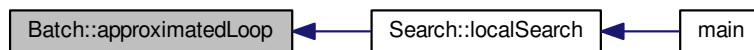
### 3.2.2 Member Function Documentation

### 3.2.2.1 sCandidates Batch::approximatedLoop ( int & iteration, optJrParameters & par )

Here is the call graph for this function:

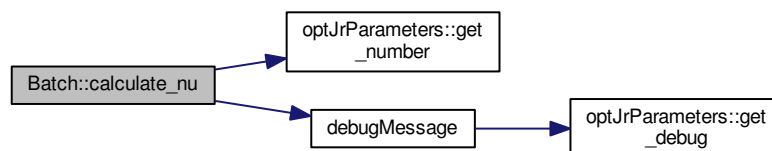


Here is the caller graph for this function:

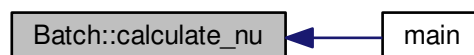


### 3.2.2.2 void Batch::calculate\_nu ( optJrParameters & par )

Here is the call graph for this function:

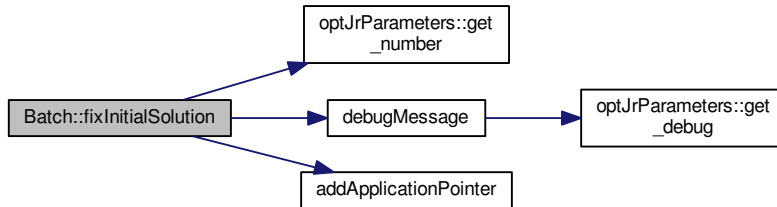


Here is the caller graph for this function:

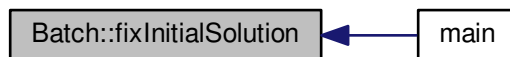


### 3.2.2.3 void Batch::fixInitialSolution ( optJrParameters & par )

Here is the call graph for this function:

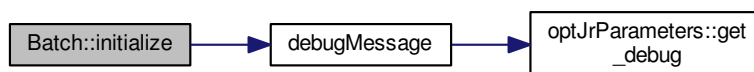


Here is the caller graph for this function:



### 3.2.2.4 void Batch::initialize ( sConfiguration & configuration, MYSQL \* conn, optJrParameters & par )

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.2.3 Member Data Documentation

#### 3.2.3.1 `std::vector<Application> Batch::APPs`

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

- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/batch.hh](#)
- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/batch.cpp](#)

## 3.3 Bounds Class Reference

```
#include <bounds.hh>
```

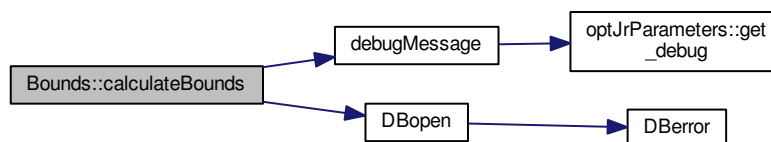
### Static Public Member Functions

- static void [calculateBounds](#) ([Batch](#) &app\_manager, int n\_threads, [sConfiguration](#) &configuration, MYSQL \*conn, [optJrParameters](#) &par)

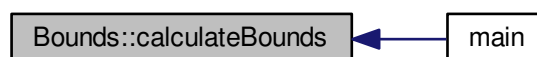
#### 3.3.1 Member Function Documentation

3.3.1.1 void `Bounds::calculateBounds` ( `Batch` &*app\_manager*, int *n\_threads*, `sConfiguration` &*configuration*, MYSQL \**conn*, `optJrParameters` &*par* ) [static]

Here is the call graph for this function:



Here is the caller graph for this function:



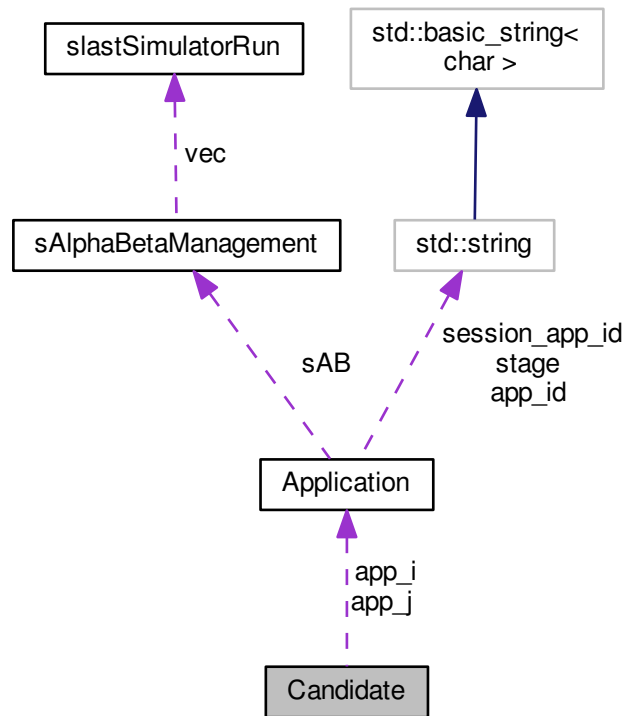
The documentation for this class was generated from the following files:

- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/bounds.hh](#)
- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/bounds.cpp](#)

### 3.4 Candidate Class Reference

```
#include <candidates.hh>
```

Collaboration diagram for Candidate:



#### Public Member Functions

- `Candidate` (`Application` \*i, `Application` \*j, int NCi, int NCj, double D\_FO, int d\_i, int d\_j)

#### Public Attributes

- `Application` \* app\_i
- int newCoreAssignment\_i
- int delta\_i
- double real\_i
- `Application` \* app\_j
- int newCoreAssignment\_j
- int delta\_j
- double real\_j
- int nodes\_i
- int nodes\_j
- double deltaFO

### 3.4.1 Constructor & Destructor Documentation

3.4.1.1 `Candidate::Candidate ( Application * i, Application * j, int NCi, int NCj, double D_FO, int d_i, int d_j )`  
`[inline]`

### 3.4.2 Member Data Documentation

3.4.2.1 `Application*` `Candidate::app_i`

3.4.2.2 `Application*` `Candidate::app_j`

3.4.2.3 `int` `Candidate::delta_i`

3.4.2.4 `int` `Candidate::delta_j`

3.4.2.5 `double` `Candidate::deltaFO`

3.4.2.6 `int` `Candidate::newCoreAssignment_i`

3.4.2.7 `int` `Candidate::newCoreAssignment_j`

3.4.2.8 `int` `Candidate::nodes_i`

3.4.2.9 `int` `Candidate::nodes_j`

3.4.2.10 `double` `Candidate::real_i`

3.4.2.11 `double` `Candidate::real_j`

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

- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/candidates.hh`

## 3.5 ObjFun Class Reference

```
#include <objectiveFunction.hh>
```

### Static Public Member Functions

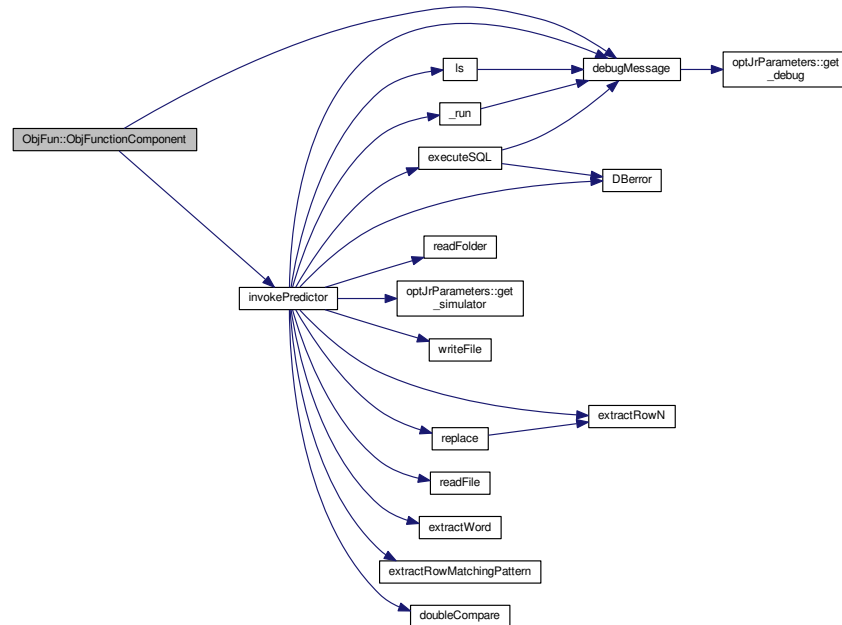
- static double `ObjFunctionComponent` (`sConfiguration` &configuration, `MYSQL *`conn, `Application` &app, `opt-JrParameters` &par)
- static double `ObjFunctionComponentApprox` (`Application` &App, `optJrParameters` &par)
- static double `ObjFunctionGlobal` (`sConfiguration` &configuration, `MYSQL *`conn, `Batch` &App\_manager, `opt-JrParameters` &par)

### 3.5.1 Member Function Documentation

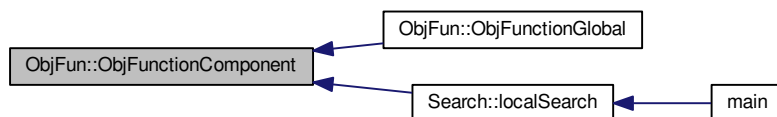


3.5.1.1 `double ObjFun::ObjFunctionComponent ( sConfiguration & configuration, MYSQL * conn, Application & app, optJrParameters & par ) [static]`

Here is the call graph for this function:



Here is the caller graph for this function:

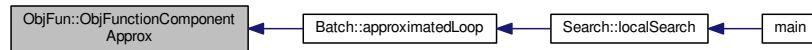


3.5.1.2 `double ObjFun::ObjFunctionComponentApprox ( Application & App, optJrParameters & par ) [static]`

Here is the call graph for this function:

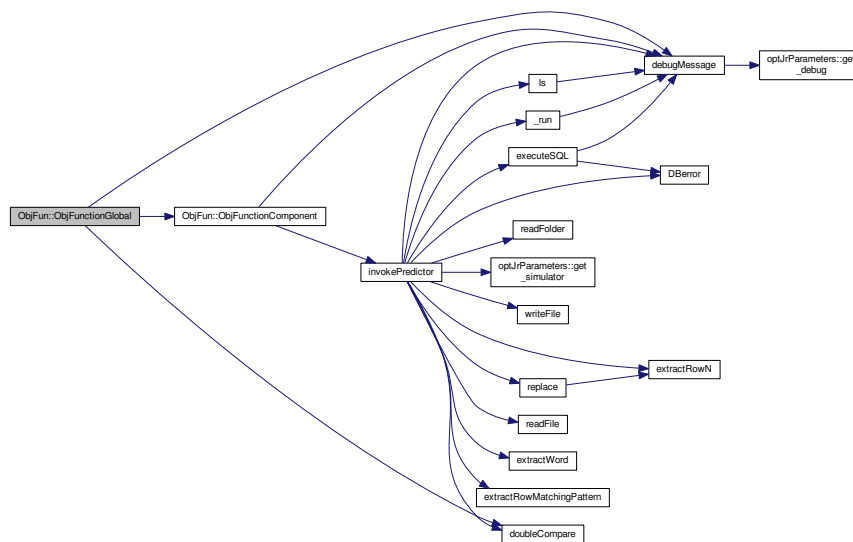


Here is the caller graph for this function:



3.5.1.3 `double ObjFun::ObjFunctionGlobal ( sConfiguration & configuration, MYSQL * conn, Batch & App_manager, optJrParameters & par ) [static]`

Here is the call graph for this function:



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

- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.hh`
- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.cpp`

## 3.6 optJrParameters Class Reference

```
#include <optJrParameters.hh>
```

### Public Member Functions

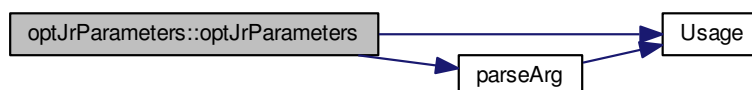
- `optJrParameters (char **args, int argc)`
- `const std::string get_filename ()`
- `const int get_debug ()`
- `const int get_cache ()`
- `const int get_globalFOcalculation ()`
- `const int get_K ()`
- `const int get_simulator ()`

- const int [get\\_number](#) ()
- const int [get\\_maxIteration](#) ()

### 3.6.1 Constructor & Destructor Documentation

#### 3.6.1.1 optJrParameters::optJrParameters ( char \*\* args, int argc )

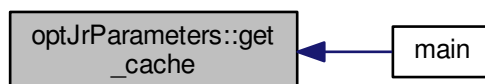
Here is the call graph for this function:



