

Configurable Multiagent System for Plug&Produce

AUTHOR
Fredrik Danielsson

Version

2.13

CMAS Main Page

Introduction

The CMAS from University West is a software package for Multiagent control (MAS) to facilitate the implementation of Plug&Produce concepts. Plug&Produce refer to autonomous manufacturing systems that can handle changes in both hardware and requirements online. CMAS is an acronym for Configurable Multiagent System, and its fundamental characteristic is its capacity to configure the system's overall behaviour rather than traditional programming approaches.

The work is based on efforts, many years of research and knowledge at University West in the Flexible Industrial Automation research group.

- This manual is divided in the following sections:**Introduction to Plug&Produce and Multiagent**
- **How to use CMAS**
- **Process plans**
- **Interfaces**
- **Agent actions**
- **Planner**
- **Automating deployment**
- **Variable adapters**
- **Structured text**

Plug & Produce and Multiagent

CMAS

CMAS is a framework for Plug & Produce that makes it possible to use configurations rather than programming to model and handle parts in a manufacturing system. The key tasks are to define goals for parts. Goals are central in the design of autonomous multi-agent systems. The final goal for a part is to become manufactured and this can be divided into several subgoals. A goal can achieve its completion by requesting skills from available resources. Therefore, the task of defining skills for resources is important. Typically, a resource skill is established upon resource acquisition and tends to remain relatively constant throughout the resource's lifespan. It's worth noting that a skill might demand additional control logic to function effectively, hence, it is possible to develop more conventional programs for skill execution as well. Typically, before the first deployment the system integrators will carrying out the definition of low-level skills.

Goal-oriented multi-agent control is one approach to handle flexible and reconfigurable manufacturing systems. Goals are important to be able to create autonomous agents. An autonomous agent must have the possibility to decide not only how to achieve a goal, but also which goal to select. For parts that will be manufactured, a state-oriented model is adopted implying that each agent associated with a part tries to reach a final state by executing all assigned goals. When all goals are finalized, the part is ready. A part is defined as a part of a product or the product itself that will be manufactured within the boundary of the actual

manufacturing system. Hence, a goal can only be fulfilled by utilizing skills on resources within this boundary. To assist part-agents to reach their goals, process plans exist that defines skills needed to reach a specific goal. Process plans can be defined without specifying resources, only required skills, requirements, and abstract interfaces are specified. Abstract interfaces are connected to a resource-agent when needed during runtime, the resource-agent are selected after a negotiation process. Hence, the process plans of goals separate the resources from the goals. This makes it possible to have several potential solutions in the system for the autonomous agents to select the most suitable process plans. A goal depicts the future desired state of the system. However, the sequence of goals is important, methods to configure and execute concurrent and parallel sequence of goals. Plug & Produce systems, controlled by CMAS, enables the possibility to rapidly plug in and out resources to adapt to the actual manufacturing situation. Resources prepared with suitable skills and process plans can be plugged in without doing any software reconfiguration of the CMAS. Parallel goals can utilize resources concurrently in parallel. If only one resource is available, goals that are configured to execute in parallel will be executed in sequence instead by the single resource. This approach enables the possibility to rapidly avoid bottlenecks in the production that can appear due to for example changed customer orders just by plugging in suitable resources. No reprogramming or software reconfiguration is needed in these situations.

How to use CMAS

This section aims to familiarise users with CMAS and demonstrate its practical application through some examples. CMAS serves as a powerful tool for modelling flexible Plug&Produce capable automation solutions. However, the flexibility is dependent on the modelling technique used. The following segment will delve into the fundamentals of CMAS, accompanied by illustrative examples to enhance comprehension and practical understanding.

- Configuration
- Example 1 - Name tag
- Example 2 - Crane

Configuration

There are two major tasks for CMAS to handle:

1. Configuration
2. Online control

Configuration is the task to describe for CMAS what you want to achieve and is a planning activity. The configuration can mainly be divided into some basic activities:

Activity	Action	Where
Add resources	When new resources have been acquired	Modelling tree
Add parts	Introduction of a new type of product	Modelling tree
Specify goals	Whenever you want to add or modify a part	On parts
Specify process plans	Whenever you want to specify or modify how a specific goal is achieved	Process plans
Specify skills	When a skill is added or modified	On resources

A good configuration approach is crucial when it comes to a configurable multi-agent system. The configuration is based on an **ontology model**. To understand how to make a configuration we will start with a small example.

