

# JGomas Reference Manual

version 0.0.1

Generated by Doxygen 1.4.5

Fri Dec 16 20:13:43 2005



# Contents

<b>1</b>	<b>JGomas Namespace Index</b>	<b>1</b>
1.1	JGomas Package List . . . . .	1
<b>2</b>	<b>JGomas Hierarchical Index</b>	<b>3</b>
2.1	JGomas Class Hierarchy . . . . .	3
<b>3</b>	<b>JGomas Class Index</b>	<b>5</b>
3.1	JGomas Class List . . . . .	5
<b>4</b>	<b>JGomas File Index</b>	<b>7</b>
4.1	JGomas File List . . . . .	7
<b>5</b>	<b>JGomas Namespace Documentation</b>	<b>9</b>
5.1	Package Classes . . . . .	9
5.2	Package java.io . . . . .	10
5.3	Package java.net . . . . .	11
<b>6</b>	<b>JGomas Class Documentation</b>	<b>13</b>
6.1	Classes.CAmmoPack Class Reference . . . . .	13
6.2	Classes.CBase Class Reference . . . . .	15
6.3	Classes.CDinObject Class Reference . . . . .	17
6.4	Classes.CFieldOps Class Reference . . . . .	18
6.5	Classes.CJGomasAgent Class Reference . . . . .	21
6.6	Classes.CManager Class Reference . . . . .	23
6.7	Classes.CMedic Class Reference . . . . .	25
6.8	Classes.CMedicPack Class Reference . . . . .	28
6.9	Classes.CMicroAgent Class Reference . . . . .	30
6.10	Classes.CMobile Class Reference . . . . .	32
6.11	Classes.CObjPack Class Reference . . . . .	34
6.12	Classes.CPack Class Reference . . . . .	36

6.13	Classes.CRegistry Class Reference . . . . .	39
6.14	Classes.CServer Class Reference . . . . .	40
6.15	Classes.CService Class Reference . . . . .	41
6.16	Classes.CSight Class Reference . . . . .	42
6.17	Classes.CSocketThread Class Reference . . . . .	43
6.18	Classes.CSoldier Class Reference . . . . .	45
6.19	Classes.CTask Class Reference . . . . .	47
6.20	Classes.CTerrain Class Reference . . . . .	49
6.21	Classes.CTerrainMap Class Reference . . . . .	50
6.22	Classes.CThreshold Class Reference . . . . .	52
6.23	Classes.CTroop Class Reference . . . . .	56
6.24	Classes.Vector3D Class Reference . . . . .	74
<b>7</b>	<b>JGomas File Documentation</b>	<b>77</b>
7.1	CAmmoPack.java File Reference . . . . .	77
7.2	CFieldOps.java File Reference . . . . .	78
7.3	CJGomasAgent.java File Reference . . . . .	79
7.4	CManager.java File Reference . . . . .	80
7.5	CMedic.java File Reference . . . . .	81
7.6	CMedicPack.java File Reference . . . . .	82
7.7	CMobile.java File Reference . . . . .	83
7.8	CObjPack.java File Reference . . . . .	84
7.9	CPack.java File Reference . . . . .	85
7.10	CRegistry.java File Reference . . . . .	86
7.11	CServer.java File Reference . . . . .	87
7.12	CService.java File Reference . . . . .	88
7.13	CSight.java File Reference . . . . .	89
7.14	CSoldier.java File Reference . . . . .	90
7.15	CTask.java File Reference . . . . .	91
7.16	CTerrainMap.java File Reference . . . . .	92
7.17	CTroop.java File Reference . . . . .	93
7.18	Vector3D.java File Reference . . . . .	94

# Chapter 1

## JGomas Namespace Index

### 1.1 JGomas Package List

Here are the packages with brief descriptions (if available):

<b>Classes</b>	9
<b>java.io</b>	10
<b>java.net</b>	11



## Chapter 2

# JGomas Hierarchical Index

### 2.1 JGomas Class Hierarchy

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

Classes.CBase . . . . .	15
Classes.CDinObject . . . . .	17
Classes.CJGomasAgent . . . . .	21
Classes.CManager . . . . .	23
Classes.CPack . . . . .	36
Classes.CAmmoPack . . . . .	13
Classes.CMedicPack . . . . .	28
Classes.CObjPack . . . . .	34
Classes.CTroop . . . . .	56
Classes.CFieldOps . . . . .	18
Classes.CMedic . . . . .	25
Classes.CSoldier . . . . .	45
Classes.CMicroAgent . . . . .	30
Classes.CMobile . . . . .	32
Classes.CRegistry . . . . .	39
Classes.CServer . . . . .	40
Classes.CService . . . . .	41
Classes.CSight . . . . .	42
Classes.CSocketThread . . . . .	43
Classes.CTask . . . . .	47
Classes.CTerrain . . . . .	49
Classes.CTerrainMap . . . . .	50
Classes.CThreshold . . . . .	52
Classes.Vector3D . . . . .	74





## Chapter 3

# JGomas Class Index

### 3.1 JGomas Class List

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

<b>Classes.CAmmoPack</b>	13
<b>Classes.CBase</b>	15
<b>Classes.CDinObject</b>	17
<b>Classes.CFieldOps</b>	18
<b>Classes.CJGomasAgent</b>	21
<b>Classes.CManager</b>	23
<b>Classes.CMedic</b>	25
<b>Classes.CMedicPack</b>	28
<b>Classes.CMicroAgent</b>	30
<b>Classes.CMobile</b>	32
<b>Classes.CObjPack</b>	34
<b>Classes.CPack</b>	36
<b>Classes.CRegistry</b>	39
<b>Classes.CServer</b>	40
<b>Classes.CService</b>	41
<b>Classes.CSight</b>	42
<b>Classes.CSocketThread</b>	43
<b>Classes.CSoldier</b>	45
<b>Classes.CTask</b>	47
<b>Classes.CTerrain</b>	49
<b>Classes.CTerrainMap</b>	50
<b>Classes.CThreshold</b>	52
<b>Classes.CTroop</b>	56
<b>Classes.Vector3D</b>	74



## Chapter 4

# JGomas File Index

### 4.1 JGomas File List

Here is a list of all files with brief descriptions:

<b>CAmmoPack.java</b>	77
<b>CFieldOps.java</b>	78
<b>CJGomasAgent.java</b>	79
<b>CManager.java</b>	80
<b>CMedic.java</b>	81
<b>CMedicPack.java</b>	82
<b>CMobile.java</b>	83
<b>CObjPack.java</b>	84
<b>CPack.java</b>	85
<b>CRegistry.java</b>	86
<b>CServer.java</b>	87
<b>CService.java</b>	88
<b>CSight.java</b>	89
<b>CSoldier.java</b>	90
<b>CTask.java</b>	91
<b>CTerrainMap.java</b>	92
<b>CTroop.java</b>	93
<b>Vector3D.java</b>	94



## Chapter 5

# JGomas Namespace Documentation

### 5.1 Package Classes

#### Classes

- class **CAmmoPack**
- class **CFieldOps**
- class **CJGomasAgent**
- class **CMicroAgent**
- class **CDinObject**
- class **CManager**
- class **CMedic**
- class **CMedicPack**
- class **CMobile**
- class **CObjPack**
- class **CPack**
- class **CRegistry**
- class **CServer**
- class **CSocketThread**
- class **CService**
- class **CSight**
- class **CSoldier**
- class **CTask**
- class **CBase**
- class **CTerrain**
- class **CTerrainMap**
- class **CThreshold**
- class **CTroop**
- class **Vector3D**

## 5.2 Package java.io

## 5.3 Package java.net





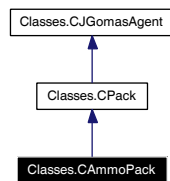
## Chapter 6

# JGomas Class Documentation

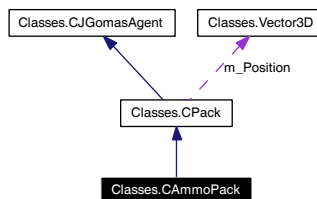
### 6.1 Classes.CAmmoPack Class Reference

Inherits **Classes.CPack**.

Inheritance diagram for Classes.CAmmoPack:



Collaboration diagram for Classes.CAmmoPack:



### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- final void **PerformPackTaken** (String \_sContent)

## 6.1.1 Member Function Documentation

### 6.1.1.1 void **Classes.CAmmoPack.setup ()** [protected]

'setup' method of jade agents.

This method registers all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CPack** (p. 37).

### 6.1.1.2 final void **Classes.CAmmoPack.PerformPackTaken (String *\_sContent*)** [protected]

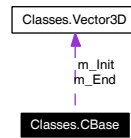
Reimplemented from **Classes.CPack** (p. 37).