### 3.6.2 Member Function Documentation

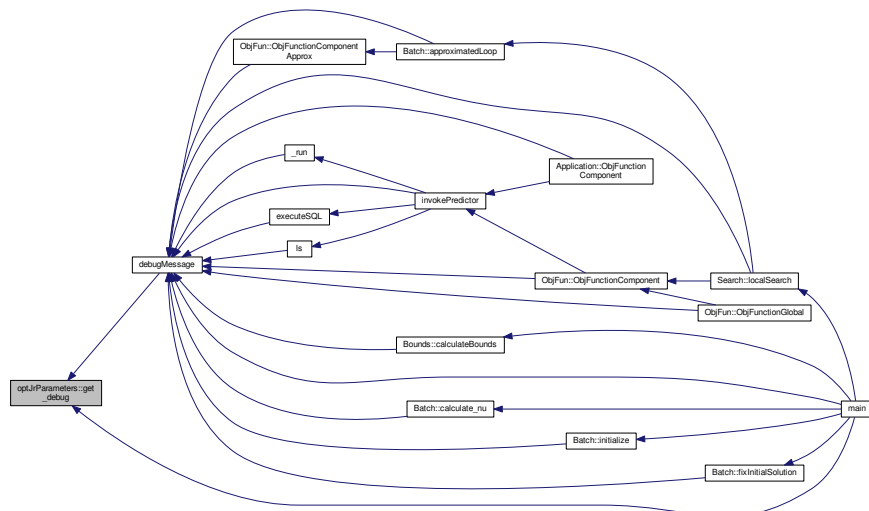
#### 3.6.2.1 const int optJrParameters::get\_cache ( )

Here is the caller graph for this function:



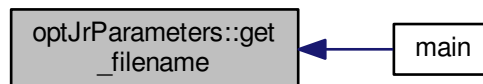
### 3.6.2.2 `const int optJrParameters::get_debug ( )`

Here is the caller graph for this function:



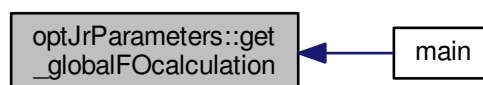
### 3.6.2.3 `const std::string optJrParameters::get_filename ( )`

Here is the caller graph for this function:



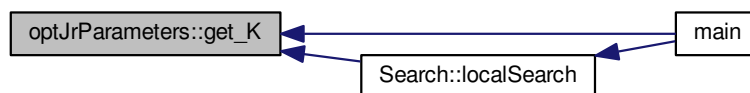
### 3.6.2.4 `const int optJrParameters::get_globalFOcalculation ( )`

Here is the caller graph for this function:



3.6.2.5 `const int optJrParameters::get_K ( )`

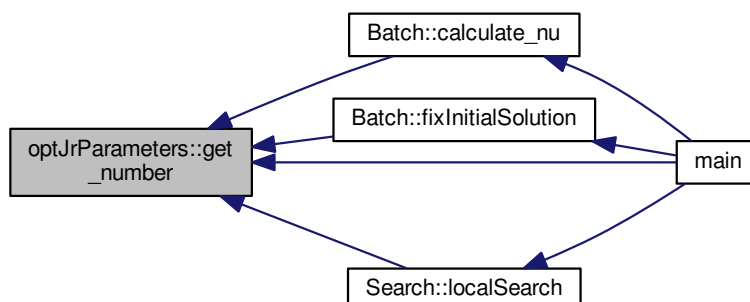
Here is the caller graph for this function:

3.6.2.6 `const int optJrParameters::get_maxIteration ( )`

Here is the caller graph for this function:

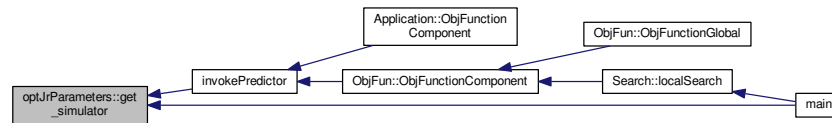
3.6.2.7 `const int optJrParameters::get_number ( )`

Here is the caller graph for this function:



### 3.6.2.8 `const int optJrParameters::get_simulator ( )`

Here is the caller graph for this function:



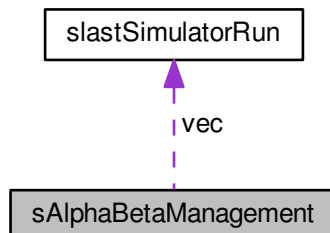
The documentation for this class was generated from the following files:

- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrParameters.hh`
- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/optjrparameters.cpp`

## 3.7 sAlphaBetaManagement Class Reference

```
#include <application.hh>
```

Collaboration diagram for sAlphaBetaManagement:



### Public Attributes

- `slastSimulatorRun vec [HYP_INTERPOLATION_POINTS]`
- `int index`

### 3.7.1 Member Data Documentation

#### 3.7.1.1 `int sAlphaBetaManagement::index`

#### 3.7.1.2 `slastSimulatorRun sAlphaBetaManagement::vec[HYP_INTERPOLATION_POINTS]`

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

- `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/application.hh`

## 3.8 Search Class Reference

```
#include <search.hh>
```

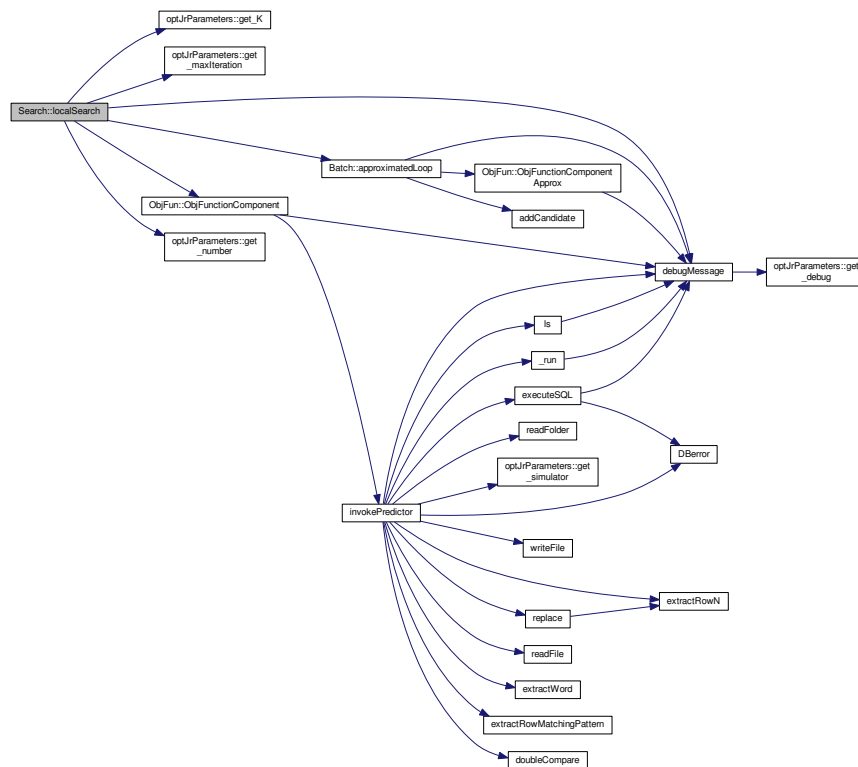
### Static Public Member Functions

- static void [localSearch](#) ([sConfiguration](#) &configuration, [MYSQL](#) \*conn, [Batch](#) &App\_manager, [optJrParameters](#) &par)

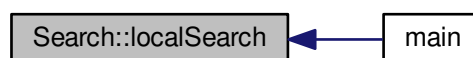
#### 3.8.1 Member Function Documentation

3.8.1.1 void [Search::localSearch](#) ( [sConfiguration](#) & configuration, [MYSQL](#) \* conn, [Batch](#) & App\_manager, [optJrParameters](#) & par ) [static]

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:

- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/search.hh](#)
- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/search.cpp](#)

## 3.9 slastSimulatorRun Class Reference

```
#include <application.hh>
```

### Public Attributes

- `int` [nCores](#)
- `double` [R](#)

### 3.9.1 Member Data Documentation

3.9.1.1 `int` `slastSimulatorRun::nCores`

3.9.1.2 `double` `slastSimulatorRun::R`

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

- [/vagrant/PROJECT\\_SPARK/PACS\\_PROJECT/opt\\_jr/src/application.hh](#)



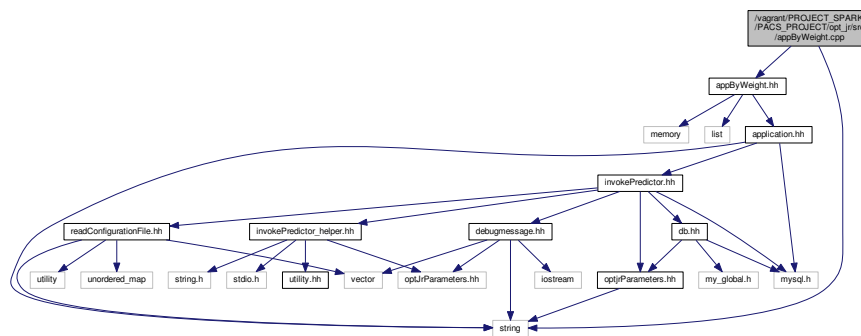
## Chapter 4

# File Documentation

### 4.1 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/appByWeight.cpp File Reference

```
#include "appByWeight.hh"
```

Include dependency graph for appByWeight.cpp:



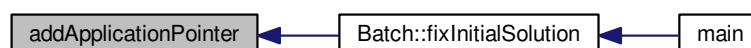
## Functions

- void [addApplicationPointer](#) ([appByWeight](#) &LP, [Application](#) &App)

### 4.1.1 Function Documentation

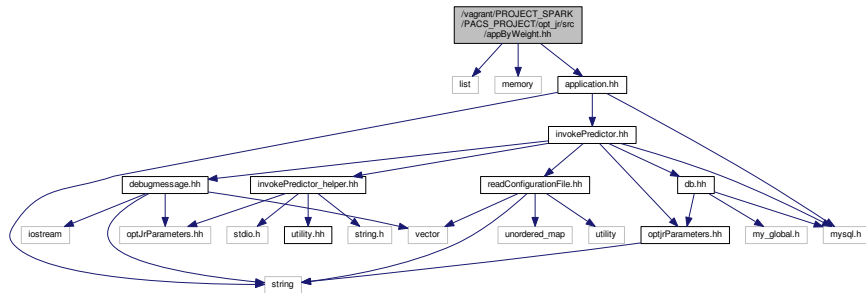
#### 4.1.1.1 void addApplicationPointer ( [appByWeight](#) &LP, [Application](#) &App )

Here is the caller graph for this function:

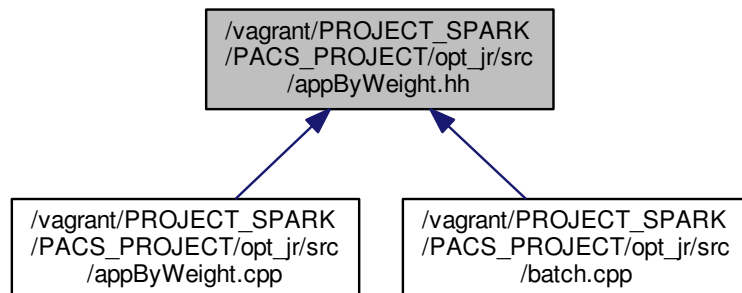


## 4.2 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/appByWeight.hh File Reference

```
#include <list>
#include <memory>
#include "application.hh"
Include dependency graph for appByWeight.hh:
```



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



### Typedefs

- using `appByWeight` = `std::list< Application * >`

### Functions

- void `addApplicationPointer` (`appByWeight &LP`, `Application &App`)

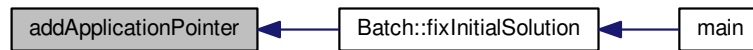
#### 4.2.1 Typedef Documentation

- 4.2.1.1 using `appByWeight` = `std::list< Application* >`

#### 4.2.2 Function Documentation

## 4.2.2.1 void addApplicationPointer ( appByWeight &amp; LP, Application &amp; App )

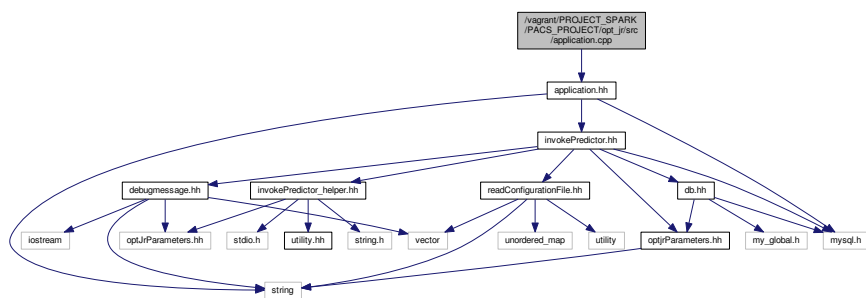
Here is the caller graph for this function:



## 4.3 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/application.cpp File Reference

```
#include "application.hh"
```

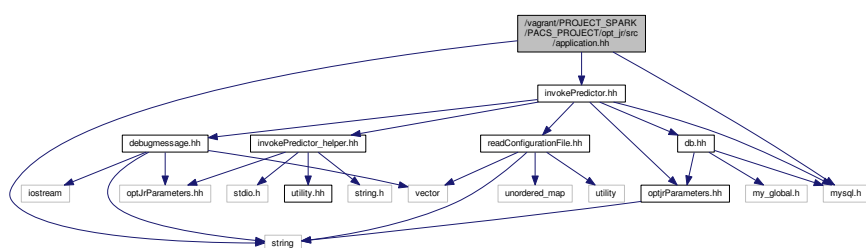
Include dependency graph for application.cpp:



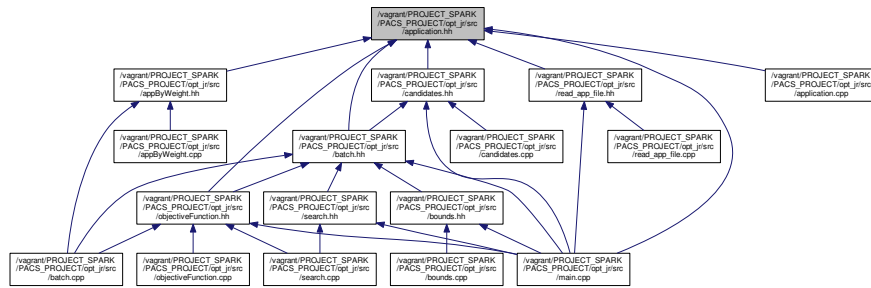
## 4.4 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/application.hh File Reference

```
#include <string>
#include <mysql.h>
#include "invokePredictor.hh"
```

Include dependency graph for application.hh:



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



## Classes

- class [slastSimulatorRun](#)
- class [sAlphaBetaManagement](#)
- class [Application](#)

## Macros

- `#define` [HYP\\_INTERPOLATION\\_POINTS](#) 2
- `#define` [WHOLE\\_EXECUTION\\_TIME](#) 0
- `#define` [RESIDUAL\\_EXECUTION\\_TIME](#) 1
- `#define` [R\\_ALGORITHM](#) 0
- `#define` [CORES\\_ALGORITHM](#) 1
- `#define` [NCORES\\_ALGORITHM](#) 2

### 4.4.1 Macro Definition Documentation

#### 4.4.1.1 `#define` [CORES\\_ALGORITHM](#) 1

#### 4.4.1.2 `#define` [HYP\\_INTERPOLATION\\_POINTS](#) 2

#### 4.4.1.3 `#define` [NCORES\\_ALGORITHM](#) 2

#### 4.4.1.4 `#define` [R\\_ALGORITHM](#) 0

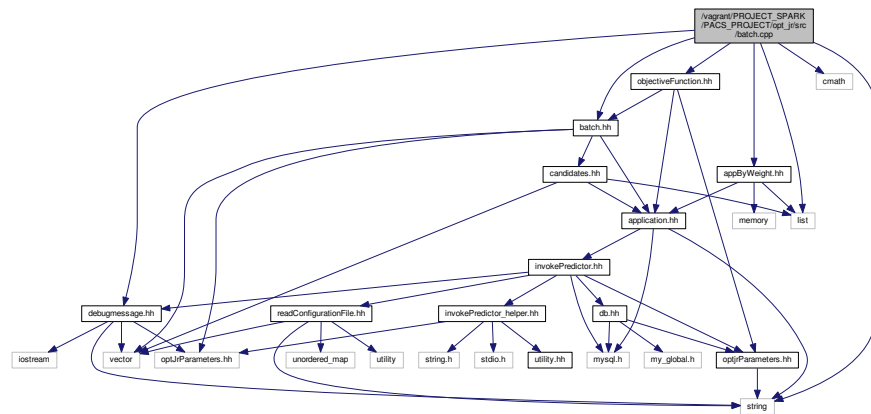
#### 4.4.1.5 `#define` [RESIDUAL\\_EXECUTION\\_TIME](#) 1

#### 4.4.1.6 `#define` [WHOLE\\_EXECUTION\\_TIME](#) 0

## 4.5 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/batch.cpp File Reference

```
#include "batch.hh"
#include "debugmessage.hh"
#include "objectiveFunction.hh"
#include "appByWeight.hh"
#include <string>
#include <cmath>
#include <list>
```

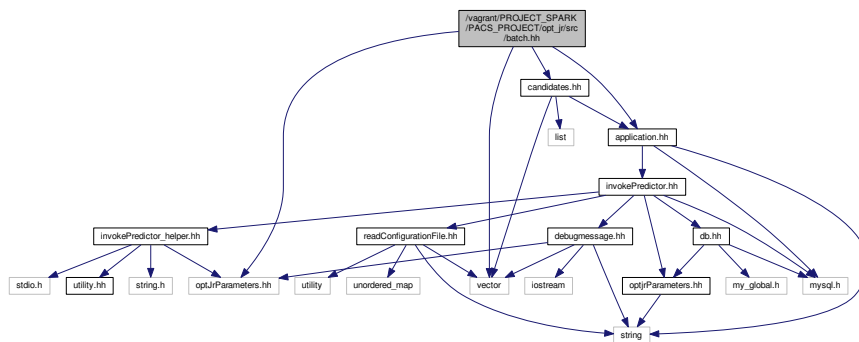
Include dependency graph for batch.cpp:



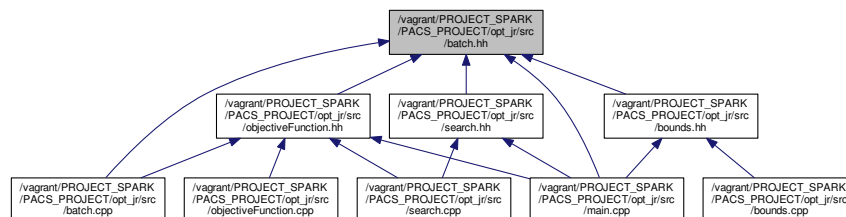
## 4.6 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/batch.h File Reference

```
#include <vector>
#include "optJrParameters.hh"
#include "application.hh"
#include "candidates.hh"
```

Include dependency graph for batch.h:



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



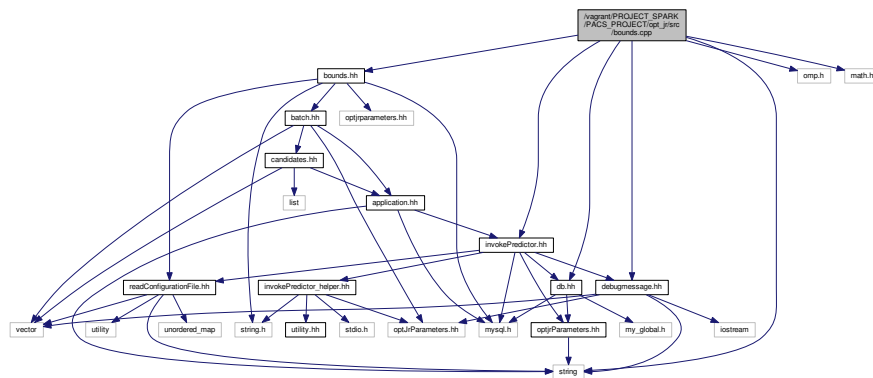
## Classes

- class [Batch](#)

## 4.7 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/bounds.cpp File Reference

```
#include "bounds.hh"
#include "debugmessage.hh"
#include "db.hh"
#include "invokePredictor.hh"
#include <omp.h>
#include <math.h>
```

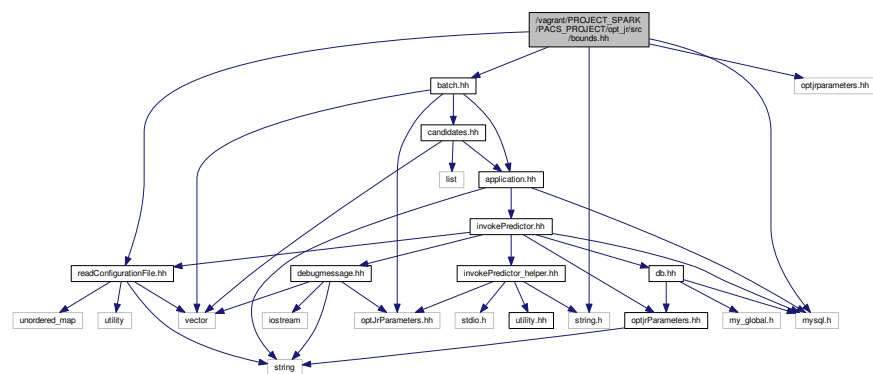
Include dependency graph for bounds.cpp:



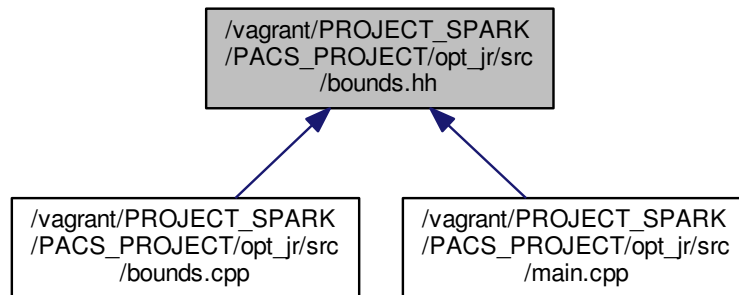
## 4.8 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/bounds.hh File Reference

```
#include "batch.hh"
#include "readConfigurationFile.hh"
#include "optjParameters.hh"
#include <mysql.h>
#include <string.h>
```

Include dependency graph for bounds.hh:



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



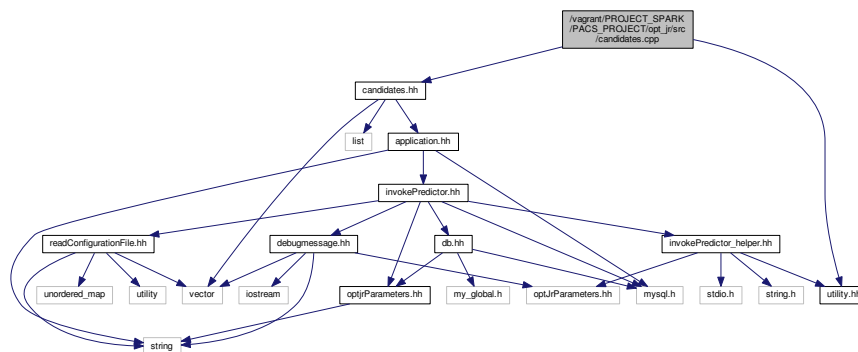
## Classes

- class [Bounds](#)

## 4.9 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/candidates.cpp File Reference

```
#include "candidates.hh"
#include "utility.hh"
```

Include dependency graph for candidates.cpp:



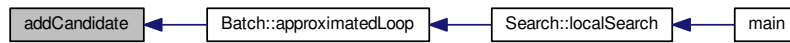
## Functions

- void [addCandidate](#) ([sCandidates](#) &cand, [Application](#) &app\_i, [Application](#) &app\_j, int contr1, int contr2, double delta, double delta\_i, double delta\_j)

### 4.9.1 Function Documentation

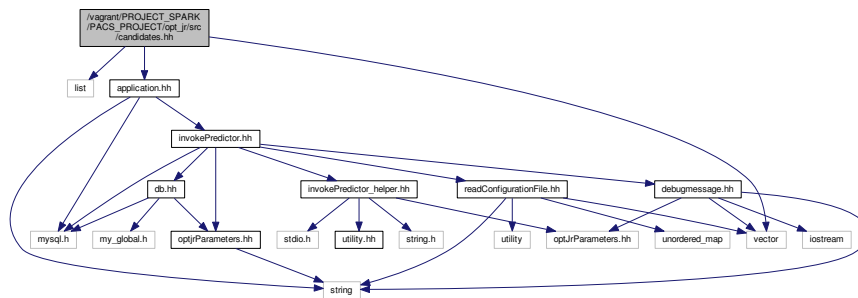
4.9.1.1 void addCandidate ( sCandidates & cand, Application & app\_i, Application & app\_j, int contr1, int contr2, double delta, double delta\_i, double delta\_j )

Here is the caller graph for this function:

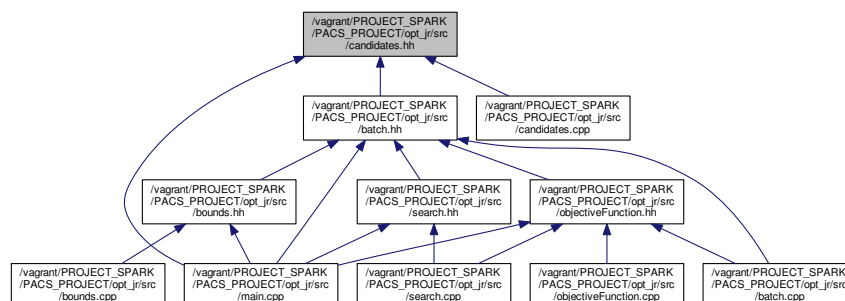


## 4.10 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/candidates.hh File Reference

```
#include <list>
#include <vector>
#include "application.hh"
Include dependency graph for candidates.hh:
```



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



## Classes

- class [Candidate](#)



## Typedefs

- using `sCandidates` = `std::list< Candidate >`

## Functions

- void `addCandidate` (`sCandidates` &`cand`, `Application` &`app_i`, `Application` &`app_j`, int `contr1`, int `contr2`, double `delta`, double `delta_i`, double `delta_j`)

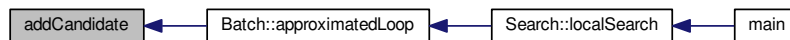
### 4.10.1 Typedef Documentation

#### 4.10.1.1 using `sCandidates` = `std::list<Candidate>`

### 4.10.2 Function Documentation

#### 4.10.2.1 void `addCandidate` ( `sCandidates` &`cand`, `Application` &`app_i`, `Application` &`app_j`, int `contr1`, int `contr2`, double `delta`, double `delta_i`, double `delta_j` )

Here is the caller graph for this function:



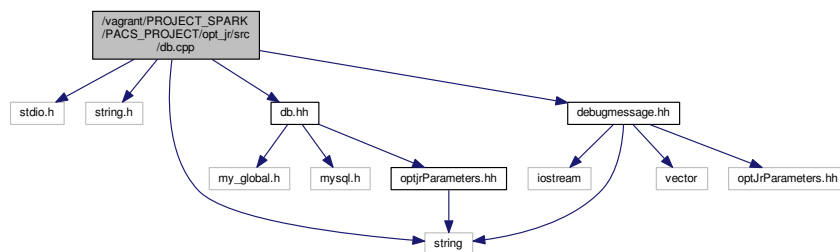
## 4.11 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/db.cpp File Reference

```

#include <stdio.h>
#include <string.h>
#include <string>
#include "db.hh"
#include "debugmessage.hh"

```

Include dependency graph for db.cpp:



## Functions

- void `DBerror` (`MYSQL` \*`conn`, char \*`msg`)
- `MYSQL_ROW` `executeSQL` (`MYSQL` \*`conn`, char \*`statement`, `optJrParameters` par)

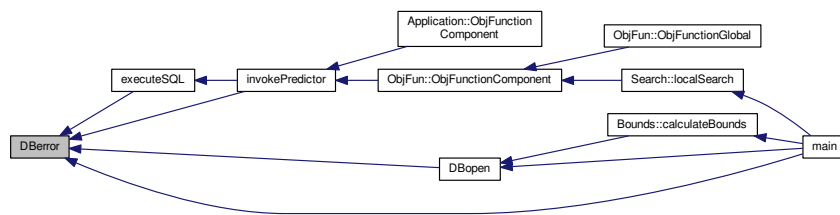
- MYSQL \* [DBopen](#) (char \*host, char \*login, char \*passw, char \*dbName)
- void [DBclose](#) (MYSQL \*conn)

#### 4.11.1 Function Documentation

4.11.1.1 void [DBclose](#) ( MYSQL \* *conn* )

4.11.1.2 void [DBerror](#) ( MYSQL \* *conn*, char \* *msg* )

Here is the caller graph for this function:

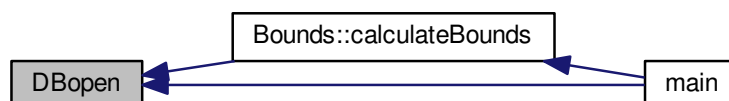


4.11.1.3 MYSQL\* [DBopen](#) ( char \* *host*, char \* *login*, char \* *passw*, char \* *dbName* )

Here is the call graph for this function:

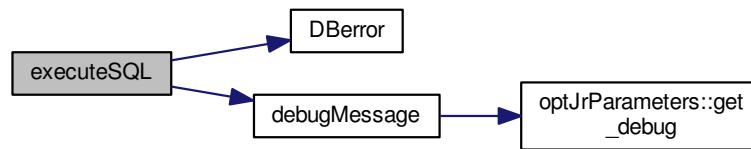


Here is the caller graph for this function:

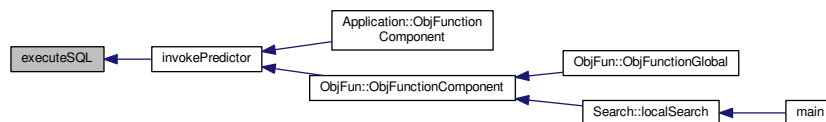


## 4.11.1.4 MYSQL\_ROW executeSQL ( MYSQL \* conn, char \* statement, optJrParameters par )

Here is the call graph for this function:



Here is the caller graph for this function:



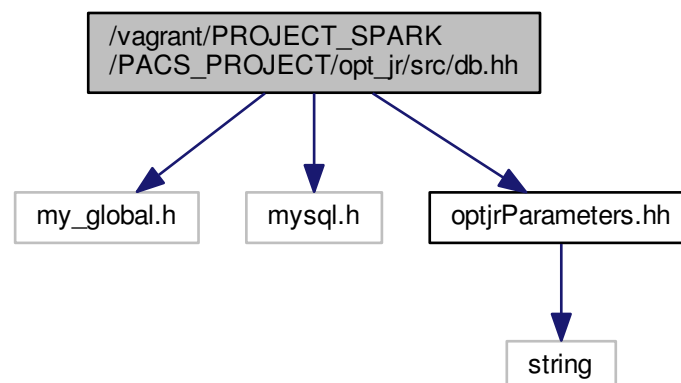
## 4.12 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/db.hh File Reference

```

#include <my_global.h>
#include <mysql.h>
#include "optjrParameters.hh"

```

Include dependency graph for db.hh:



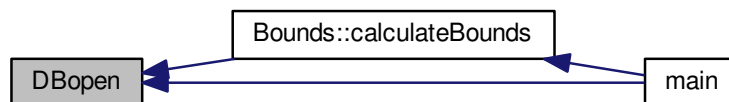


4.12.1.3 MySQL\* DBopen ( char \* *host*, char \* *login*, char \* *passw*, char \* *dbName* )

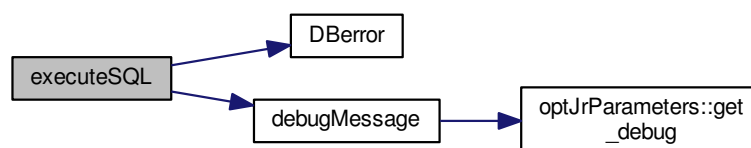
Here is the call graph for this function:



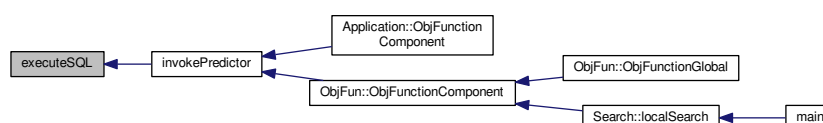
Here is the caller graph for this function:

4.12.1.4 MYSQL\_ROW executeSQL ( MYSQL \* *conn*, char \* *statement*, optJrParameters *par* )

Here is the call graph for this function:



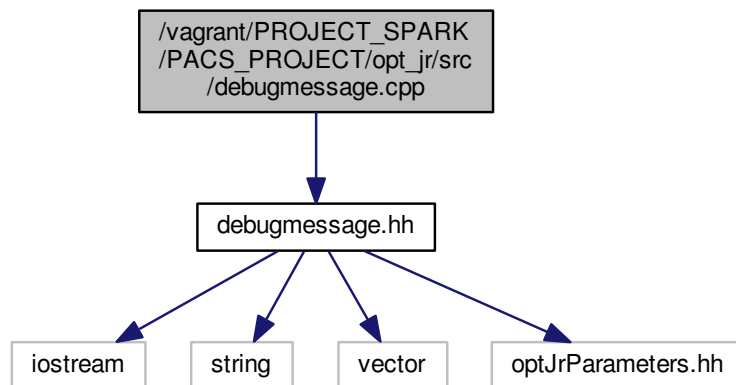
Here is the caller graph for this function:



## 4.13 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/debugmessage.cpp File Reference

```
#include "debugmessage.hh"
```

Include dependency graph for debugmessage.cpp:



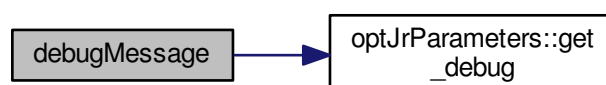
### Functions

- void `debugMessage` (std::string &string, `optJrParameters` &par)

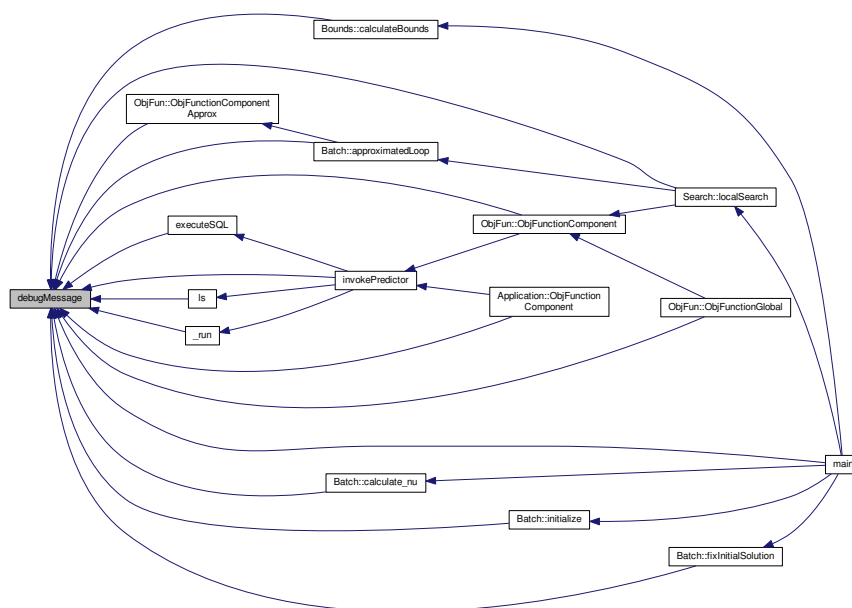
#### 4.13.1 Function Documentation

4.13.1.1 void `debugMessage` ( std::string & *string*, `optJrParameters` & *par* )

Here is the call graph for this function:



Here is the caller graph for this function:



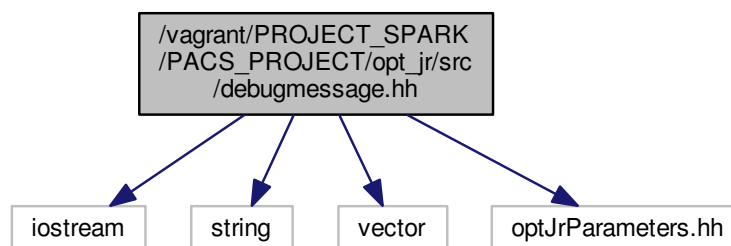
#### 4.14 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/debugmessage.hh File Reference

```

#include <iostream>
#include <string>
#include <vector>
#include "optJrParameters.hh"

```

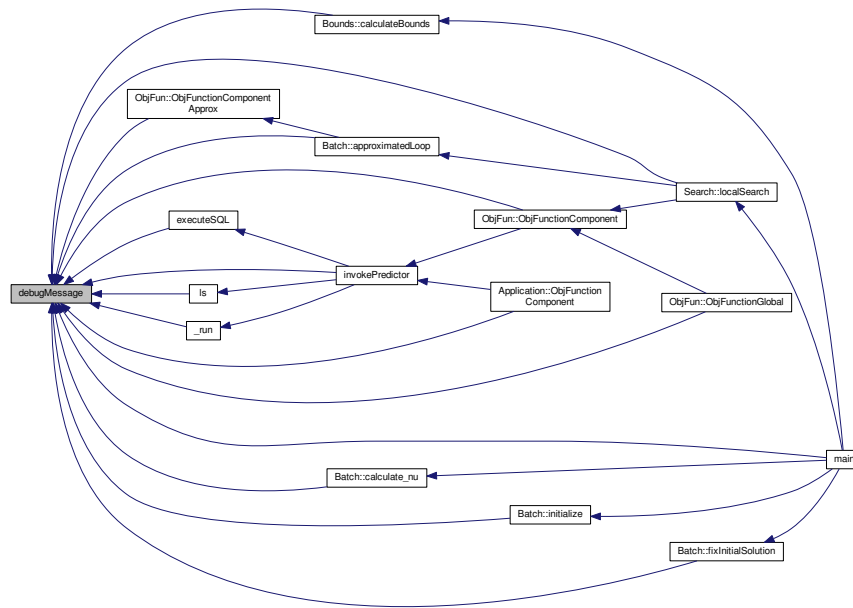
Include dependency graph for debugmessage.hh:





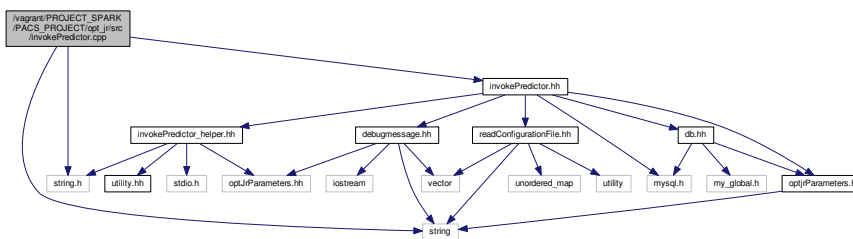


Here is the caller graph for this function:



## 4.15 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/invokePredictor.cpp File Reference

```
#include "invokePredictor.hh"
#include <string>
#include <string.h>
Include dependency graph for invokePredictor.cpp:
```



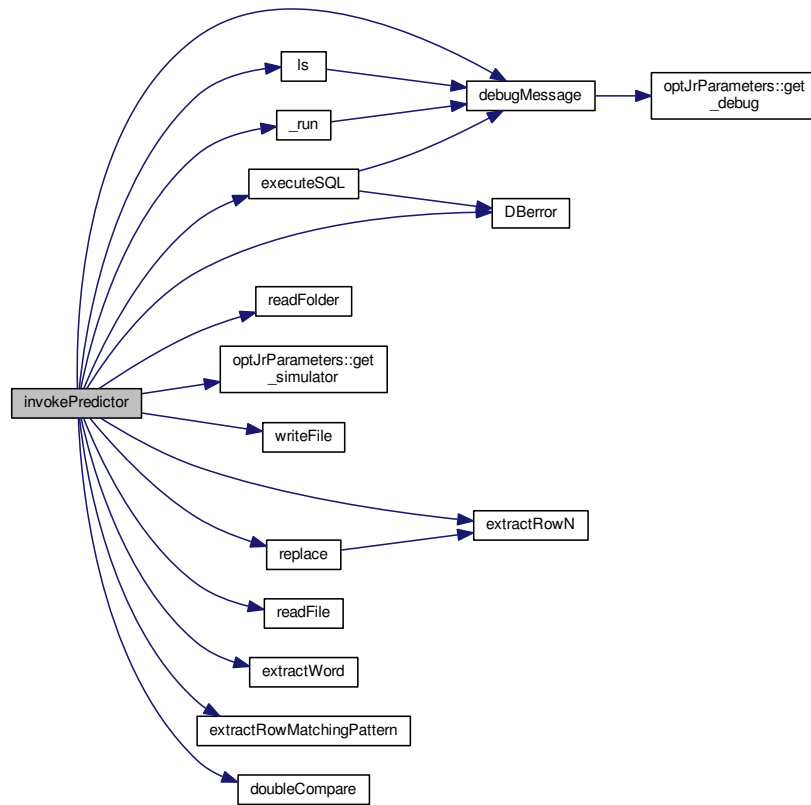
### Functions

- char \* [invokePredictor](#) ([sConfiguration](#) &configuration, MYSQL \*conn, int nNodes, int currentCores, char \*memory, int datasize, char \*sessionId, char \*appld, char \*stage, [optJrParameters](#) &par, int flagDagsim)

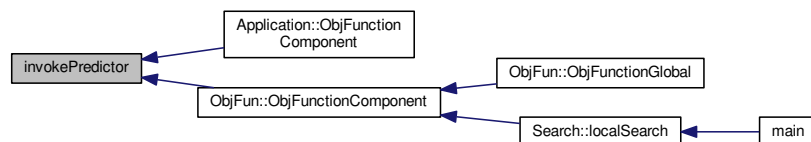
#### 4.15.1 Function Documentation

4.15.1.1 `char* invokePredictor ( sConfiguration & configuration, MYSQL * conn, int nNodes, int currentCores, char * memory, int datasize, char * sessionId, char * appld, char * stage, optJrParameters & par, int flagDagsim )`

Here is the call graph for this function:



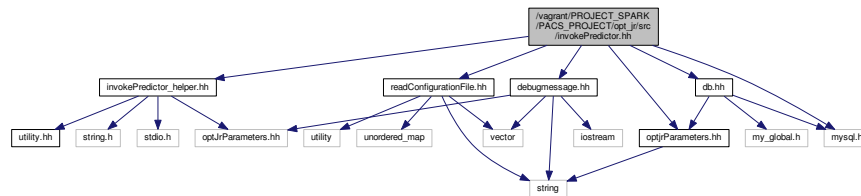
Here is the caller graph for this function:



## 4.16 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/invokePredictor.hh File Reference

```
#include "invokePredictor_helper.hh"
```

Include dependency graph for invokePredictor.hh:



```

graph TD
    SPARK_PROJECT[PROJECT.SPARK]
    SPARK_MAIN[PROJECT.SPARK main.cpp]
    SPARK_BATCH[PROJECT.SPARK batch.cpp]
    SPARK_SEARCH[PROJECT.SPARK search.cpp]
    SPARK_REPLICATION[PROJECT.SPARK replication.cpp]
    SPARK_CANDIDATE[PROJECT.SPARK candidate.cpp]
    SPARK_READ[PROJECT.SPARK read.cpp]
    SPARK_PREDICTOR[PROJECT.SPARK predictor.cpp]
    SPARK_MODEL[PROJECT.SPARK model.cpp]

    SPARK_PROJECT --> SPARK_MAIN
    SPARK_PROJECT --> SPARK_BATCH
    SPARK_PROJECT --> SPARK_SEARCH
    SPARK_PROJECT --> SPARK_REPLICATION
    SPARK_PROJECT --> SPARK_CANDIDATE
    SPARK_PROJECT --> SPARK_READ
    SPARK_PROJECT --> SPARK_PREDICTOR
    SPARK_PROJECT --> SPARK_MODEL
  
```

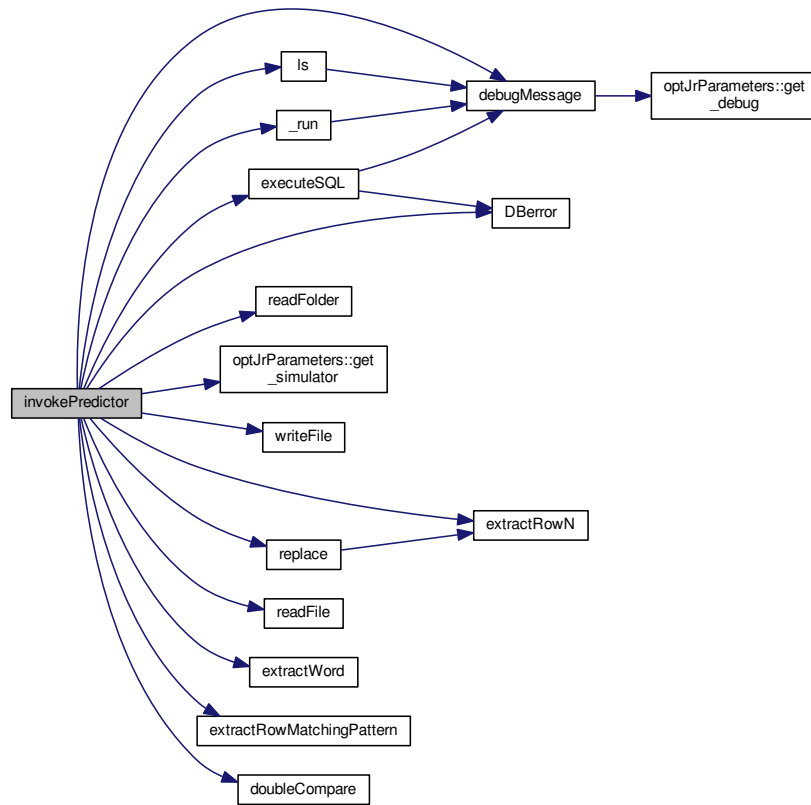
- #define WHOLE\_DAGSIM 0
- #define RESIDUAL\_DAGSIM 1

- char \* [invokePredictor](#) ([SConfiguration](#) &configuration, MYSQL \*conn, int nNodes, int currentCores, char \*memory, int datasize, char \*sessionId, char \*appld, char \*stage, [optJrParameters](#) &par, int flagDagsim)

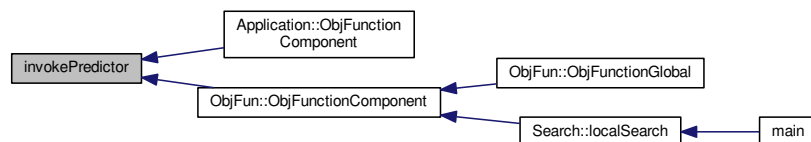
### 4.16.2 Function Documentation

4.16.2.1 `char* invokePredictor ( sConfiguration & configuration, MYSQL * conn, int nNodes, int currentCores, char * memory, int datasize, char * sessionId, char * appld, char * stage, optJrParameters & par, int flagDagsim )`

Here is the call graph for this function:



Here is the caller graph for this function:

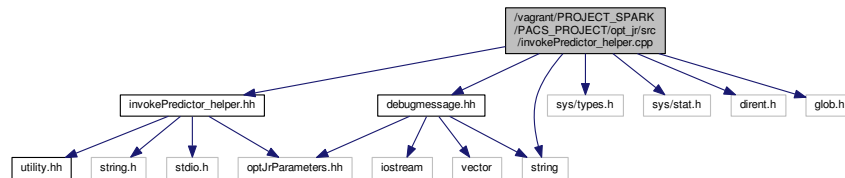


## 4.17 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/invokePredictor\_helper.cpp File Reference

```
#include "invokePredictor_helper.hh"
```

```
#include "debugmessage.hh"
#include <string>
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <glob.h>
```

Include dependency graph for InvokePredictor\_helper.cpp:



## Macros

- `#define BIG_LINE 4000`
- `#define BIG_TEXT 20000`

## Functions

- `char * readFolder (char *path)`
- `void writeFile (const char *filepath, const char *data)`
- `char * ls (char *pattern, optJrParameters &par)`
- `char * extractRowN (char *text, int row)`
- `char * replace (char *text, char *newline)`
- `char * readFile (char *filename)`
- `char * _run (char *cmd, optJrParameters &par)`
- `char * extractWord (char *line, int pos)`
- `char * extractRowMatchingPattern (char *text, char *pattern)`

### 4.17.1 Macro Definition Documentation

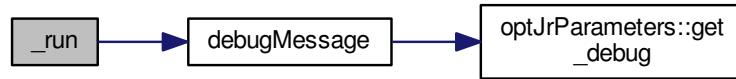
#### 4.17.1.1 `#define BIG_LINE 4000`

#### 4.17.1.2 `#define BIG_TEXT 20000`

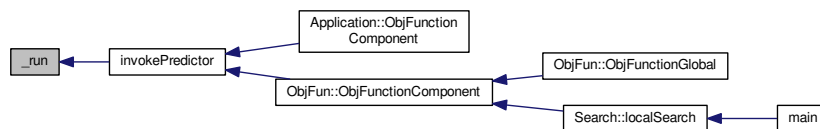
### 4.17.2 Function Documentation

#### 4.17.2.1 `char* _run ( char * cmd, optJrParameters & par )`

Here is the call graph for this function:

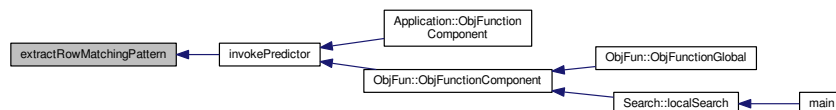


Here is the caller graph for this function:



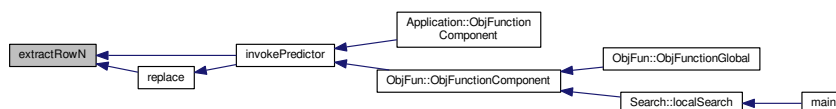
#### 4.17.2.2 `char* extractRowMatchingPattern ( char * text, char * pattern )`

Here is the caller graph for this function:



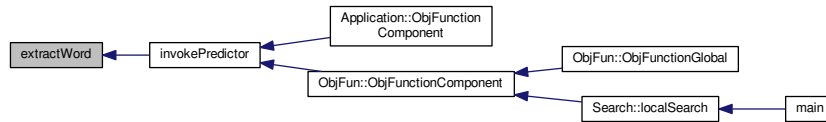
#### 4.17.2.3 `char* extractRowN ( char * text, int row )`