Goals

Let's say we want to transform a metal plate into a name tag, see figure below. The first step is to create a virtual representation of the raw metal plate within the system. This virtual representation is managed by an entity called a **part agent**. The part agent represents the raw metal plate and oversees its transformation into a name tag. In this example, the transformation involves smoothing the corners, drilling four holes, and printing a name on the plate.

These transformations are expressed as **goals**. Every part agent is assigned goals, which allow it to dynamically plan and schedule a sequence of actions needed to achieve its overall objective, in this case, becoming a finalized name tag. A goal defines a desirable state for the part agent. In this example, the overall aim for the part agent, "becoming a name tag", can be broken down into the following goals:

#	Goal	Precondition
1.	Get rounded corners	None
2.	Get hole in the NE corner	Goal 1 is finished
3.	Get hole in the NW corner	Goal 1 is finished
4.	Get hole in the SE corner	Goal 1 is finished
5.	Get hole in the SW corner	Goal 1 is finished
6.	Get a printed name	Goal 2 & 3 & 4 & 5 is finished

The order of the goals is important, in this scenario, goal 1 can be reached whenever as it does not depend on any other goals. Goals 2, 3, 4 and 5 can only be reached after goal 1 is completed. These goals can be completed in any order or even in parallel. Finally, goal 6 must be completed at the end, as it depends on the completion of all preceding goals. This order of goals forms a **process plan**, which can be expressed and visualised using a **goal chart**.

Note

Goals does not state how the goals should be reach only the order of them. It is also important to break down the problem in a feasible amount of goals to take advantage of the flexibility within CMAS.

Process plans

A part agent must be able to reach its goals. A **process plan** simply describes a sequence of skills to be used to reach a specific goal. Hence, a process plan assists in transforming a part agent from its current state to a new state where the goal is fulfilled. See the figure below for an example of a process plan to get rounded corners.

Step #	Skill
1	Grind corner NE
2	Grind corner NW
3	Grind corner SE
4	Grind corner SW

Note

A process plan only describes the main steps and skills involved to reach a goal. It does not describe what resources to use. This makes the solution more independent of the real hardware and thus more flexible.

Skills

To be able to manufacture a name tag we need resources. A **resource agent** is a virtual representation of a resource in the manufacturing system. A typical resource might be a robot, a drill bit or a fixture. Every resource agent can be assigned one or several **skills** (abilities). A skill represents a way to do something (change the state of the world) and is published via an **interface** on the specific resource agent.

Examples of skills might be transportation, painting, or drilling. A resource might have several skills, e.g., a robot could represent both painting and transportation. A skill might require other skills on other resources to be able to execute. An example might be a make hole skill on a drill bit resource that requires a drill skill on another resource. This chain of skills is a key element to get the system more flexible. A skill is described as a sequence of local actions and remote skills executed in a specific order, see figure below for an example of a skill called make hole.

Step #	Action/Skill	Resource involved
1	Move to the target position	Executed on robot
2	Start Drill motor	Executed on drill motor
3	Stop drill motor	Executed on drill motor
4	Move home	Executed on robot

Example 1 - name tag

To perform a real configuration for the scenario described above, let's create a simplified example focusing on the goals for drilling holes in the name tag. The manufacturing process involves a drill station and its associated drill bits. We will begin by designing a basic model of the agents needed. The resources are divided into two distinct resource agents: *DrillBit* and *DrillStation*. Additionally, a Part Agent is required to represent the name tag being manufactured.

Now start the CMAS tool, it should look something like this.

Resource Agent 1 - Drill Bit

The first step is to add a **resource agent**. In CMAS, you can use the **Modeling Tree** viewer located on the left side. This is the primary interface for modeling and displays all entities within the model.

At the top of the CMAS window, you will find the main toolbar, organized into different categories across various tabs. Begin by switching to the **Modeling** tab. Each tab is further divided into groups. Within the Modeling tab, locate the **Agents** group. Click the **[resource]** button to create a resource agent, which will then appear in the **Modeling Tree** viewer.

Note

You can also click the **[Add new]** button in the modelling tree view. A dialogue window will then ask you what type of entity to add.