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

- **CAmmoPack.java**

## 6.2 Classes.CBase Class Reference

Collaboration diagram for Classes.CBase:



### Package Functions

- **CBase ()**
- double **GetInitX ()**
- double **GetInitY ()**
- double **GetInitZ ()**
- double **GetEndX ()**
- double **GetEndY ()**
- double **GetEndZ ()**

### Package Attributes

- Vector3D **m\_Init**
- Vector3D **m\_End**

### 6.2.1 Constructor & Destructor Documentation

**6.2.1.1** `Classes.CBase.CBase ()` [package]

### 6.2.2 Member Function Documentation

**6.2.2.1** `double Classes.CBase.GetInitX ()` [package]

**6.2.2.2** `double Classes.CBase.GetInitY ()` [package]

**6.2.2.3** `double Classes.CBase.GetInitZ ()` [package]

**6.2.2.4** `double Classes.CBase.GetEndX ()` [package]

**6.2.2.5** `double Classes.CBase.GetEndY ()` [package]

**6.2.2.6** `double Classes.CBase.GetEndZ ()` [package]

### 6.2.3 Member Data Documentation

**6.2.3.1** `Vector3D Classes.CBase.m_Init` [package]

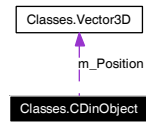
**6.2.3.2** `Vector3D Classes.CBase.m_End` [package]

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

- CTerrainMap.java

## 6.3 Classes.CDinObject Class Reference

Collaboration diagram for Classes.CDinObject:



### Package Functions

- **CDinObject ()**

### Package Attributes

- Integer **m\_id**
- AID **m\_AID**
- Vector3D **m\_Position**
- int **m\_eType**
- boolean **m\_bTaken**
- int **m\_Owner**

### 6.3.1 Constructor & Destructor Documentation

**6.3.1.1** **Classes.CDinObject.CDinObject ()** [package]

### 6.3.2 Member Data Documentation

**6.3.2.1** **Integer Classes.CDinObject.m\_id** [package]

**6.3.2.2** **AID Classes.CDinObject.m\_AID** [package]

**6.3.2.3** **Vector3D Classes.CDinObject.m\_Position** [package]

**6.3.2.4** **int Classes.CDinObject.m\_eType** [package]

**6.3.2.5** **boolean Classes.CDinObject.m\_bTaken** [package]

**6.3.2.6** **int Classes.CDinObject.m\_Owner** [package]

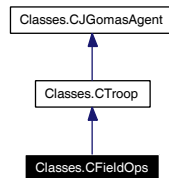
The documentation for this class was generated from the following file:

- **CManager.java**

## 6.4 Classes.CFieldOps Class Reference

Inherits **Classes.CTroop**.

Inheritance diagram for Classes.CFieldOps:



Collaboration diagram for Classes.CFieldOps:



### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- void **SetUpPriorities** ()  
*Definition of priorities for each kind of task.*
- boolean **checkAmmoAction** (String \_sContent)  
*Decides if agent accepts the CFA request.*
- void **PerformTargetReached** (CTask \_CurrentTask)  
*Action to do when this agent reaches the target of current task.*
- final int **CreateAmmoPack** ()  
*Creates ammo packs if possible.*

### Static Protected Attributes

- static int **m\_iPacksDelivered** = 0  
*inner variable used to name packs*

#### 6.4.1 Member Function Documentation

##### 6.4.1.1 void Classes.CFieldOps.setup () [protected]

'setup' method of jade agents.

This method perform actions in common to **CFieldOps**(p. 18) agents (and derived classes) and calls parent's setup.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 60).

#### 6.4.1.2 void Classes.CFieldOps.SetUpPriorities () [protected]

Definition of priorities for each kind of task.

This method can be implemented in CTroop's derived classes to define the task's priorities in agreement to the role of the new class. Priorities must be defined in the array `m_TaskPriority`.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 67).

#### 6.4.1.3 boolean Classes.CFieldOps.checkAmmoAction (String \_sContent) [protected]

Decides if agent accepts the CFA request.

This method is a decision function invoked when a CALL FOR AMMO request has arrived. Parameter `sContent` is the content of message received in CFA responder behaviour as result of a `CallForAmmo` request, so it must be: ( `x` , `y` , `z` ) ( `ammo` ) . By default, the return value is `TRUE`, so agents always accept all CFA requests.

*It's very useful to overload this method.*

**Parameters:**

`_sContent`

**Returns:**

`TRUE`

#### 6.4.1.4 void Classes.CFieldOps.PerformTargetReached (CTask \_CurrentTask) [protected]

Action to do when this agent reaches the target of current task.

This method is called when this agent goes to state `TARGET_REACHED`. If current task is `TASK_GIVE_AMMOPACKS`, agent must give ammo packs, but in other case, it calls to parent's method.

*It's very useful to overload this method.*

**Parameters:**

*\_CurrentTask*

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 68).

**6.4.1.5 final int Classes.CFieldOps.CreateAmmoPack () [protected]**

Creates ammo packs if possible.

This method allows to create medic packs if there is enough power in the agent's power bar.

**Parameters:**

-

**Returns:**

iPacksDelivered: number of medic packs created

**6.4.2 Member Data Documentation****6.4.2.1 int Classes.CFieldOps.m\_iPacksDelivered = 0 [static, protected]**

inner variable used to name packs

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

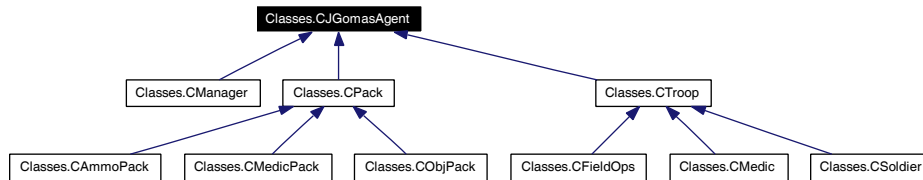
- **CFieldOps.java**



## 6.5 Classes.CJGomasAgent Class Reference

Inherited by **Classes.CManager**, **Classes.CPack**, and **Classes.CTroop**.

Inheritance diagram for Classes.CJGomasAgent:



### Protected Member Functions

- final void **AddService** (CService \_Service)
- void **setup** ()  
*'setup' method of jade agents.*
- void **takeDown** ()  
*'takeDown' method of jade agents.*

### Protected Attributes

- int **m\_ID** = -1
- List **m\_ServiceList**

### 6.5.1 Member Function Documentation

**6.5.1.1 final void Classes.CJGomasAgent.AddService (CService \_Service) [protected]**

**6.5.1.2 void Classes.CJGomasAgent.setup () [protected]**

'setup' method of jade agents.

This method registers all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented in **Classes.CAmmoPack** (p. 14), **Classes.CFieldOps** (p. 18), **Classes.CManager** (p. 23), **Classes.CMedic** (p. 25), **Classes.CMedicPack** (p. 28), **Classes.CObjPack** (p. 34), **Classes.CPack** (p. 37), **Classes.CSoldier** (p. 45), and **Classes.CTroop** (p. 60).

**6.5.1.3 void Classes.CJGomasAgent.takeDown () [protected]**

'takeDown' method of jade agents.

This method deregisters all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented in **Classes.CManager** (p. 24), **Classes.CPack** (p. 37), and **Classes.CTroop** (p. 61).

**6.5.2 Member Data Documentation****6.5.2.1 int Classes.CJGomasAgent.m\_ID = -1 [protected]****6.5.2.2 List Classes.CJGomasAgent.m\_ServiceList [protected]**

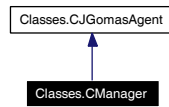
The documentation for this class was generated from the following file:

- **CJGomasAgent.java**

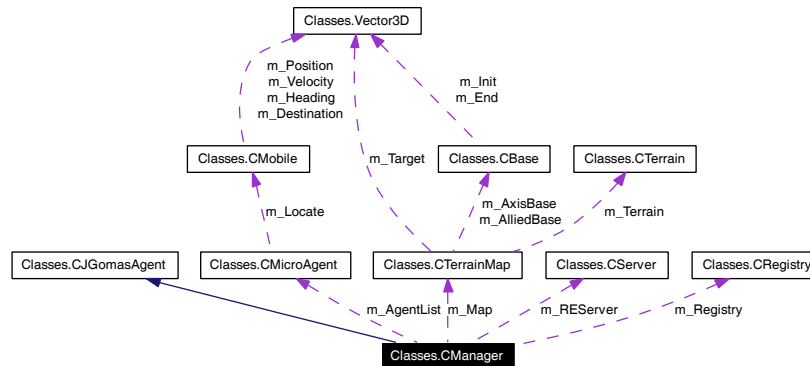
## 6.6 Classes.CManager Class Reference

Inherits **Classes.CJGomasAgent**.

Inheritance diagram for Classes.CManager:



Collaboration diagram for Classes.CManager:



### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- void **takeDown** ()  
*'takeDown' method of jade agents.*

### Static Package Attributes

- static final int **ARG\_PLAYERS** = 0
- static final int **ARG\_MAP\_NAME** = 1
- static final int **ARG\_FPS** = 2
- static final int **ARG\_MATCH\_TIME** = 3

#### 6.6.1 Member Function Documentation

##### 6.6.1.1 void **Classes.CManager.setup** () [protected]

'setup' method of jade agents.

This method registers all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CJGomasAgent** (p. 21).

**6.6.1.2 void Classes.CManager.takeDown () [protected]**

'takeDown' method of jade agents.

This method deregisters all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CJGomasAgent** (p. 22).