Here is the caller graph for this function:

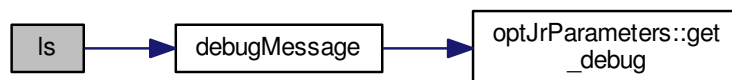


4.17.2.4 `char* extractWord ( char * line, int pos )`

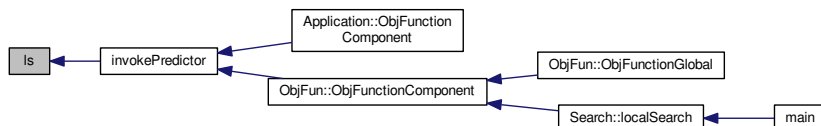
Here is the caller graph for this function:

4.17.2.5 `char* ls ( char * pattern, optJrParameters & par )`

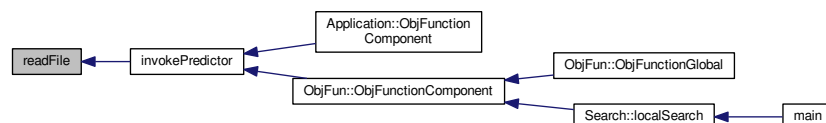
Here is the call graph for this function:



Here is the caller graph for this function:

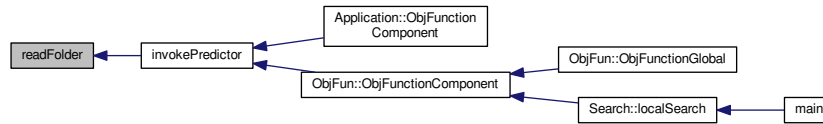
4.17.2.6 `char* readFile ( char * filename )`

Here is the caller graph for this function:



#### 4.17.2.7 `char* readFolder ( char * path )`

Here is the caller graph for this function:

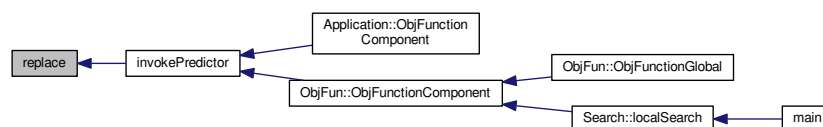


#### 4.17.2.8 `char* replace ( char * text, char * newLine )`

Here is the call graph for this function:

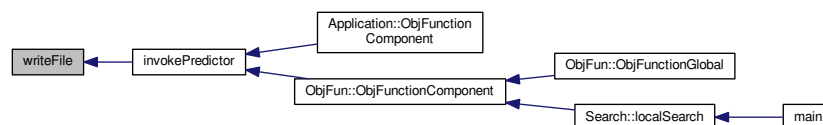


Here is the caller graph for this function:



#### 4.17.2.9 `void writeFile ( const char * filepath, const char * data )`

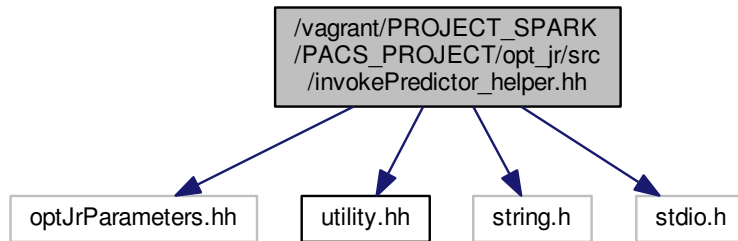
Here is the caller graph for this function:



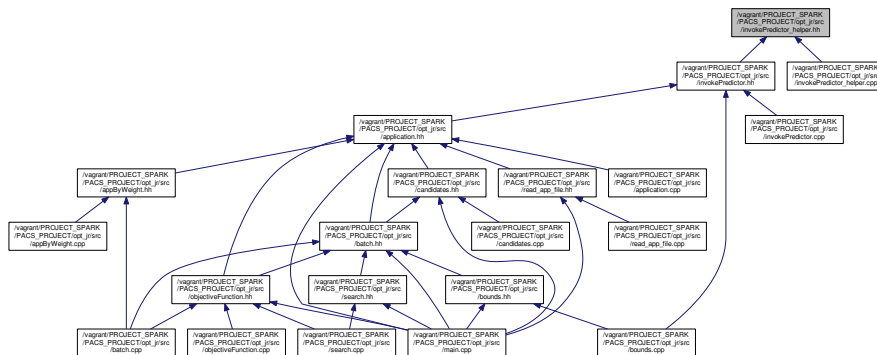


## File Reference

Include dependency graph for invokePredictor\_helper.hh:



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



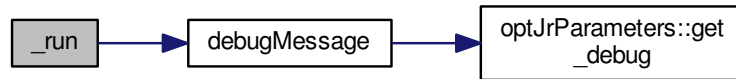
## Functions

- char \* [readFolder](#) (char \*path)
- void [writeFile](#) (const char \*filepath, const char \*data)
- char \* [ls](#) (char \*pattern, [optJrParameters](#) &par)
- char \* [extractRowN](#) (char \*text, int row)
- char \* [replace](#) (char \*text, char \*newLine)
- char \* [readFile](#) (char \*filename)
- char \* [\\_run](#) (char \*cmd, [optJrParameters](#) &par)
- char \* [extractWord](#) (char \*line, int pos)
- char \* [extractRowMatchingPattern](#) (char \*text, char \*pattern)

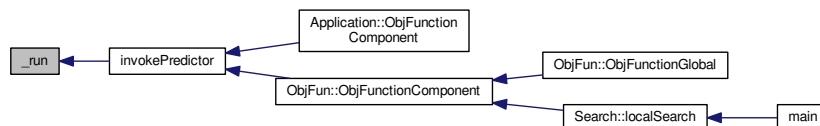
### 4.18.1 Function Documentation

#### 4.18.1.1 `char* _run ( char * cmd, optJrParameters & par )`

Here is the call graph for this function:

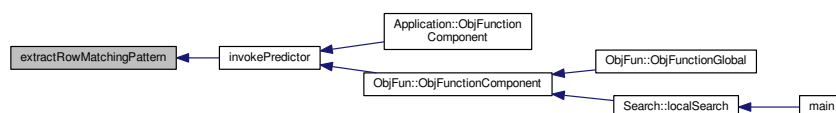


Here is the caller graph for this function:



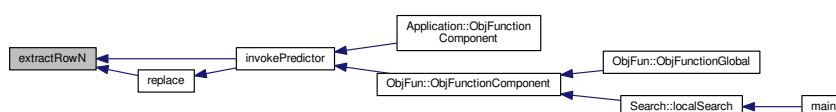
#### 4.18.1.2 `char* extractRowMatchingPattern ( char * text, char * pattern )`

Here is the caller graph for this function:



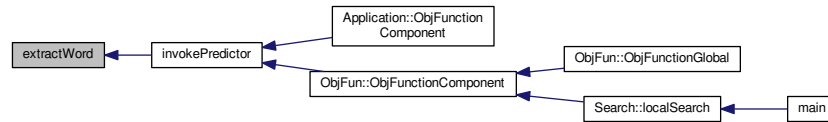
#### 4.18.1.3 `char* extractRowN ( char * text, int row )`

Here is the caller graph for this function:

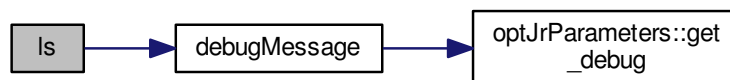


4.18.1.4 `char* extractWord ( char * line, int pos )`

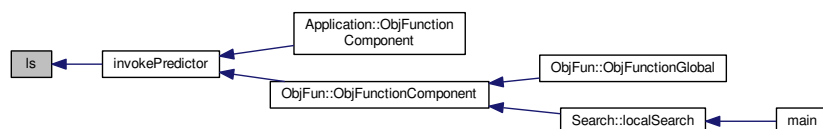
Here is the caller graph for this function:

4.18.1.5 `char* ls ( char * pattern, optJrParameters & par )`

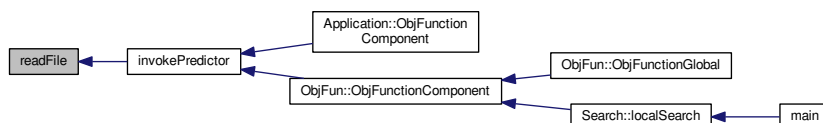
Here is the call graph for this function:



Here is the caller graph for this function:

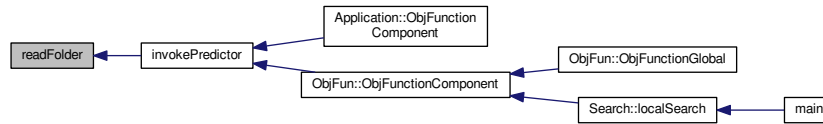
4.18.1.6 `char* readFile ( char * filename )`

Here is the caller graph for this function:



#### 4.18.1.7 `char* readFolder ( char * path )`

Here is the caller graph for this function:

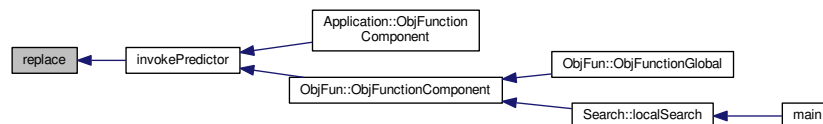


#### 4.18.1.8 `char* replace ( char * text, char * newLine )`

Here is the call graph for this function:

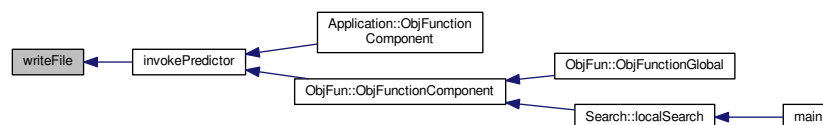


Here is the caller graph for this function:



#### 4.18.1.9 `void writeFile ( const char * filepath, const char * data )`

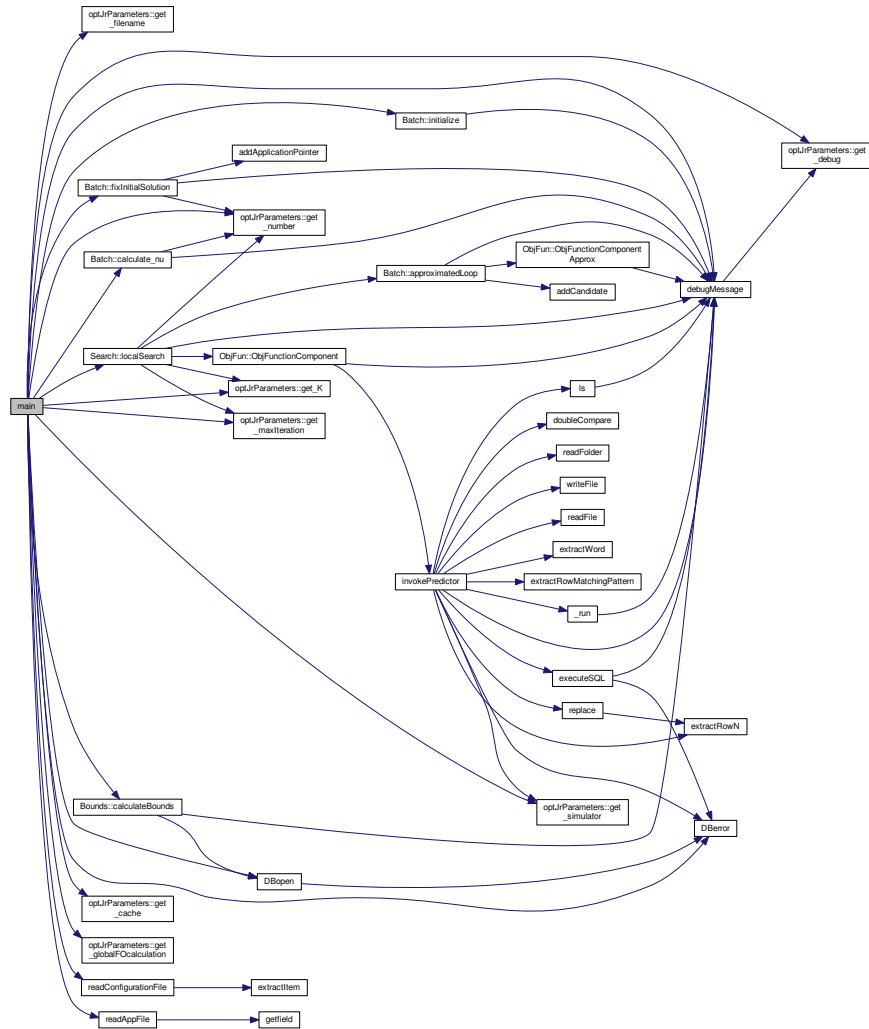
Here is the caller graph for this function:





#### 4.19.1.1 `int main ( int argc, char ** argv )`

Here is the call graph for this function:



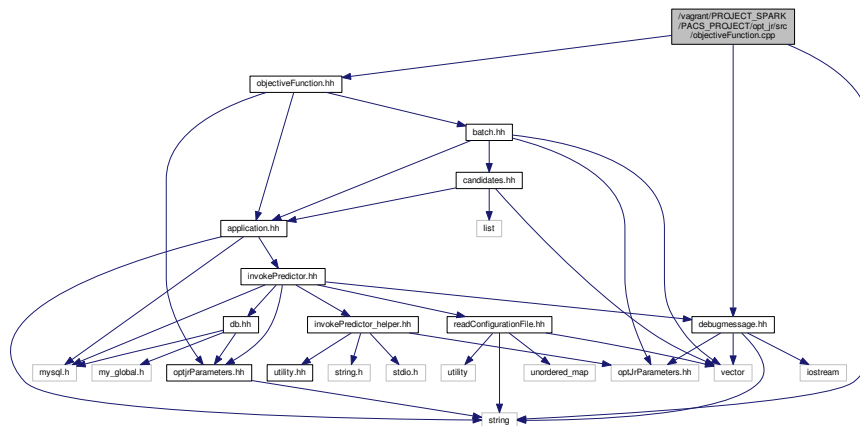
## 4.20 `/vagrant/PROJECT_SPARK/PACS_PROJECT/opt_jr/src/objectiveFunction.cpp` File Reference

```

#include "objectiveFunction.hh"
#include <string>
#include "debugmessage.hh"

```

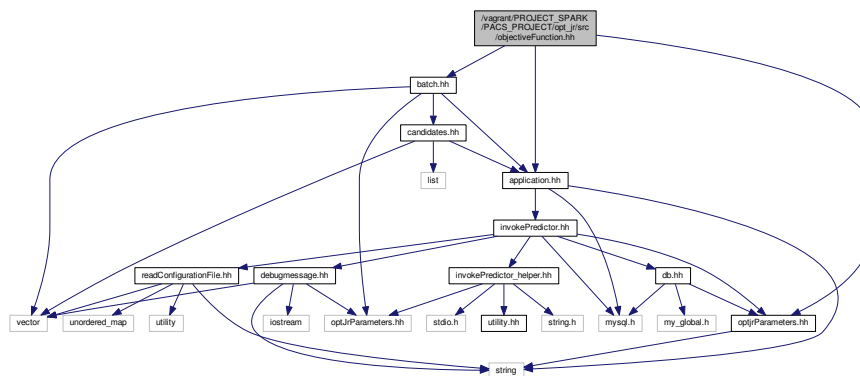
Include dependency graph for objectiveFunction.cpp:



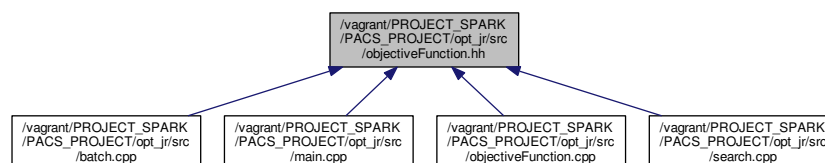
## 4.21 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/objectiveFunction.hh File Reference

```
#include "application.hh"
#include "optJrParameters.hh"
#include "batch.hh"
```

Include dependency graph for objectiveFunction.hh:



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



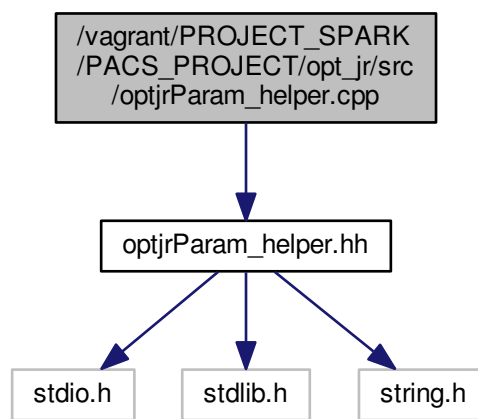
## Classes

- class [ObjFun](#)

## 4.22 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/optjrParam\_helper.cpp File Reference

```
#include "optjrParam_helper.hh"
```

Include dependency graph for `optjrParam_helper.cpp`:



## Functions

- void [Usage](#) ()
- char \* [parseArg](#) (char \*string, char \*gap, int type)

### 4.22.1 Function Documentation

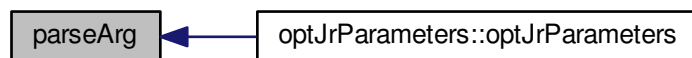
4.22.1.1 char\* `parseArg` ( char \* *string*, char \* *gap*, int *type* )

Here is the call graph for this function:



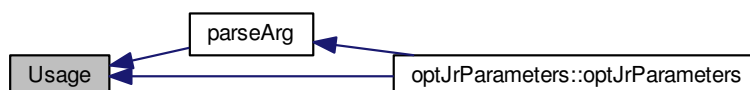


Here is the caller graph for this function:



#### 4.22.1.2 void Usage ( )

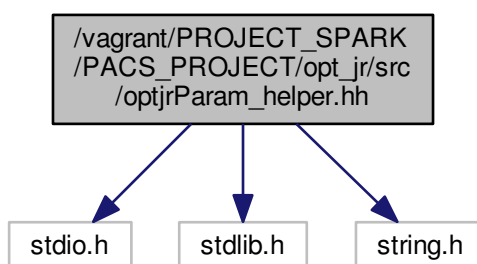
Here is the caller graph for this function:



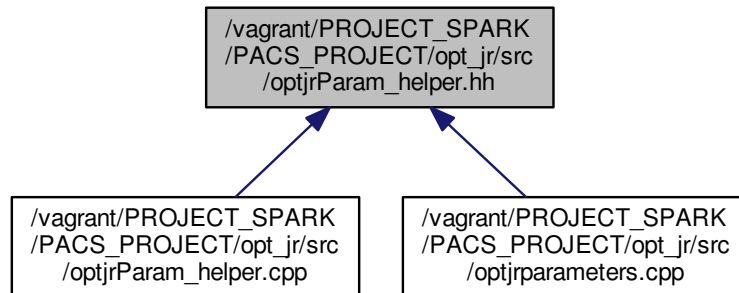
## 4.23 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/optjrParam\_helper.hh File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for `optjrParam_helper.hh`:



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



## Macros

- #define [ARGS](#) 8
- #define [FILENAME](#) "-f="
- #define [NUM\\_N](#) "-n="
- #define [LIST\\_LIMIT](#) "-k="
- #define [DEBUG](#) "-d="
- #define [MAX\\_ITERATIONS](#) "-i="
- #define [SIMULATOR](#) "-s="
- #define [GLOBAL\\_FO\\_CALCULATION](#) "-g"
- #define [NUMBER](#) 0
- #define [STRING](#) 1
- #define [YES\\_NO](#) 2
- #define [NO](#) 0
- #define [YES](#) 1

## Functions

- void [Usage](#) ()
- char \* [parseArg](#) (char \*string, char \*gap, int type)

### 4.23.1 Macro Definition Documentation

#### 4.23.1.1 #define ARGS 8

#### 4.23.1.2 #define DEBUG "-d="

#### 4.23.1.3 #define FILENAME "-f="

#### 4.23.1.4 #define GLOBAL\_FO\_CALCULATION "-g"

#### 4.23.1.5 #define LIST\_LIMIT "-k="

#### 4.23.1.6 #define MAX\_ITERATIONS "-i="

4.23.1.7 `#define NO 0`

4.23.1.8 `#define NUM_N "-n="`

4.23.1.9 `#define NUMBER 0`

4.23.1.10 `#define SIMULATOR "-s="`

4.23.1.11 `#define STRING 1`

4.23.1.12 `#define YES 1`

4.23.1.13 `#define YES_NO 2`

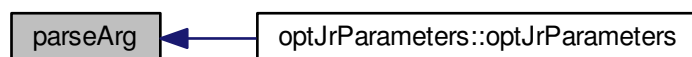
## 4.23.2 Function Documentation

4.23.2.1 `char* parseArg ( char * string, char * gap, int type )`

Here is the call graph for this function:

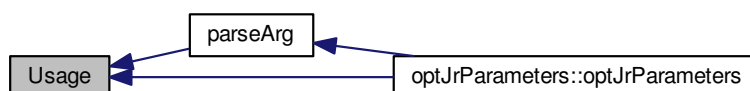


Here is the caller graph for this function:



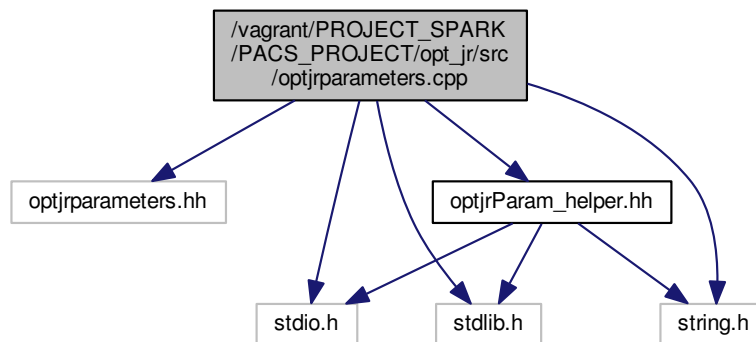
4.23.2.2 `void Usage ( )`

Here is the caller graph for this function:



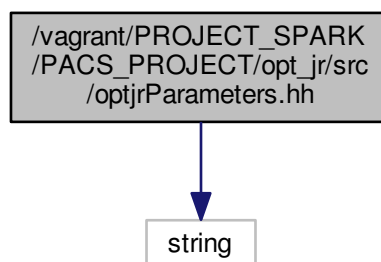
#### 4.24 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/optjrparameters.cpp File Reference

```
#include "optjrparameters.hh"  
#include "optjrParam_helper.hh"  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
Include dependency graph for optjrparameters.cpp:
```



#### 4.25 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/optjrParameters.hh File Reference

```
#include <string>  
Include dependency graph for optjrParameters.hh:
```



[illegible]

- class `optJrParameters`

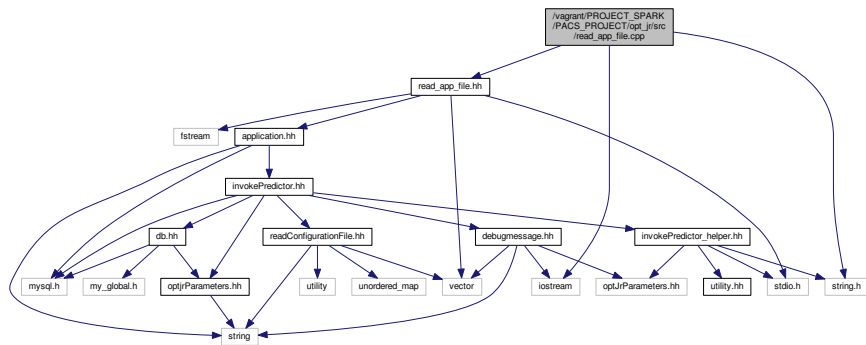
- #define DAGSIM 0
- #define LUNDSTROM 1

#### 4.25.1.1 #define DAGSIM 0

#### 4.25.1.2 #define LUNDSTROM 1

```
#include "read_app_file.hh"
#include <string.h>
#include <iostream>
```

Include dependency graph for `read_app_file.cpp`:



## Macros

- `#define` [MAX\\_APP\\_LENGTH](#) 1024

## Functions

- `char *` [getfield](#) (`char *line`, `int num`)
- `std::vector<` [Application](#) `>` [readAppFile](#) (`FILE *stream`)

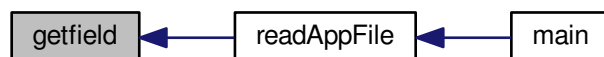
### 4.26.1 Macro Definition Documentation

#### 4.26.1.1 `#define` [MAX\\_APP\\_LENGTH](#) 1024

### 4.26.2 Function Documentation

#### 4.26.2.1 `char*` [getfield](#) ( `char * line`, `int num` )

Here is the caller graph for this function:



4.26.2.2 `std::vector<Application> readAppFile ( FILE * stream )`

Here is the call graph for this function:



Here is the caller graph for this function:



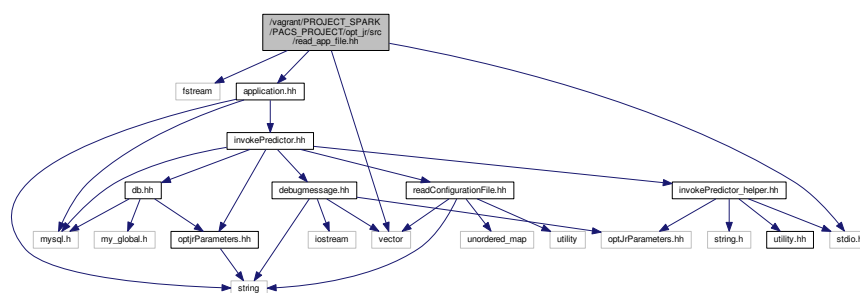
## 4.27 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/read\_app\_file.hh File Reference

```

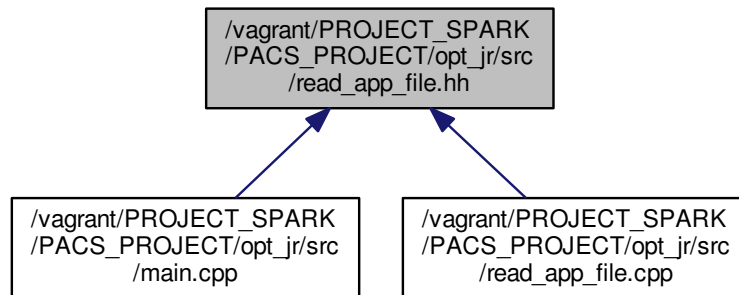
#include <fstream>
#include <stdio.h>
#include <vector>
#include "application.hh"

```

Include dependency graph for `read_app_file.hh`:



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



## Functions

- char \* [getfield](#) (char \*line, int num)
- std::vector< [Application](#) > [readAppFile](#) (FILE \*stream)

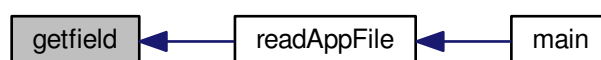
## Variables

- const int [MAX\\_LINE\\_LENGTH](#) = 1024
- const int [\\_SESSION\\_APP\\_ID](#) = 1
- const int [\\_APP\\_ID](#) = 2
- const int [\\_W](#) = 3
- const int [\\_CHI\\_0](#) = 4
- const int [\\_CHI\\_C](#) = 5
- const int [\\_M](#) = 6
- const int [\\_m](#) = 7
- const int [\\_V](#) = 8
- const int [\\_v](#) = 9
- const int [\\_D](#) = 10
- const int [\\_St](#) = 11
- const int [\\_Dsz](#) = 12
- const int [PARAMETERS](#) = 12

### 4.27.1 Function Documentation

#### 4.27.1.1 char\* getfield ( char \* line, int num )

Here is the caller graph for this function:





#### 4.27.1.2 `std::vector<Application> readAppFile ( FILE * stream )`

Here is the call graph for this function:



Here is the caller graph for this function:



### 4.27.2 Variable Documentation

4.27.2.1 `const int _APP_ID = 2`

4.27.2.2 `const int _CHI_0 = 4`

4.27.2.3 `const int _CHI_C = 5`

4.27.2.4 `const int _D = 10`

4.27.2.5 `const int _Dsz = 12`

4.27.2.6 `const int _M = 6`

4.27.2.7 `const int _m = 7`

4.27.2.8 `const int _SESSION_APP_ID = 1`

4.27.2.9 `const int _St = 11`

4.27.2.10 `const int _V = 8`

4.27.2.11 `const int _v = 9`

4.27.2.12 `const int _W = 3`

4.27.2.13 `const int MAX_LINE_LENGTH = 1024`

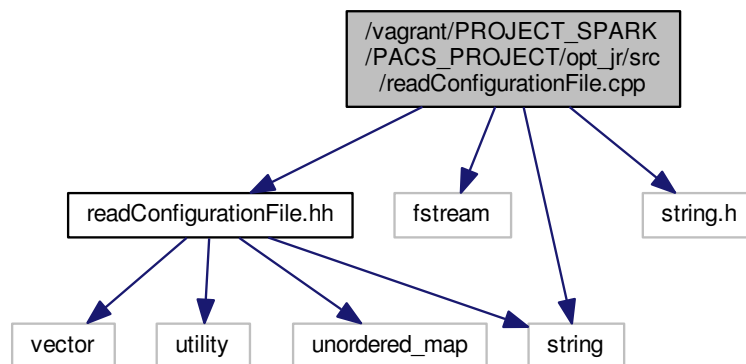
4.27.2.14 `const int PARAMETERS = 12`

## 4.28 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/readConfigurationFile.cpp

### File Reference

```
#include "readConfigurationFile.hh"
#include <fstream>
#include <string>
#include <string.h>
```

Include dependency graph for readConfigurationFile.cpp:



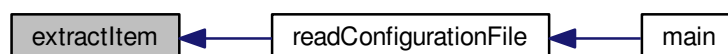
## Functions

- `char * extractItem (char *const string, char *const left, const char *const right)`
- `sConfiguration readConfigurationFile ()`

### 4.28.1 Function Documentation

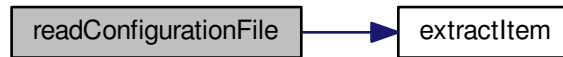
4.28.1.1 `char * extractItem ( char *const string, char *const left, const char *const right )`

Here is the caller graph for this function:



## 4.28.1.2 sConfiguration readConfigurationFile ( )

Here is the call graph for this function:



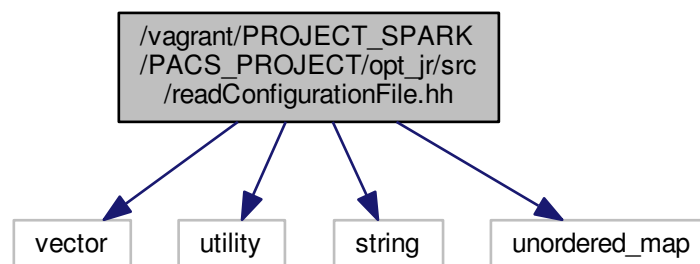
Here is the caller graph for this function:



## 4.29 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/readConfigurationFile.hh File Reference

```
#include <vector>
#include <utility>
#include <string>
#include <unordered_map>
```

Include dependency graph for readConfigurationFile.hh:





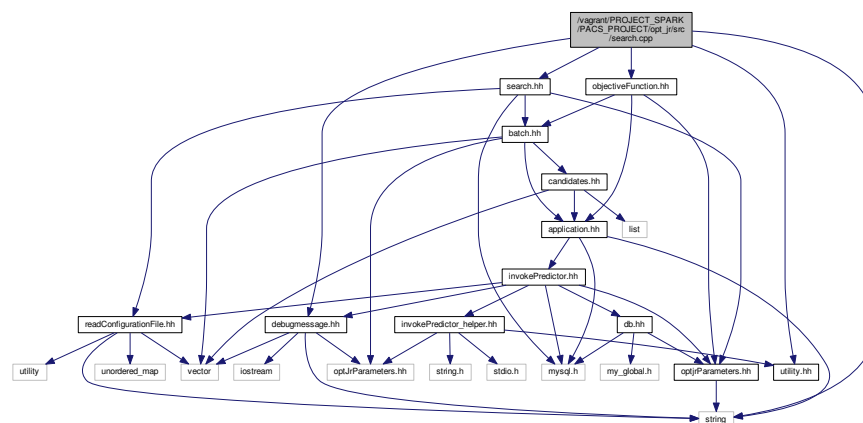
Here is the caller graph for this function:



#### 4.30 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/search.cpp File Reference

```
#include "search.hh"
#include "debugmessage.hh"
#include "utility.hh"
#include "objectiveFunction.hh"
#include <string>
```

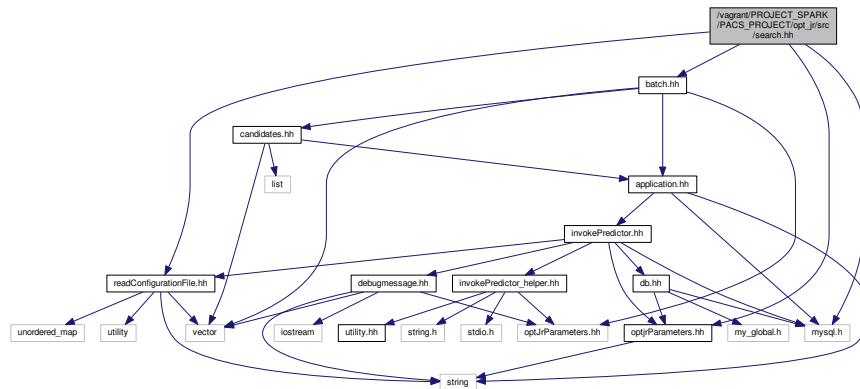
Include dependency graph for search.cpp:



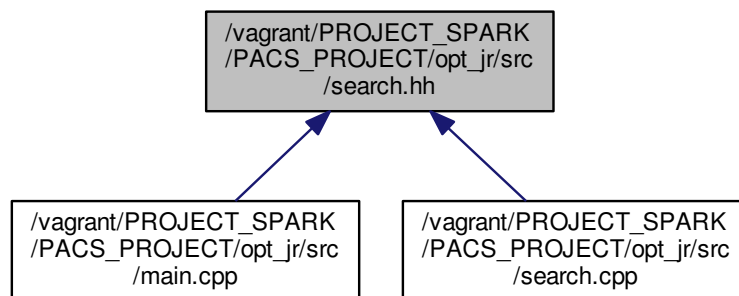
#### 4.31 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/search.hh File Reference

```
#include "readConfigurationFile.hh"
#include "batch.hh"
#include "optJrParameters.hh"
#include <mysql.h>
```

Include dependency graph for search.hh:



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



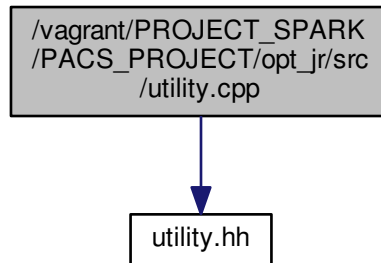
## Classes

- class [Search](#)

## 4.32 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/utility.cpp File Reference

```
#include "utility.hh"
```

Include dependency graph for utility.cpp:



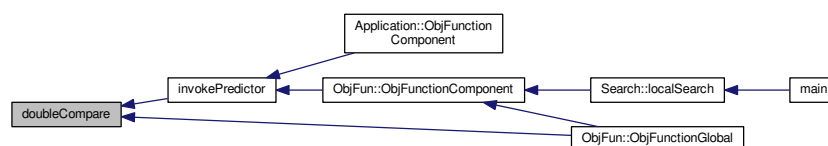
## Functions

- int `doubleCompare` (double `a`, double `b`)

### 4.32.1 Function Documentation

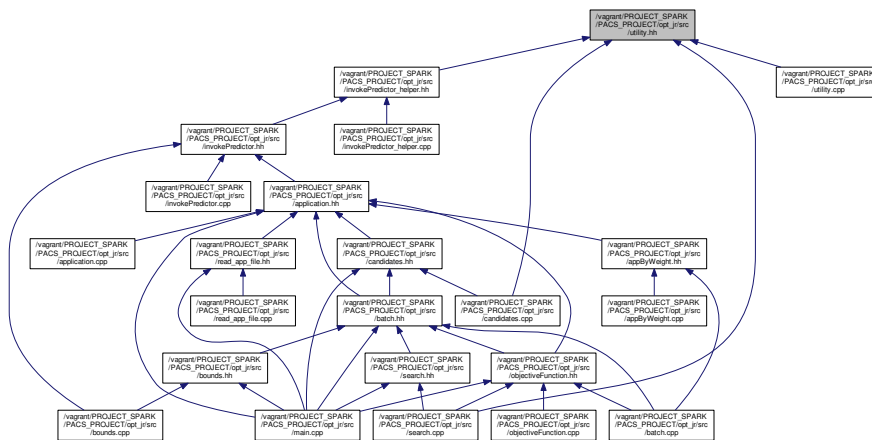
#### 4.32.1.1 int `doubleCompare` ( double `a`, double `b` )

Here is the caller graph for this function:



### 4.33 /vagrant/PROJECT\_SPARK/PACS\_PROJECT/opt\_jr/src/utility.hh File Reference

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



#### Functions

- int [doubleCompare](#) (double a, double b)

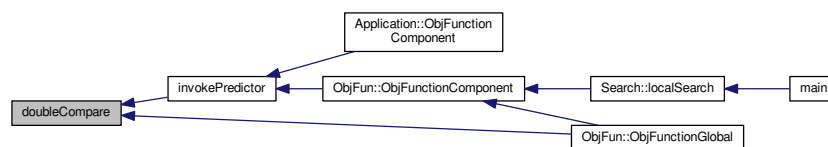
#### Variables

- const double [epsilon](#) = 0.001

#### 4.33.1 Function Documentation

##### 4.33.1.1 int doubleCompare ( double a, double b )

Here is the caller graph for this function:



#### 4.33.2 Variable Documentation

##### 4.33.2.1 const double epsilon = 0.001