When the new entity is created it is automatically added to the **Modelling Tree** and selected. You can see all properties of the selected entity to the right in the **Properties**

viewer. To be able to find our new resource in the system we need to give it a friendly **Name**, let's name it *DrillBit*. You can change the name by navigating to the **Properties** viewer and replace the automatically generated name with *DrillBit*. Further, it is also possible to give a more detailed description of the resource in the **Description**. The **Entity ID** is a unique *string* used by the platform to keep track of our agents during runtime. It is possible to give **Entity IDs** manually but usually it is best to let the system decide an **Entity ID** automatically.

Note

An **Entity ID** should be unique in the entire configuration.

The next step is to assign interfaces to our drill bit agent. An **interface** is a defined way for agents to communicate or interact with each other. An **interface** also represents physical and logical compatibility. Our drill bit needs to interact with two other agents, the *name tag* and the *drill station*. Hence, we need two interfaces, the first to be able to drill a hole in a part and the second to be able to utilise a drill station for rotation of the bit (our self).

#	Interface name	Description
1	Hole	Enables the drill bit to create holes in the name tag.
2	Chuck_B10	Allows the drill bit to interact with the drill station for rotation of the bit .

First, create the **Interface** to interact with parts. To do this expand the *DrillBit* resource in the modelling view and select the relation **Interfaces**. When the **Interfaces** is selected press the [interface] button in the interface group in the tool bar. While the new interface is selected in the modelling tree view go to the properties view and change the name to *Hole*.

To reflect the size of the hole we need a variable. On the *Hole* interface let's create an **Interface variable** named *Diameter*. To add variables to an **Interface** expand the interface in the modelling view and select the relation **Variables**. When the relation variables is selected press the [variablereal] button in tool bar to create a new variable. Now go to the properties view and change the variable name to *Diameter*, also set the **UpperBound** and **LowerBound** to *5.0* to represent a diameter of exactly 5 mm. Our drill bit **interface** is now compatible with holes that have a diameter of 5.

Note

Interface information is used to check for compatibility. I.e., the **Name** of the **interfaces** must match and then all the **Variables** types plus their range (**UpperBound** / **LowerBound**), for more details see **negotiation**.

Further, a hole also has a position lets create two more **Interface variable** called *x* and *y*. We can use the same type *VariableReal* to represent a real number. We do not give a range here, hence, any range will be compatible with these variables. With this modelling strategy it is now possible to create several drill bits with the same interfaces and name but with different diameters.

We also need to create a **Skill** on the *Hole* **interface**. A skill represents an action that other agents can use to change the state of the world, in this case to make a hole by the use of our drill bit **resource agent**. Now select the **Interface Hole** and select the relation **Skills**. Click the [processplanstructuredtext] button on the tool bar.

Go to the properties view and name the new skill *MakeAHole*. We can leave the other skill details for now and come back to them later.

The *DrillBit* by itself cannot make a hole since it cannot rotate nor fixate. Therefore, it must utilise another resource, a drill station of some kind. To be able to interact with a drill station we need another **interface**. Let us, therefore, create a new **interface** named *Chuck_B10* on the resource. Remember that the *DrillBit* agent knows the position of the hole through the first interface but cannot make use of it. Therefore, the hole position must be transferred to the drill. For that reason, add two **Interface variables** to the *Chuck_B10* interface namely, x and y. The *DrillBit* agent is now configured. It is now possible to create many drill bits in the same way representing different chucks or hole diameters. Our configuration can now be visualised in a figure.

Resource Agent 2 - Drill Station

Now create a drill station as a **Resource Agent** with a compatible interface to our drill bit *Chuck_B10* with two **Interface variables** named x and y. Also, create an empty skill named *Drill* on the same **interface**.

Note

This chain of agents with matching interfaces is an important key to flexibility. It is now possible to duplicate resources or replace resources as long as they are compatible with each other (i.e. compatible interfaces). We will see later how this work.

Part Agent - NameTag

Now when the resources are ready the next step is to create a **Part agent** that will be the virtual representation of the product, name tag. To add a part agent select the root in the modelling tree and press the [part] button in the tool bar. Change the name to *NameTag* for our new part.