**6.6.2 Member Data Documentation**

**6.6.2.1 final int Classes.CManager.ARG\_PLAYERS = 0 [static, package]**

**6.6.2.2 final int Classes.CManager.ARG\_MAP\_NAME = 1 [static, package]**

**6.6.2.3 final int Classes.CManager.ARG\_FPS = 2 [static, package]**

**6.6.2.4 final int Classes.CManager.ARG\_MATCH\_TIME = 3 [static, package]**

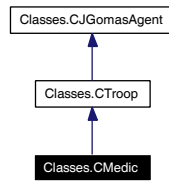
The documentation for this class was generated from the following file:

- **CManager.java**

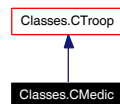
## 6.7 Classes.CMedic Class Reference

Inherits **Classes.CTroop**.

Inheritance diagram for Classes.CMedic:



Collaboration diagram for Classes.CMedic:



### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- void **SetUpPriorities** ()  
*Definition of priorities for each kind of task.*
- boolean **checkMedicAction** (String \_sContent)  
*Decides if agent accepts the CFM request.*
- void **PerformTargetReached** (CTask \_CurrentTask)  
*Action to do when this agent reaches the target of current task.*
- final int **CreateMedicPack** ()  
*Creates medic packs if possible.*

### Static Protected Attributes

- static int **m\_iPacksDelivered** = 0  
*inner variable used to name packs*

#### 6.7.1 Member Function Documentation

##### 6.7.1.1 void Classes.CMedic.setup () [protected]

'setup' method of jade agents.

This method perform actions in common to **CMedic**(p. 25) agents (and derived classes) and calls parent's setup.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 60).

### 6.7.1.2 void Classes.CMedic.SetupPriorities () [protected]

Definition of priorities for each kind of task.

This method can be implemented in CTroop's derived classes to define the task's priorities in agreement to the role of the new class. Priorities must be defined in the array `m_TaskPriority`.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 67).

### 6.7.1.3 boolean Classes.CMedic.checkMedicAction (String \_sContent) [protected]

Decides if agent accepts the CFM request.

This method is a decision function invoked when a CALL FOR MEDIC request has arrived. Parameter `sContent` is the content of message received in CFM responder behaviour as result of a `CallForMedic` request, so it must be: `( x , y , z ) ( health )`. By default, the return value is `TRUE`, so agents always accept all CFM requests.

*It's very useful to overload this method.*

**Parameters:**

`_sContent`

**Returns:**

`TRUE`

### 6.7.1.4 void Classes.CMedic.PerformTargetReached (CTask \_CurrentTask) [protected]

Action to do when this agent reaches the target of current task.

This method is called when this agent goes to state `TARGET_REACHED`. If current task is `TASK_GIVE_MEDICPAKS`, agent must give medic packs, but in other case, it calls to parent's method.

*It's very useful to overload this method.*

**Parameters:**

*\_CurrentTask*

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 68).

**6.7.1.5 final int Classes.CMedic.CreateMedicPack () [protected]**

Creates medic packs if possible.

This method allows to create medic packs if there is enough power in the agent's power bar.

**Parameters:**

-

**Returns:**

iPacksDelivered: number of medic packs created

**6.7.2 Member Data Documentation****6.7.2.1 int Classes.CMedic.m\_iPacksDelivered = 0 [static, protected]**

inner variable used to name packs

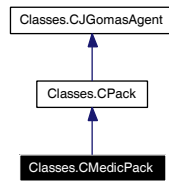
The documentation for this class was generated from the following file:

- **CMedic.java**

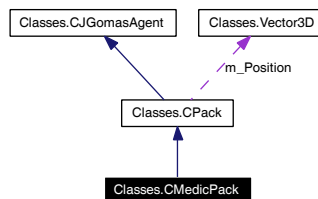
## 6.8 Classes.CMedicPack Class Reference

Inherits **Classes.CPack**.

Inheritance diagram for Classes.CMedicPack:



Collaboration diagram for Classes.CMedicPack:



### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- final void **PerformPackTaken** (String *\_sContent*)

### 6.8.1 Member Function Documentation

#### 6.8.1.1 void Classes.CMedicPack.setup () [protected]

'setup' method of jade agents.

This method registers all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CPack** (p. 37).

#### 6.8.1.2 final void Classes.CMedicPack.PerformPackTaken (String *\_sContent*) [protected]

Reimplemented from **Classes.CPack** (p. 37).

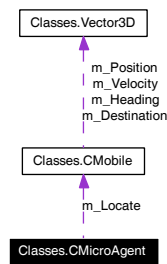
The documentation for this class was generated from the following file:



- CMedicPack.java

## 6.9 Classes.CMicroAgent Class Reference

Collaboration diagram for Classes.CMicroAgent:



### Package Functions

- **CMicroAgent ()**

### Package Attributes

- **AID m\_AID**
- **String m\_sName**
- **int m\_id**
- **int m\_eTeam**
- **int m\_eType**
- **int m\_iHealth**
- **int m\_iAmmo**
- **CMobile m\_Locate**
- **boolean m\_bCarryingObjective**
- **boolean m\_bShooting**

## 6.9.1 Constructor & Destructor Documentation

**6.9.1.1** `Classes.CMicroAgent.CMicroAgent ()` [package]

## 6.9.2 Member Data Documentation

**6.9.2.1** `AID Classes.CMicroAgent.m_AID` [package]

**6.9.2.2** `String Classes.CMicroAgent.m_sName` [package]

**6.9.2.3** `int Classes.CMicroAgent.m_id` [package]

**6.9.2.4** `int Classes.CMicroAgent.m_eTeam` [package]

**6.9.2.5** `int Classes.CMicroAgent.m_eType` [package]

**6.9.2.6** `int Classes.CMicroAgent.m_iHealth` [package]

**6.9.2.7** `int Classes.CMicroAgent.m_iAmmo` [package]

**6.9.2.8** `CMobile Classes.CMicroAgent.m_Locate` [package]

**6.9.2.9** `boolean Classes.CMicroAgent.m_bCarryingObjective` [package]

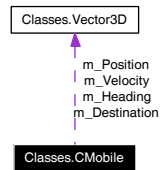
**6.9.2.10** `boolean Classes.CMicroAgent.m_bShooting` [package]

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

- `CManager.java`

## 6.10 Classes.CMobile Class Reference

Collaboration diagram for Classes.CMobile:



### Package Functions

- **CMobile ()**
- final void **SetSize** (int \_MaxX, int \_MaxZ)
- final boolean **CalculatePosition** ()
- final void **CalculateNewOrientation** ()
- final void **CalculateNewDestination** (int \_iRadiusX, int \_iRadiusY)

### Package Attributes

- **Vector3D m\_Position**
- **Vector3D m\_Destination**
- **Vector3D m\_Velocity**
- **Vector3D m\_Heading**
- double **m\_dViewRadius**
- double **m\_dAngle**
- long **m\_lLastTime**

## 6.10.1 Constructor & Destructor Documentation

**6.10.1.1** `Classes.CMobile.CMobile ()` [package]

## 6.10.2 Member Function Documentation

**6.10.2.1** `final void Classes.CMobile.SetSize (int _MaxX, int _MaxZ)` [package]

**6.10.2.2** `final boolean Classes.CMobile.CalculatePosition ()` [package]

**6.10.2.3** `final void Classes.CMobile.CalculateNewOrientation ()` [package]

**6.10.2.4** `final void Classes.CMobile.CalculateNewDestination (int _iRadiusX, int _iRadiusY)`  
[package]

## 6.10.3 Member Data Documentation

**6.10.3.1** `Vector3D Classes.CMobile.m_Position` [package]

**6.10.3.2** `Vector3D Classes.CMobile.m_Destination` [package]

**6.10.3.3** `Vector3D Classes.CMobile.m_Velocity` [package]

**6.10.3.4** `Vector3D Classes.CMobile.m_Heading` [package]

**6.10.3.5** `double Classes.CMobile.m_dViewRadius` [package]

**6.10.3.6** `double Classes.CMobile.m_dAngle` [package]

**6.10.3.7** `long Classes.CMobile.m_lLastTime` [package]

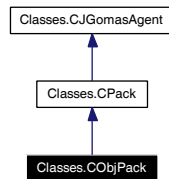
The documentation for this class was generated from the following file:

- CMobile.java

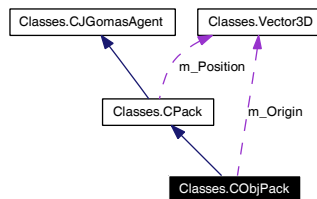
## 6.11 Classes.CObjPack Class Reference

Inherits **Classes.CPack**.

Inheritance diagram for Classes.CObjPack:



Collaboration diagram for Classes.CObjPack:



### Protected Member Functions

- final void **SetTaken** (boolean \_bTaken)
- void **setup** ()  
*'setup' method of jade agents.*
- final void **PerformPackTaken** (String \_sContent)

#### 6.11.1 Member Function Documentation

**6.11.1.1 final void Classes.CObjPack.SetTaken (boolean *\_bTaken*)** [protected]

**6.11.1.2 void Classes.CObjPack.setup ()** [protected]

'setup' method of jade agents.

This method registers all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CPack** (p. 37).

**6.11.1.3** final void **Classes.CObjPack.PerformPackTaken** (String *\_sContent*) [protected]

Reimplemented from **Classes.CPack** (p. 37).

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

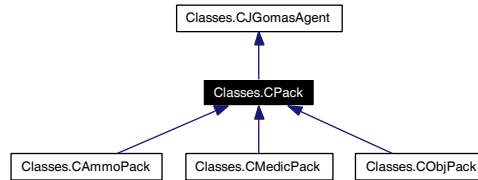
- **CObjPack.java**

## 6.12 Classes.CPack Class Reference

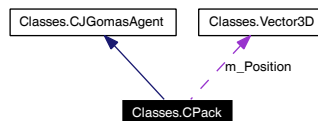
Inherits **Classes.CJGomasAgent**.

Inherited by **Classes.CAmmoPack**, **Classes.CMedicPack**, and **Classes.CObjPack**.

Inheritance diagram for Classes.CPack:



Collaboration diagram for Classes.CPack:



### Static Public Attributes

- static final int **PACK\_NONE** = 1000
- static final int **PACK\_MEDICPACK** = 1001
- static final int **PACK\_AMMOPACK** = 1002
- static final int **PACK\_OBJPACK** = 1003

### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- final void **takeDown** ()  
*'takeDown' method of jade agents.*
- void **PerformPackTaken** (String \_sContent)

### Protected Attributes

- int **m\_eType**
- **Vector3D m\_Position**
- AID **m\_Manager** = null



## 6.12.1 Member Function Documentation

### 6.12.1.1 void Classes.CPack.setup () [protected]

'setup' method of jade agents.

This method registers all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CJGomasAgent** (p. 21).

Reimplemented in **Classes.CAmmoPack** (p. 14), **Classes.CMedicPack** (p. 28), and **Classes.CObjPack** (p. 34).

### 6.12.1.2 final void Classes.CPack.takeDown () [protected]

'takeDown' method of jade agents.

This method deregisters all services offered by an agent.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CJGomasAgent** (p. 22).

### 6.12.1.3 void Classes.CPack.PerformPackTaken (String *\_sContent*) [protected]

Reimplemented in **Classes.CAmmoPack** (p. 14), **Classes.CMedicPack** (p. 28), and **Classes.CObjPack** (p. 35).

## 6.12.2 Member Data Documentation

**6.12.2.1** `final int Classes.CPack.PACK_NONE = 1000` [static]

**6.12.2.2** `final int Classes.CPack.PACK_MEDICPACK = 1001` [static]

**6.12.2.3** `final int Classes.CPack.PACK_AMMOPACK = 1002` [static]

**6.12.2.4** `final int Classes.CPack.PACK_OBJPACK = 1003` [static]

**6.12.2.5** `int Classes.CPack.m_eType` [protected]

**6.12.2.6** `Vector3D Classes.CPack.m_Position` [protected]

**6.12.2.7** `AID Classes.CPack.m_Manager = null` [protected]

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

- **CPack.java**

## 6.13 Classes.CRegistry Class Reference

### Static Public Attributes

- static final int **MAX\_TOTAL\_SERVICES** = 100

### Package Functions

- **CRegistry** ()
- **CService RegisterService** (String *\_sServiceType*)
- final **CService RegisterService** (String *\_sServiceType*, boolean *\_bKeyCode*)

### 6.13.1 Constructor & Destructor Documentation

**6.13.1.1** **Classes.CRegistry.CRegistry** () [package]

### 6.13.2 Member Function Documentation

**6.13.2.1** **CService Classes.CRegistry.RegisterService** (String *\_sServiceType*) [package]

**6.13.2.2** final **CService Classes.CRegistry.RegisterService** (String *\_sServiceType*, boolean *\_bKeyCode*) [package]

### 6.13.3 Member Data Documentation

**6.13.3.1** final int **Classes.CRegistry.MAX\_TOTAL\_SERVICES** = 100 [static]

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

- **CRegistry.java**

## 6.14 Classes.CServer Class Reference

### Public Member Functions

- void **run** ()

### Static Protected Attributes

- static ArrayList **m\_ConnectionList** = null

### Package Functions

- **CServer** (String \_sMapName)

### Static Package Attributes

- static final int **SERVER\_PORT** = 8001

### 6.14.1 Constructor & Destructor Documentation

6.14.1.1 **Classes.CServer.CServer** (String *\_sMapName*) [package]

### 6.14.2 Member Function Documentation

6.14.2.1 void **Classes.CServer.run** ()

### 6.14.3 Member Data Documentation

6.14.3.1 final int **Classes.CServer.SERVER\_PORT** = 8001 [static, package]

6.14.3.2 ArrayList **Classes.CServer.m\_ConnectionList** = null [static, protected]

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

- **CServer.java**

## 6.15 Classes.CService Class Reference

### Package Attributes

- String **m\_sDFType**
- String **m\_sDFName**

### 6.15.1 Member Data Documentation

**6.15.1.1 String Classes.CService.m\_sDFType** [package]

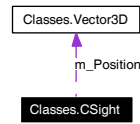
**6.15.1.2 String Classes.CService.m\_sDFName** [package]

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

- **CService.java**

## 6.16 Classes.CSight Class Reference

Collaboration diagram for Classes.CSight:



### Protected Attributes

- `int m_id`
- `int m_eType`
- `Vector3D m_Position`
- `double m_dDistance`
- `double m_dAngle`
- `int m_iHealth`

### Package Functions

- `CSight ()`

### 6.16.1 Constructor & Destructor Documentation

**6.16.1.1** `Classes.CSight.CSight ()` [package]

### 6.16.2 Member Data Documentation

**6.16.2.1** `int Classes.CSight.m_id` [protected]

**6.16.2.2** `int Classes.CSight.m_eType` [protected]

**6.16.2.3** `Vector3D Classes.CSight.m_Position` [protected]

**6.16.2.4** `double Classes.CSight.m_dDistance` [protected]

**6.16.2.5** `double Classes.CSight.m_dAngle` [protected]

**6.16.2.6** `int Classes.CSight.m_iHealth` [protected]

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

- `CSight.java`

## 6.17 Classes.CSocketThread Class Reference

### Public Member Functions

- void **run** ()
- void **SendMsgToRenderEngine** (int \_msgType, String \_msg)

### Public Attributes

- InputStream **in**
- OutputStream **out**
- BufferedReader **reader**
- PrintWriter **writer**

### Static Public Attributes

- static final int **TCP\_COM** = 0
- static final int **TCP\_AGL** = 1
- static final int **TCP\_MAP** = 2
- static final int **TCP\_TIME** = 3

### Package Functions

- **CSocketThread** (Socket socket, String \_sMapName)

### Package Attributes

- Socket **m\_Socket**

## 6.17.1 Constructor & Destructor Documentation

**6.17.1.1** `Classes.CSocketThread.CSocketThread (Socket socket, String _sMapName)`  
[package]

## 6.17.2 Member Function Documentation

**6.17.2.1** `void Classes.CSocketThread.run ()`

**6.17.2.2** `void Classes.CSocketThread.SendMsgToRenderEngine (int _msgType, String _msg)`

## 6.17.3 Member Data Documentation

**6.17.3.1** `final int Classes.CSocketThread.TCP_COM = 0` [static]

**6.17.3.2** `final int Classes.CSocketThread.TCP_AGL = 1` [static]

**6.17.3.3** `final int Classes.CSocketThread.TCP_MAP = 2` [static]

**6.17.3.4** `final int Classes.CSocketThread.TCP_TIME = 3` [static]

**6.17.3.5** `InputStream Classes.CSocketThread.in`

**6.17.3.6** `OutputStream Classes.CSocketThread.out`

**6.17.3.7** `BufferedReader Classes.CSocketThread.reader`

**6.17.3.8** `PrintWriter Classes.CSocketThread.writer`

**6.17.3.9** `Socket Classes.CSocketThread.m_Socket` [package]

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

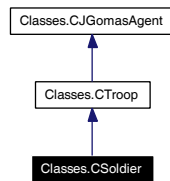
- `CServer.java`



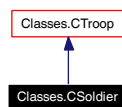
## 6.18 Classes.CSoldier Class Reference

Inherits **Classes.CTroop**.

Inheritance diagram for Classes.CSoldier:



Collaboration diagram for Classes.CSoldier:



### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- void **SetUpPriorities** ()  
*Definition of priorities for each kind of task.*
- boolean **checkBackupAction** (String \_sContent)  
*Decides if agent accepts the CFB request.*

### 6.18.1 Member Function Documentation

#### 6.18.1.1 void Classes.CSoldier.setup () [protected]

'setup' method of jade agents.

This method perform actions in common to **CSoldier**(p. 45) agents (and derived classes) and calls parent's setup.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 60).

**6.18.1.2 void Classes.CSoldier.SetupPriorities () [protected]**

Definition of priorities for each kind of task.

This method can be implemented in CTroop's derived classes to define the task's priorities in agreement to the role of the new class. Priorities must be defined in the array `m_TaskPriority`.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CTroop** (p. 67).

**6.18.1.3 boolean Classes.CSoldier.checkBackupAction (String \_sContent) [protected]**

Decides if agent accepts the CFB request.

This method is a decision function invoked when a CALL FOR BACKUP request has arrived. Parameter `sContent` is the content of message received in CFB responder behaviour as result of a `CallForBackup` request, so it must be: `( x , y , z ) ( SoldiersCount )`. By default, the return value is `TRUE`, so agents always accepts all CFB requests.