On the **Part agent** we need to create four holes each hole will be expressed as a **Goal**. A **Goal** is something a **Part agent** will try to fulfil by utilising **Process plans** and/or **Skills**. To add a **Goal** expand the **Part agent** in the modelling view and select the relation **Goals** and click the [goal] button on the tool bar. Change the name of the new **Goal** to *Hole1*. To reach the goal we need to name a good **Process plans**. To do this select the button [...] behind the ProcessPlanName to get a list of available process plans and select *MakeAHole*. Now we have a **Goal** to get a hole but we also need to give information on where we want the hole.

To do this we need to create **Goal variables**. Expand the *Hole1* in the modelling tree and select **Variables** and the click the [variablereal] button on the tool bar. Now create two **VariableReal** variables and name them x and y. For x set the **Value** to 1 and y to 1. Further, also create a variable called *Diameter* and set the **Value** to 5.0. In this way, we create a demand for our **Part agent** to only consider a hole with a specific diameter of 5.0.

In the same way, as we added **Interfaces** to our **Resources** we have now created an interface on our goal. The main difference is the minor adjustment needed while designating the interface name in the context of a **Goal**. In the properties view for the goal click the button [...] behind **Interface** property. You should now get a list of available interface names. Select the *Hole* interface in the list. This is a convenient way to set the name of the interface.

Now, in the same way create three more goals with the same variables but with

different values. This can be done by selecting *Hole1* in the modelling view and press [Duplicate] button in the modelling view. You will now get a copy of the **Goal** *Hole1* and you can easily rename it to *Hole2* and set new values for x and y. To easier see the difference between all goals you can give more details in the **Description** field. The **Part agent** is now ready and should look something like this.

To make unique positions for each hole change the x and y variables for goal 2, 3 and 4 to:

Goal	x	y
Hole1	1	1
Hole2	10	1
Hole3	10	10
Hole4	1	10

Process Plans

A **Process plan** is a description on how to reach goals. **Process plans** can be created on many different levels but for now let's focus on the part *NameTag*. Select the *NameTag* in the modelling view and in the properties view you can open the **ProcessPlan** editor by pressing [Edit].

A new editor will open.

The start and the end of the **ProcessPlan** is indicated in the editor. It is now possible to drag the **Goals** from the goal view in to the editor. However, since there is no specific order of the 4 holes we can say to the system that any order is valid. To do this drag the **Parallel** statement from the statement view and drop it on the start statement. The parallel statement will then be inserted after the start statement.

The **Parallel** statement is now visible as some parallel lines. Now drag each *Goal* in the goal list and drop them on the top line of the **Parallel** statement.

The editor now shows all 4 goals in parallel. It is also possible to see and change the name of each goal and the interface name and the requested process plan name in the editor. Now close the editor.

Skills

We also need to define the **skill** for *MakeAHole* on the *DrillBit* agent. To do this select the skill *MakeAHole* in the modelling view and in the properties view press the button Actions. The process plan for the **skill** is now visible, this is a Structure Text process plan. We now have the flexibility to describe in Structure Text what we want to achieve. This is an alternative to the **Goal** oriented process plan used for the part.

```
Program Processplan;

function onRun() : integer;
begin
    // Add actions here

    // Return value - running, done, failed or interrupted
    onRun:=processplan.done;
end.
```

What we need to do is to add actions on how to drill with the bit. The drill bit need another resource to be able to make a hole, i.e. the drill bit by it self cannot make a

hole. That resource must have an **interface** that is compatible with our **interface**. However, we do not care about which resource it is at this planning stage we just want it to be a resource **interface** that we can use. This unknown but required **interface** we can refer to as an **abstract interface** and all **abstract interfaces** must be declared at the top of a process plan. To be able to refer to this **abstract interface** in our process plan we give it a name when we declare it. It is possible to declare as many **abstract interfaces** as we wish in a process plan but for this plan one is enough. In the main block in-between **Begin** and **End** we need to give the details on what we want to accomplish and how to make use of our abstract interface.

The position of the hole is something that we need to give when we ask for the **skill** *MakeHole*. We can do that by assigning the position from the **goal** (*Goal.x*) to the skills interface variable. However, since we have declared all variable on the interfaces all values are transferred automatically.

An abstract interface is a way to describe what kind of resource we need without a hard link to a specific resource. An abstract interface is converted to a real interface by a **negotiation process** during runtime.