*It's very useful to overload this method.*

**Parameters:**

*`_sContent`*

**Returns:**

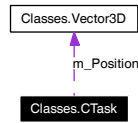
`TRUE`

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

- **CSoldier.java**

## 6.19 Classes.CTask Class Reference

Collaboration diagram for Classes.CTask:



### Static Public Attributes

- static final int **TASK\_NONE** = 0
- static final int **TASK\_GIVE\_MEDICPAKS** = 1
- static final int **TASK\_GIVE\_AMMOPACKS** = 2
- static final int **TASK\_GIVE\_BACKUP** = 3
- static final int **TASK\_GET\_OBJECTIVE** = 4
- static final int **TASK\_ATTACK** = 6
- static final int **TASK\_RUN\_AWAY** = 7
- static final int **TASK\_GOTO\_POSITION** = 8
- static final int **TASK\_PATROLLING** = 9
- static final int **TASK\_WALKING\_PATH** = 10
- static final int **MAX\_TASK** = 100

### Package Functions

- **CTask ()**

### Package Attributes

- Integer **m\_id**
- AID **m\_AID**
- int **m\_iType**
- int **m\_iPriority**
- **Vector3D m\_Position**
- long **m\_StampTime**
- int **m\_iData**
- float **m\_fData**
- int **m\_iPacksDelivered**
- Object **m\_ObjPointer**
- boolean **m\_bErasable**

## 6.19.1 Constructor & Destructor Documentation

6.19.1.1 `Classes.CTask.CTask ()` [package]

## 6.19.2 Member Data Documentation

6.19.2.1 `final int Classes.CTask.TASK_NONE = 0` [static]

6.19.2.2 `final int Classes.CTask.TASK_GIVE_MEDICPAKS = 1` [static]

6.19.2.3 `final int Classes.CTask.TASK_GIVE_AMMOPACKS = 2` [static]

6.19.2.4 `final int Classes.CTask.TASK_GIVE_BACKUP = 3` [static]

6.19.2.5 `final int Classes.CTask.TASK_GET_OBJECTIVE = 4` [static]

6.19.2.6 `final int Classes.CTask.TASK_ATTACK = 6` [static]

6.19.2.7 `final int Classes.CTask.TASK_RUN_AWAY = 7` [static]

6.19.2.8 `final int Classes.CTask.TASK_GOTO_POSITION = 8` [static]

6.19.2.9 `final int Classes.CTask.TASK_PATROLLING = 9` [static]

6.19.2.10 `final int Classes.CTask.TASK_WALKING_PATH = 10` [static]

6.19.2.11 `final int Classes.CTask.MAX_TASK = 100` [static]

6.19.2.12 `Integer Classes.CTask.m_id` [package]

6.19.2.13 `AID Classes.CTask.m_AID` [package]

6.19.2.14 `int Classes.CTask.m_iType` [package]

6.19.2.15 `int Classes.CTask.m_iPriority` [package]

6.19.2.16 `Vector3D Classes.CTask.m_Position` [package]

6.19.2.17 `long Classes.CTask.m_StampTime` [package]

6.19.2.18 `int Classes.CTask.m_iData` [package]

6.19.2.19 `float Classes.CTask.m_fData` [package]

6.19.2.20 `int Classes.CTask.m_iPacksDelivered` [package]

6.19.2.21 `Object Classes.CTask.m_ObjPointer` [package]

6.19.2.22 `boolean Classes.CTask.m_bErasable` [package]

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

- `CTask.java`

## 6.20 Classes.CTerrain Class Reference

### Package Attributes

- double **m\_dHeight**
- boolean **m\_bCanWalk**
- int **m\_iCost**

### 6.20.1 Member Data Documentation

**6.20.1.1 double Classes.CTerrain.m\_dHeight** [package]

**6.20.1.2 boolean Classes.CTerrain.m\_bCanWalk** [package]

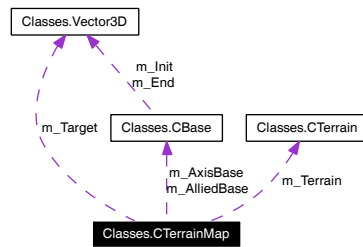
**6.20.1.3 int Classes.CTerrain.m\_iCost** [package]

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

- **CTerrainMap.java**

## 6.21 Classes.CTerrainMap Class Reference

Collaboration diagram for Classes.CTerrainMap:



### Protected Member Functions

- final void **LoadMap** (String \_sMainFile)

### Protected Attributes

- CBase **m\_AlliedBase**
- CBase **m\_AxisBase**

### Package Functions

- **CTerrainMap** ()
- final int **GetSizeX** ()
- final int **GetSizeZ** ()
- final double **GetTargetX** ()
- final double **GetTargetY** ()
- final double **GetTargetZ** ()
- final boolean **CanWalk** (int x, int z)

## 6.21.1 Constructor & Destructor Documentation

6.21.1.1 `Classes.CTerrainMap.CTerrainMap ()` [package]

## 6.21.2 Member Function Documentation

6.21.2.1 `final int Classes.CTerrainMap.GetSizeX ()` [package]

6.21.2.2 `final int Classes.CTerrainMap.GetSizeZ ()` [package]

6.21.2.3 `final double Classes.CTerrainMap.GetTargetX ()` [package]

6.21.2.4 `final double Classes.CTerrainMap.GetTargetY ()` [package]

6.21.2.5 `final double Classes.CTerrainMap.GetTargetZ ()` [package]

6.21.2.6 `final boolean Classes.CTerrainMap.CanWalk (int x, int z)` [package]

6.21.2.7 `final void Classes.CTerrainMap.LoadMap (String _sMainFile)` [protected]

## 6.21.3 Member Data Documentation

6.21.3.1 `CBase Classes.CTerrainMap.m_AlliedBase` [protected]

6.21.3.2 `CBase Classes.CTerrainMap.m_AxisBase` [protected]

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

- `CTerrainMap.java`

## 6.22 Classes.CThreshold Class Reference

### Protected Member Functions

- final int **GetHealth** ()  
*Get the stablished limit of health.*
- final int **GetAmmo** ()  
*Get the stablished limit of ammunition.*
- final int **GetAim** ()  
*Get the stablished number of times that the agent must aim the enemy before to shoot.*
- final int **GetShot** ()  
*Get the stablished number of times that the agent must shoot consecutively before doing other action.*
- final int **GetLook** ()  
*Get the stablished number of times (cycles) that the agent must wait (moving blindly) before looking again.*
- final void **SetHealth** (int \_iHealth)  
*Stablish the limit of health.*
- final void **SetAmmo** (int \_iAmmo)  
*Stablish the limit of ammunition.*
- final void **SetAim** (int \_iAim)  
*Stablish the number of times that the agent must aim the enemy before to shoot.*
- final void **SetShot** (int \_iShot)  
*Stablish the number of times that the agent must shoot consecutively before doing other action.*
- final void **SetLook** (int \_iLook)  
*Stablish the number of times (cycles) that the agent must wait (moving blindly) before looking again.*

### Package Functions

- **CThreshold** ()

### 6.22.1 Constructor & Destructor Documentation

#### 6.22.1.1 Classes.CThreshold.CThreshold () [package]

### 6.22.2 Member Function Documentation

#### 6.22.2.1 final int Classes.CThreshold.GetHealth () [protected]

Get the stablished limit of health.

Agent can perform some actions if its health is lower than this value.



**Parameters:**

-

**Returns:**

m\_iHealth: current threshold for health

**6.22.2.2 final int Classes.CThreshold.GetAmmo () [protected]**

Get the stablished limit of ammunition.

Agent can perform some actions if its ammo is lower than this value.

**Parameters:**

-

**Returns:**

m\_iAmmo: current threshold for ammo

**6.22.2.3 final int Classes.CThreshold.GetAim () [protected]**

Get the stablished number of times that the agent must aim the enemy before to shoot.

**Parameters:**

-

**Returns:**

m\_iAim: current threshold for aim

**6.22.2.4 final int Classes.CThreshold.GetShot () [protected]**

Get the stablished number of times that the agent must shoot consecutively before doing other action.

**Parameters:**

-

**Returns:**

m\_iShot: current threshold for shot

**6.22.2.5 final int Classes.CThreshold.GetLook () [protected]**

Get the stablished number of times (cycles) that the agent must wait (moving blindly) before looking again.

**Parameters:**

-

**Returns:**

m\_iLook: current threshold for look

**6.22.2.6 final void Classes.CThreshold.SetHealth (int *\_iHealth*) [protected]**

Stablish the limit of health.

Agent can perform some actions if its health is lower than this value. Rank is [0..100].

**Parameters:**

*\_iHealth*,: desired threshold for health

**Returns:**

-

**6.22.2.7 final void Classes.CThreshold.SetAmmo (int *\_iAmmo*) [protected]**

Stablish the limit of ammunition.

Agent can perform some actions if its ammo is lower than this value. Rank is [0..100].

**Parameters:**

*\_iAmmo*,: desired threshold for ammo

**Returns:**

-

**6.22.2.8 final void Classes.CThreshold.SetAim (int *\_iAim*) [protected]**

Stablish the number of times that the agent must aim the enemy before to shoot.

Rank is [1..20].

**Parameters:**

*\_iAim*,: desired threshold for aim

**Returns:**

-

**6.22.2.9 final void Classes.CThreshold.SetShot (int *\_iShot*) [protected]**

Stablish the number of times that the agent must shoot consecutively before doing other action.

Rank is [1..20].

**Parameters:**

*\_iShot*,: desired threshold for shot

**Returns:**

-

**6.22.2.10** final void Classes.CThreshold.SetLook (int *\_iLook*) [protected]

Stablish the number of times (cycles) that the agent must wait (moving blindly) before looking again.