When defined, it is possible to use the abstract interface and request a skill by using the abstract interface.

```
Program ProcessplanMakeAHole;

abstract aDrillResource : interface;

function onRun() : integer;
begin
    // Add actions here
    aDrillResource.Drill();

    // Return value - running, done, failed or interrupted
    onRun:=processplan.done;
end.
```

You can now close the editor for the skill *MakeAHole*.

Now we need to do the same for the Drill station to define a skill.

The action for the **skill** *Drill* on the *DrillStation* is to now move the spindle to the right position and start the spindle. To simplify we only write some messages on the console with *writeln*. Note, that it is indeed possible to control a real drill station within the skill or to send messages/set signals to an external controller.

```
Program ProcessplanDrill;

function onRun() : integer;
begin
    // Add actions here
    writeln('Will drill a hole at position ', Interface.x, ' ', Interface.y);

    // Return value - running, done, failed or interrupted
    onRun:=processplan.done;
end.
```

Deployment

Our configuration should now look something like this and this is now our flexible control system.

We are now ready to test our configuration in the runtime environment. By pressing the [Run] button on the tool bar (first switch tab to Run), instances of our configuration will be transferred to the runtime environment. A new runtime tree

viewer will open to visualise the runtime environment. The tree is divided in four different areas the first is called *Depot*. This is an area where agents that are ready to be deployed will remain until they are deployed manually, automatically by CMAS or by user logic.

Note

CMAS can be configured to automatically deploy all resource agents at runtime by setting the **DeployResourcesOnStart** option to **true** in the **Preferences** viewer. If this option is enabled, your *DrillBit* and *DrillStation* agents will appear in the *Running* folder after start.

We are now ready to deploy our agents and can do that by pressing the [Deploy all] button in the runtime view. If you want to deploy each agent individually right click on the agent a chose deploy or make use of the main tool bar. When an agent is deployed it will move to the *Running* area. When a **part** agent is deployed and move to the *Running* area a new part is generated in the **Depot** ready to be deployed.

Note

If you get a compiler error you must go back to the configuration and correct the error. To test again you must restart the runtime environment.

In bottom of CMAS you should have a console window that display user outputs from agents. With our *writeln* call inside the skill *Drill* we should now be able to see that the drill will produce 4 holes. When a part agent has reached all goals it will be moved to the *Finished* or *Failed* area depending on the result. You can expand the *Finished* area to see the part and check that all goals has become ready and green. It is now possible to produce as many name tags as desired by deploying new *NameTag* agents with the [Deploy all] button. To stop the runtime environment press the [Stop] button.

You can also view the history of a finished part. To do so, select the part you want to examine from the **Finished** folder. Right-click on the part to open the context menu and choose **Show History**. A new window will open, displaying the complete history of the part.

Of course it is possible to add print outs for other activities to see what's going on. Print outs is one way to see activities in the system. An example of the process plan *GetHole* with print outs:

```
Program ProcessplanMakeAHole;

abstract aDrillResource : interface;

function onRun() : integer;
begin
    // Add actions here
    writeln('Start of process plan ', this.Name, ' with a diameter of ',
interface.Diameter);
    aDrillResource.Drill();
    writeln('End of process plan ', this.Name);

    // Return value - running, done, failed or interrupted
    onRun:=processplan.done;
end.
```

Errors

Sometimes, identifying errors or mistakes in the model can be challenging, as they are typically discovered during runtime. If you encounter a compiler error (such as a mistake in a Structured Text skill), try following the prompts in the dialog boxes.

These will guide you to the correct editor and highlight the line that caused the error. Additionally, note that the editor will display a red underline to help you identify

errors as you code.

In some cases, you may see multiple error dialogs due to a single mistake. As in this example, the primary error is a missing semicolon (;) at the end of the statement on line 6 in the **ProcessPlanDrill** program. However, when the **ProcessPlanMakeAHole** attempts to use the skill, it will also fail due to the problem to execute the skill.

Another common issue is when nothing happens after deploying all your agents. To troubleshoot and see what is happening in the system, you can open the **Agent Thread** viewer.

This viewer shows the negotiation status for all agents. In this example, you can observe that the busy **ProcessPlanMakeAHole** on the **Hole** interface of the **DrillBit** resource is unable to find any matching interfaces during the negotiation process.

To check interface compatibility, you can use the **Schema** viewer. At the top of the **Schema** viewer, change the **Show** dropdown menu to **Show folder or selection content**. Then, select the agent you want to check. The schema viewer will display the agent, and all compatible agents will be connected by lines. If any connections are missing, it indicates a problem with the interfaces. For compatibility, the interfaces must match in both name and variables.

It is also possible to select a specific interface in the model viewer. Right-click on the interface to open the context menu and choose **Verify Interface**. This will display a list of all compatible interfaces.

Another way to check for errors is by using **Project Verification**. Change the console view to **Verification** and click the **Verify Project** button. CMAS will then attempt to detect any possible errors. Note that, since we are not running the system, it can be challenging to check complex, abstract interfaces during the modeling phase. E.g, interface variables are only updated during runtime if they are depend on external signals.

Crane Simulation Example

Introduction

Start the crane simulation program, it should look like the picture below. Within the simulation, there's a crane capable of motions along the x y plane. Equipped with a vacuum gripper, the crane can affix products to itself and move them around. Within the simulation model, there exist two sources: source 1 for part A and source 2 for part B. Additionally, two processes are available for part processing. Upon completion, a part can be released into the sink for removal from the system.

The manufacturing environment includes sensors and control signals that can be reach through MODBUS TCP.

Table of MODBUS signals

Name	Type	Direction	Address	Explanation
setX	Integer	Write	1	Crane set value for x
setY	Integer	Write	2	Crane set value

				for y
vacuum	Boolean	Write	3	Crane lift mechanism Vacuum on/off
startp1	Boolean	Write	4	Start signal to process 1
startp2	Boolean	Write	5	Start signal to process 2
atX	Integer	Read	15	Crane current position in x
atY	Integer	Read	16	Crane current position in y
source1	Boolean	Read	17	Proximity sensor in source 1
source2	Boolean	Read	18	Proximity sensor in source 2
p1running	Boolean	Read	19	True if process 1 is running
p2running	Boolean	Read	20	True if process 2 is running
p1sensor	Boolean	Read	21	Proximity sensor in process 1
p2sensor	Boolean	Read	22	Proximity sensor in process 2
button1	Boolean	Read	23	Button 1 state
button2	Boolean	Read	24	Button 2 state

Modelling with planner

In this example, we'll develop a CMAS model that utilises the integrated planner. This planner can be used to automate transportation operations, i.e., handling the movement of parts or resources. By employing this approach, we can avoid specific transportation goals and concentrate solely on manufacturing planning for parts. Additionally, this technique can be applied to resources, such as robot tools, e.g., the planner can then automatically swap tools on the robot as necessary. The planner requires; location interfaces, transportation interfaces and skill requirements. A location is a specialised interface that can serve as a start or end point. While a transportation is a interface that can be used to transport between two locations.

Begin by creating a new empty model in CMAS.

Create Parts

In this step we will create the two parts ProductA and ProductB. To better organise the project start by adding a new folder to the modelling tree and rename it to **Products**. Select the folder and enable the property *MakeInstances* by setting it to true.

Note

If *MakeInstances* is **false** the entity inside that folder will not be instantiated during run.

Inside that folder create two *Part* agents and name them **ProductA** and **ProductB**. For each product set *AutoDeploy* to **false** to make sure that products are not deployed

automatically during the start. We will later create our own logic for determining when a part can be deployed.

Interface

Open the *Interfaces* folder and create a new Interface for each product and name it **InterfaceSource**. This interface is to make sure that we can place a part in the **Source** resource later on. In the same way also create an *Interface* named **Crane** to enable the crane to handle a part.

Note

It is not necessary to create interfaces for the two processes as they will be handled by the goals.

Create template variable

Create a new folder in the modelling root and name it **Templates** make sure that *MakeInstances* property is set to **false**. Inside the **Templates** folder create a new *VariableObject* and name it **Location**. This will be used as a base for any location in the system. Open the **Location** variable and inside *Items* create two new *VariableInteger* and name them **x** and **y**.