Rank is [0..100].

**Parameters:**

*\_iLook*,: desired threshold for look

**Returns:**

-

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

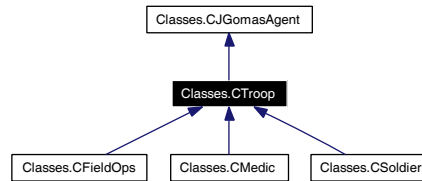
- CTroop.java

## 6.23 Classes.CTroop Class Reference

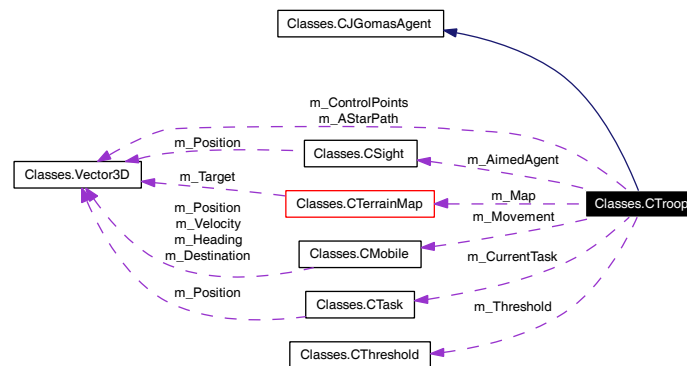
Inherits **Classes.CJGomasAgent**.

Inherited by **Classes.CFieldOps**, **Classes.CMedic**, and **Classes.CSoldier**.

Inheritance diagram for Classes.CTroop:



Collaboration diagram for Classes.CTroop:



### Static Public Attributes

- static final int **ARG\_TEAM** = 0
- static final int **TEAM\_NONE** = 0
- static final int **TEAM\_ALLIED** = 100
- static final int **TEAM\_AXIS** = 200
- static final int **CLASS\_NONE** = 0
- static final int **CLASS\_SOLDIER** = 1
- static final int **CLASS\_MEDIC** = 2
- static final int **CLASS\_ENGINEER** = 3
- static final int **CLASS\_FIELDOPS** = 4

### Protected Member Functions

- void **setup** ()  
*'setup' method of jade agents.*
- void **takeDown** ()  
*'takeDown' method of jade agents.*

- final int **Move** ()
- final int **GetHealth** ()  
*Get the current health of the agent.*
- final int **GetAmmo** ()  
*Get the current ammunition of the agent.*
- final int **GetStamina** ()  
*Get the current stamina of the agent.*
- final void **UseStamina** ()  
*Use stamina from the stamina bar if possible (there is at least 5 units).*
- final int **GetPower** ()  
*Get the current power of the agent.*
- final void **UsePower** ()  
*Use power from the power bar if possible (there is at least 25 units).*
- final void **AddServiceType** (String \_sServiceType)  
*Adds a type of service to the service type list.*
- final boolean **CheckStaticPosition** ()  
*Checks our position on the static map.*
- final boolean **CheckStaticPosition** (double \_x, double \_z)  
*Checks a position on the static map.*
- final void **AddTask** (int \_tTypeOfTask, AID \_Owner, String \_sContent)  
*Adds a task to the task list.*
- final void **AddTask** (int \_tTypeOfTask, AID \_Owner, String \_sContent, int \_iPriority)  
*Adds a task to the task list with a modified priority.*
- final void **Look** ()  
*The agent looks in the direction he is walking.*
- final boolean **Shot** (int \_iShotNum)  
*The agent shoots in the direction he is aiming.*
- final void **PerformAimAction** ()  
*Action to do when agent has an enemy at sight.*
- final boolean **HaveAgentToShot** ()  
*To know if an enemy is aimed.*
- final int **GetTeamFromAgentInSight** (int \_eType)  
*To know the team of an enemy in sight.*

- final int **GetClassFromAgentInSight** (int \_eType)  
*To know the class of an enemy in sight.*
- void **CallForMedic** ()  
*Request for medicine.*
- void **CallForAmmo** ()  
*Request for ammunition.*
- void **CallForBackup** ()  
*Request for backup.*
- void **UpdateTargets** ()  
*Update priority of all 'prepared (to execute)' tasks.*
- boolean **ShouldUpdateTargets** ()  
*Should we update now all 'prepared (to execute)' tasks?*
- void **ObjectivePackTaken** ()  
*The agent has got the objective pack.*
- void **SetUpPriorities** ()  
*Definition of priorities for each kind of task.*
- void **PerformNoAmmoAction** ()  
*Action to do if this agent cannot shoot.*
- void **PerformTargetReached** (CTask \_CurrentTask)  
*Action to do when this agent reaches the target of current task.*
- void **GenerateEscapePosition** ()  
*Calculates a new destiny position to escape.*
- boolean **GeneratePath** ()  
*Calculates a new destiny position to walk.*
- void **CreateControlPoints** ()  
*Calculates an array of positions for patrolling.*
- void **PerformEscapeAction** ()  
*Action to do when the agent tries to escape.*
- boolean **GetAgentToAim** ()  
*Calculates if there is an enemy at sight.*
- void **PerformLookAction** ()  
*Action to do when the agent is looking at.*

## Protected Attributes

- **AID m\_Manager** = null  
*Variable used to store the AID of Manager.*
- **Hashtable m\_TaskList**  
*List of prepared to execut tasks.*
- **CTask m\_CurrentTask**  
*Variable used to point the current task in execution.*
- **boolean m\_bObjectiveCarried**  
*Variable indicating if this agent is carrying the objective pack (flag).*
- **int m\_TaskPriority [ ]**  
*Array of default values of priorities for each task.*
- **Vector3D m\_ControlPoints [ ]**  
*Array of points used in patrolling task.*
- **int m\_iControlPointsIndex**  
*Current position in array m\_ControlPoints.*
- **Vector3D m\_AStarPath [ ]**  
*Array of points used in walking (a calculated) path task.*
- **int m\_iAStarPathIndex**  
*Current position in array m\_AStarPath.*
- **ArrayList m\_FOVObjects**  
*List of objects in the agent's Field Of Vision.*
- **CSight m\_AimedAgent**  
*Current aimed enemy.*
- **int m\_eTeam**
- **int m\_eClass**
- **boolean m\_bFighting**  
*Variable indicating if agent is fighting at this moment.*
- **boolean m\_bEscaping**  
*Variable indicating if agent is escaping at this moment.*
- **CMobile m\_Movement**  
*Current position, direction, and so on.*
- **int m\_iSoldiersCount**
- **int m\_iMedicsCount**
- **int m\_iEngineersCount**
- **int m\_iFieldOpsCount**

- **int m\_iTeamCount**
- **CThreshold m\_Threshold**

*Limits of some variables (to trigger some events).*

- **CTerrainMap m\_Map**

*Current Map.*

- **String m\_sMedicService**
- **String m\_sAmmoService**
- **String m\_sBackupService**

## Static Protected Attributes

- **static final int TRANSITION\_DEFAULT = 0**
- **static final int TRANSITION\_TO\_STANDING = 1**
- **static final int TRANSITION\_TO\_GOTO\_TARGET = 2**
- **static final int TRANSITION\_TO\_TARGET\_REACHED = 3**
- **static final int TRANSITION\_TO\_FIGHTING = 4**
- **static final int MV\_OK = 0**
- **static final int MV\_CANNOT\_GET\_POSITION = 1**
- **static final int MV\_NOT\_MOVED\_BY\_TIME = 2**

## Classes

- **class FSM\_Fighting**
- **class FSM\_GoToTarget**
- **class FSM\_Quit**
- **class FSM\_Standing**
- **class FSM\_TargetReached**

### 6.23.1 Member Function Documentation

#### 6.23.1.1 void **Classes.CTroop.setup()** [protected]

'setup' method of jade agents.

This method perform actions in common to **CTroop**(p. 56) agents (and derived classes) and calls parent's setup.

#### Parameters:

-

#### Returns:

-

Reimplemented from **Classes.CJGomasAgent** (p. 21).

Reimplemented in **Classes.CFieldOps** (p. 18), **Classes.CMedic** (p. 25), and **Classes.CSoldier** (p. 45).



**6.23.1.2 void Classes.CTroop.takeDown () [protected]**

'takeDown' method of jade agents.

This method calls parent's setup.

**Parameters:**

-

**Returns:**

-

Reimplemented from **Classes.CJGomasAgent** (p. 22).

**6.23.1.3 final int Classes.CTroop.Move () [protected]****6.23.1.4 final int Classes.CTroop.GetHealth () [protected]**

Get the current health of the agent.

**Parameters:**

-

**Returns:**

m\_iHealth: current value for health

**6.23.1.5 final int Classes.CTroop.GetAmmo () [protected]**

Get the current ammunition of the agent.

**Parameters:**

-

**Returns:**

m\_iAmmo: current value for ammo

**6.23.1.6 final int Classes.CTroop.GetStamina () [protected]**

Get the current stamina of the agent.

**Parameters:**

-

**Returns:**

m\_iStamina: current value for stamina bar

**6.23.1.7 final void Classes.CTroop.UseStamina () [protected]**

Use stamina from the stamina bar if possible (there is at least 5 units).

**Parameters:**

-

**Returns:**

-

**6.23.1.8 final int Classes.CTroop.GetPower () [protected]**

Get the current power of the agent.

**Parameters:**

-

**Returns:**

m\_iPower: current value for power bar

**6.23.1.9 final void Classes.CTroop.UsePower () [protected]**

Use power from the power bar if possible (there is at least 25 units).

Power bar is reduced in 25 units.

**Parameters:**

-

**Returns:**

-

**6.23.1.10 final void Classes.CTroop.AddServiceType (String *\_sServiceType*) [protected]**

Adds a type of service to the service type list.

This method registers all types of services to offer in a list, excluding repeated services.

**Parameters:**

*\_sServiceType*

**Returns:**

-

**6.23.1.11 final boolean Classes.CTroop.CheckStaticPosition () [protected]**

Checks our position on the static map.

This method checks if our position on the static map is valid to walk on, and returns the result. It must be called before to update variables.

**Parameters:**

-

**Returns:**

TRUE (agent can walk on) | FALSE (agent cannot walk on)

**6.23.1.12 final boolean Classes.CTroop.CheckStaticPosition (double *\_x*, double *\_z*)**  
[protected]

Checks a position on the static map.

This method checks if a position on the static map is valid to walk on, and returns the result.

**Parameters:***\_x**\_z***Returns:**

TRUE (agent can walk on) | FALSE (agent cannot walk on)

**6.23.1.13 final void Classes.CTroop.AddTask (int *\_iTypeOfTask*, AID *\_Owner*, String *\_sContent*)**  
[protected]

Adds a task to the task list.

This method adds a task to the task list with the default priority in agreement to the type of task. If there is a task of same type and same owner, it doesn't create a new task: simply substitutes some attributes with newer values.

**Parameters:***\_iTypeOfTask* one of the defined types of tasks.*\_Owner* the agent that induces the creation of the task.*\_sContent* is a position: ( *x* , *y* , *z* ) .**Returns:**

-

**6.23.1.14 final void Classes.CTroop.AddTask (int *\_iTypeOfTask*, AID *\_Owner*, String *\_sContent*, int *\_iPriority*)** [protected]

Adds a task to the task list with a modified priority.

This method adds a task to the task list with the priority passed as parameter, non the standard priority. If there is a task of same type and same owner, it doesn't create a new task: simply substitutes some attributes with newer values.

**Parameters:***\_iTypeOfTask* one of the defined types of tasks.*\_Owner* the agent that induces the creation of the task.*\_sContent* is a position: ( *x* , *y* , *z* ) .

*\_iPriority* priority of task

**Returns:**

-

**6.23.1.15 final void Classes.CTroop.Look () [protected]**

The agent looks in the direction he is walking.

This method sends a **FIPA INFORM** message to Manager. Once message is sent, agent will be blocked waiting a response message from Manager. The content of received message is stored in the variable `m_FOVObjects`.

**Parameters:**

-

**Returns:**

-

**6.23.1.16 final boolean Classes.CTroop.Shot (int *\_iShotNum*) [protected]**

The agent shoots in the direction he is aiming.

This method sends a **FIPA INFORM** message to Manager. Once message is sent, the variable `m_iAmmo` is decremented.

**Parameters:**

*\_iShotNum*

**Returns:**

TRUE (shot done) | FALSE (cannot shoot, has no ammo)

**6.23.1.17 final void Classes.CTroop.PerformAimAction () [protected]**

Action to do when agent has an enemy at sight.

This method is called when agent has looked and has found an enemy, calculating (in agreement to the enemy position) the new direction where is aiming.

**Parameters:**

-

**Returns:**

-

**6.23.1.18 final boolean Classes.CTroop.HaveAgentToShot () [protected]**

To know if an enemy is aimed.

This method is called just before agent can shoot. If an enemy is aimed, a value of TRUE is returned. Otherwise, the return value is FALSE. The result is used to decide if agent must shoot.

**Parameters:**

-

**Returns:**

TRUE (aimed enemy) | FALSE (no aimed enemy)

**6.23.1.19 final int Classes.CTroop.GetTeamFromAgentInSight (int \_eType) [protected]**

To know the team of an enemy in sight.

This method usually is called in the `GetAgentToAim` method to get the team of an enemy in sight from the variable `_eType` (which is a strange combination of `m_eTeam` and `m_eClass` variables). The result is returned.

**Parameters:**

`_eType,:` (team + class) of an agent.

**Returns:**

TEAM\_ALLIED | TEAM\_AXIS

**6.23.1.20 final int Classes.CTroop.GetClassFromAgentInSight (int \_eType) [protected]**

To know the class of an enemy in sight.

This method usually is called in the `GetAgentToAim` method to get the class of an enemy in sight from the variable `_eType` (which is a strange combination of `m_eTeam` and `m_eClass` variables). The result is returned.

**Parameters:**

`_eType,:` (team + class) of an agent.

**Returns:**

CLASS\_NONE | CLASS\_SOLDIER | CLASS\_MEDIC | CLASS\_ENGINEER | CLASS\_FIELDOPS

**6.23.1.21 void Classes.CTroop.CallForMedic () [protected]**

Request for medicine.

This method sends a **FIPA REQUEST** message to all agents who offers the `m_sMedicService` service.

The content of message is: ( `x` , `y` , `z` ) ( `health` ) .

Variable `m_iMedicsCount` is updated.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.22 void Classes.CTroop.CallForAmmo () [protected]**

Request for ammunition.

This method sends a **FIPA REQUEST** message to all agents who offers the `m_sAmmoService` service.

The content of message is: ( `x` , `y` , `z` ) ( `ammo` ) .

Variable `m_iFieldOpsCount` is updated.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.23 void Classes.CTroop.CallForBackup () [protected]**

Request for backup.

This method sends a **FIPA REQUEST** message to all agents who offers the `m_sBackupService` service.

The content of message is: ( `x` , `y` , `z` ) ( `SoldiersCount` ) .

Variable `m_iSoldiersCount` is updated.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.24 void Classes.CTroop.UpdateTargets () [protected]**

Update priority of all 'prepared (to execute)' tasks.

This method is invoked in the state *STANDING*, and it's used to re-calculate the priority of all tasks (targets) in the task list of the agent. The reason is because JGOMAS Kernel always execute the maximum priority task.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.25 boolean Classes.CTroop.ShouldUpdateTargets ()** [protected]

Should we update now all 'prepared (to execute)' tasks?

This method is a decision function invoked in the state *GOTO\_TARGET*. A value of `TRUE` break out the inner loop, making possible to JGOMAS Kernel extract a more priority task, or update some attributes of the current task. By default, the return value is `FALSE`, so we execute the current task until it finalizes.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

`FALSE`

**6.23.1.26 void Classes.CTroop.ObjectivePackTaken ()** [protected]

The agent has got the objective pack.

This method is called when this agent walks on the objective pack, getting it.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.27 void Classes.CTroop.SetupPriorities ()** [protected]

Definition of priorities for each kind of task.

This method can be implemented in CTroop's derived classes to define the task's priorities in agreement to the role of the new class. Priorities must be defined in the array `m_TaskPriority`.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

Reimplemented in **Classes.CFieldOps** (p. 19), **Classes.CMedic** (p. 26), and **Classes.CSoldier** (p. 46).

**6.23.1.28 void Classes.CTroop.PerformNoAmmoAction ()** [protected]

Action to do if this agent cannot shoot.

This method is called when the agent try to shoot, but has no ammo. The agent will spit enemies out. :-)

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.29 void Classes.CTroop.PerformTargetReached (CTask \_CurrentTask) [protected]**

Action to do when this agent reaches the target of current task.

This method is called when the agent goes to state *TARGET\_REACHED*. In agreement to current task, agent must realize some actions (for example, to get next point to walk from patrolling path). The actions in common to all roles are implemented at this level of hierarchy: *TASK\_PATROLLING*, *TASK\_WALKING\_PATH*, *TASK\_RUN\_AWAY*.

*It's very useful to overload this method.*

**Parameters:***\_CurrentTask***Returns:**

-

Reimplemented in **Classes.CFieldOps** (p. 19), and **Classes.CMedic** (p. 26).

**6.23.1.30 void Classes.CTroop.GenerateEscapePosition () [protected]**

Calculates a new destiny position to escape.

This method is called before the agent creates a task for escaping. It generates a valid random point in a radius of 50 units. Once position is calculated, agent updates its destiny to the new position, and automatically calculates the new direction.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.31 boolean Classes.CTroop.GeneratePath () [protected]**

Calculates a new destiny position to walk.

This method is called before the agent creates a *TASK\_GOTO\_POSITION* task. It will try (for 5 attempts) to generate a valid random point in a radius of 20 units. If it doesn't generate a valid position in this cycle, it will try it in next cycle. Once a position is calculated, agent updates its destination to the new position, and automatically calculates the new direction.

*It's very useful to overload this method.*

**Parameters:**

-



**Returns:**

TRUE: valid position generated / FALSE cannot generate a valid position

**6.23.1.32 void Classes.CTroop.CreateControlPoints () [protected]**

Calculates an array of positions for patrolling.

When this method is called, it creates an array of *n* random positions. For medics and fieldops, the rank of *n* is [1..1]. For soldiers, the rank of *n* is [5..10].

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.33 void Classes.CTroop.PerformEscapeAction () [protected]**

Action to do when the agent tries to escape.

This method is called just before this agent creates a TASK\_RUN\_AWAY task. By default, the only thing it does is to reset its aimed enemy: `m_AimedAgent = null`. If it's overloaded, it's convenient to call parent's method.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.1.34 boolean Classes.CTroop.GetAgentToAim () [protected]**

Calculates if there is an enemy at sight.

This method scans the list `m_FOVObjects` (objects in the Field Of View of the agent) looking for an enemy. If an enemy agent is found, a value of TRUE is returned and variable `m_AimedAgent` is updated. Note that there is no criterion (proximity, etc.) for the enemy founded. Otherwise, the return value is FALSE.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

TRUE: enemy found / FALSE enemy not found

**6.23.1.35 void Classes.CTroop.PerformLookAction () [protected]**

Action to do when the agent is looking at.

This method is called just after Look method has ended. At this point, the only thing it does is to update variable `m_bFighting` calling the **GetAgentToAim()**(p. 69); method.

*It's very useful to overload this method.*

**Parameters:**

-

**Returns:**

-

**6.23.2 Member Data Documentation****6.23.2.1 AID Classes.CTroop.m\_Manager = null [protected]**

Variable used to store the AID of Manager.

**6.23.2.2 Hashtable Classes.CTroop.m\_TaskList [protected]**

List of prepared to execut tasks.

**6.23.2.3 CTask Classes.CTroop.m\_CurrentTask [protected]**

Variable used to point the current task in execution.

**6.23.2.4 boolean Classes.CTroop.m\_bObjectiveCarried [protected]**

Variable indicating if this agent is carrying the objective pack (flag).

**6.23.2.5 int Classes.CTroop.m\_TaskPriority[] [protected]**

Array of default values of priorities for each task.

**6.23.2.6 Vector3D Classes.CTroop.m\_ControlPoints[] [protected]**

Array of points used in patrolling task.

**6.23.2.7 int Classes.CTroop.m\_iControlPointsIndex [protected]**

Current position in array `m_ControlPoints`.

**6.23.2.8 Vector3D Classes.CTroop.m\_AStarPath[] [protected]**

Array of points used in walking (a calculated) path task.

**6.23.2.9 int Classes.CTroop.m\_iAStarPathIndex** [protected]

Current position in array m\_AStarPath.

**6.23.2.10 ArrayList Classes.CTroop.m\_FOVObjects** [protected]

List of objects in the agent's Field Of Vision.

**6.23.2.11 CSight Classes.CTroop.m\_AimedAgent** [protected]

Current aimed enemy.

- 6.23.2.12 **final int** `Classes.CTroop.ARG_TEAM = 0` [static]
- 6.23.2.13 **final int** `Classes.CTroop.TEAM_NONE = 0` [static]
- 6.23.2.14 **final int** `Classes.CTroop.TEAM_ALLIED = 100` [static]
- 6.23.2.15 **final int** `Classes.CTroop.TEAM_AXIS = 200` [static]
- 6.23.2.16 **final int** `Classes.CTroop.CLASS_NONE = 0` [static]
- 6.23.2.17 **final int** `Classes.CTroop.CLASS_SOLDIER = 1` [static]
- 6.23.2.18 **final int** `Classes.CTroop.CLASS_MEDIC = 2` [static]
- 6.23.2.19 **final int** `Classes.CTroop.CLASS_ENGINEER = 3` [static]
- 6.23.2.20 **final int** `Classes.CTroop.CLASS_FIELDOPS = 4` [static]
- 6.23.2.21 **final int** `Classes.CTroop.TRANSITION_DEFAULT = 0` [static, protected]
- 6.23.2.22 **final int** `Classes.CTroop.TRANSITION_TO_STANDING = 1` [static, protected]
- 6.23.2.23 **final int** `Classes.CTroop.TRANSITION_TO_GOTO_TARGET = 2` [static, protected]
- 6.23.2.24 **final int** `Classes.CTroop.TRANSITION_TO_TARGET_REACHED = 3` [static, protected]
- 6.23.2.25 **final int** `Classes.CTroop.TRANSITION_TO_FIGHTING = 4` [static, protected]
- 6.23.2.26 **final int** `Classes.CTroop.MV_OK = 0` [static, protected]
- 6.23.2.27 **final int** `Classes.CTroop.MV_CANNOT_GET_POSITION = 1` [static, protected]
- 6.23.2.28 **final int** `Classes.CTroop.MV_NOT_MOVED_BY_TIME = 2` [static, protected]
- 6.23.2.29 **int** `Classes.CTroop.m_eTeam` [protected]
- 6.23.2.30 **int** `Classes.CTroop.m_eClass` [protected]
- 6.23.2.31 **boolean** `Classes.CTroop.m_bFighting` [protected]

Variable indicating if agent is fighting at this moment.

- 6.23.2.32 **boolean** `Classes.CTroop.m_bEscaping` [protected]

Variable indicating if agent is escaping at this moment.

**6.23.2.33 CMobile Classes.CTroop.m\_Movement** [protected]

Current position, direction, and so on.

..

**6.23.2.34 int Classes.CTroop.m\_iSoldiersCount** [protected]**6.23.2.35 int Classes.CTroop.m\_iMedicsCount** [protected]**6.23.2.36 int Classes.CTroop.m\_iEngineersCount** [protected]**6.23.2.37 int Classes.CTroop.m\_iFieldOpsCount** [protected]**6.23.2.38 int Classes.CTroop.m\_iTeamCount** [protected]**6.23.2.39 CThreshold Classes.CTroop.m\_Threshold** [protected]

Limits of some variables (to trigger some events).

**6.23.2.40 CTerrainMap Classes.CTroop.m\_Map** [protected]

Current Map.

**6.23.2.41 String Classes.CTroop.m\_sMedicService** [protected]**6.23.2.42 String Classes.CTroop.m\_sAmmoService** [protected]**6.23.2.43 String Classes.CTroop.m\_sBackupService** [protected]

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

- CTroop.java

## 6.24 Classes.Vector3D Class Reference

### Public Member Functions

- double **length** ()
- void **add** (**Vector3D** v)
- void **sub** (**Vector3D** v)
- double **dot** (**Vector3D** v)
- void **normalize** ()
- void **print** (String name)
- **Vector3D** (**Vector3D** v)
- **Vector3D** (double \_x, double \_y, double \_z)

### Public Attributes

- double **x**
- double **y**
- double **z**

### Package Functions

- **Vector3D** **cross** (**Vector3D** v)
- **Vector3D** ()

## 6.24.1 Constructor & Destructor Documentation

6.24.1.1 `Classes.Vector3D.Vector3D ()` [package]

6.24.1.2 `Classes.Vector3D.Vector3D (Vector3D v)`

6.24.1.3 `Classes.Vector3D.Vector3D (double _x, double _y, double _z)`

## 6.24.2 Member Function Documentation

6.24.2.1 `double Classes.Vector3D.length ()`

6.24.2.2 `void Classes.Vector3D.add (Vector3D v)`

6.24.2.3 `void Classes.Vector3D.sub (Vector3D v)`

6.24.2.4 `double Classes.Vector3D.dot (Vector3D v)`

6.24.2.5 `void Classes.Vector3D.normalize ()`

6.24.2.6 `Vector3D Classes.Vector3D.cross (Vector3D v)` [package]

6.24.2.7 `void Classes.Vector3D.print (String name)`

## 6.24.3 Member Data Documentation

6.24.3.1 `double Classes.Vector3D.x`

6.24.3.2 `double Classes.Vector3D.y`

6.24.3.3 `double Classes.Vector3D.z`

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

- `Vector3D.java`





## **Chapter 7**

# **JGomas File Documentation**

### **7.1 CAmmoPack.java File Reference**

#### **Namespaces**

- namespace **Classes**

#### **Classes**

- class **Classes.CAmmoPack**

## 7.2 CFieldOps.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CFieldOps**

## 7.3 CJGomasAgent.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CJGomasAgent**

## 7.4 CManager.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CMicroAgent**
- class **Classes.CDinObject**
- class **Classes.CManager**

## 7.5 CMedic.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CMedic**

## 7.6 CMedicPack.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CMedicPack**

## 7.7 CMobile.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CMobile**

## 7.8 CObjPack.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CObjPack**



## 7.9 CPack.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CPack**

## 7.10 CRegistry.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CRegistry**

## 7.11 CServer.java File Reference

### Namespaces

- namespace **Classes**
- namespace **java.io**
- namespace **java.net**

### Classes

- class **Classes.CServer**
- class **Classes.CSocketThread**

## 7.12 CService.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CService**

## 7.13 CSight.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CSight**

## 7.14 CSoldier.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CSoldier**

## 7.15 CTask.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CTask**

## 7.16 CTerrainMap.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CBase**
- class **Classes.CTerrain**
- class **Classes.CTerrainMap**



## 7.17 CTroop.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.CThreshold**
- class **Classes.CTroop**
- class **Classes.CTroop.FSM\_Standing**
- class **Classes.CTroop.FSM\_Quit**
- class **Classes.CTroop.FSM\_GoToTarget**
- class **Classes.CTroop.FSM\_TargetReached**
- class **Classes.CTroop.FSM\_Fighting**

## 7.18 Vector3D.java File Reference

### Namespaces

- namespace **Classes**

### Classes

- class **Classes.Vector3D**