Create Sources

Now create a new folder named **Resources** and set the property *MakeInstances* to true. Inside that folder create a new *Resource* agent and name it **Source1**. We want to make use of the proximity sensor to automatically deploy a new product agent when a product enters the source. The first step is to make sure that ***Source** can host a product by creating a new *InterfaceSpecialisation* named **InterfaceSource**. To be able to use the interface as a location change the property *SpecialisationName* to *Location*. Now we need a location variable, drag the template variable **Location** from the **Template** folder to the new interface folder called *Specialisation* and select *Copy*.

Note

It is also possible to manually create the *VariableObject* inside *Specialisation* but it can be more convenient to work with templates. If you select *Reference* instead of *Copy* a reference is made, if you then change the template all referenced objects will be updated accordingly. A template based variable can only have one value during modelling, e.g., an integer value can be 1, even if you create many references they will point to the same variable. However, during runtime each variable will become an unique instance and they can then have different values.

We also need to create two specialisation skills for the *Interface*. Create a new *Skill* inside the *Skills* folder, name it **pickPart** and change the *PostAction* to *DETACHED*. Also create a *Skill* named **placePart** with *PostAction* set to *ATTACHED*. These skills can now be used by the internal AI planner or by other resources to attach or detach from the source. For each skill we can add logic required to do the real attach/detach, however, for the source we can leave it empty since there is no specific logic required to attach/detach.

Note

It is also possible to create location interface directly by using *interfacelocation*. A *interfacelocation* will automatically get attach/detach skills and an empty *Specialisation* object.

The second step is to detect when a new part arrives, for that we can utilise the proximity sensor inside the source. To read the value of the sensor we need to add **sensor1** as a variable. Inside the *Variables* folder on **Source1** create a new *VariableBoolean* and name it **sensor1**. Select the new variable **sensor1** and set the property *source* to **modbus.tcp**: and change the *address* to **17**. The **sensor1** variable is now connected via MODBUS to source1 inside the simulation. Set the *ReadOnly* property to **true** since the sensor is an

input signal and it is not possible to write to it. Now we need to write the logic for the sensor. Open **Actions** editor for **Source1** and write the following program that will automatically deploy a part when the proximity sensor change state from **false** to **true**.

Source1 Action

```
program Source1;
var newPart:boolean;

// A general function to deploy a part and attach it
procedure Deploy(Name:string);
begin
    if not DeployAndAttach(Name) then
        error 'Unable to attach ' + agent.Name + ' to ' + Name;
end.

// Called when the Source is deployed
procedure OnDeploy;
begin
    newPart:=false;
end.

// Called cyclic
procedure Running;
begin
    if not agent.sensor1 then
        newPart:=false;
    else if agent.sensor1 and not newPart then begin
        // Part sensor at the source is true, lets deploy a new agent
        // that will represents the new product that arrived.
        Deploy('ProductA');
        newPart:=true;
    end;
    Sleep(50);
end.
```

Source 1 test run

It should now be possible to test the model by pressing run in CMAS.

Place a new a new Product inside the source by pressing the [generate] button inside **Source1** in the simulation. This will create a new part and activates the proximity sensor. Inside CMAS a new part agent **ProductA** should be deployed according to the logic in **Source1**. The product will be finished immediately and transferred to the **Finished** folder since it has no goals yet. You can see the part by expanding the run time tree and the **Finished** folder.

Create Crane

Inside the **Resources** folder create a new **Resource** agent named **Crane**. Add five **VariableInteger** in the **Variables** folder and map them to **setX**, **setY**, **atX**, **atY** and **vacuum** according to the **modbus table**. By setting **setX** and **setY** we can make the crane move and by reading **atX** and **atY** we can see the current position. The **vacuum** variable will turn on the vacuum cup to be able to pick up products. To control the crane we need to create two new **VariableInteger** named **targetX** and **targetY**. They will be used internally within the agent, hence, there is no need to map them to MODBUS. Now we need to write the

logic for the **Crane** controller. Open the *Actions* editor for **Crane** and write the following program.

Crane Action

```
program Crane;

var oldx, oldy : integer;

// During the first cycle in the run time
// we initialise the variables
procedure onRun;
begin
    agent.targetX:=agent.atX;
    agent.targetY:=agent.atY;

    oldx:=agent.targetX;
    oldy:=agent.targetY;
end.

// Main controller for the crane motion
// It will listen for changes of targetX and targetY
procedure running;
begin
    if oldx<>agent.targetX then begin
        agent.setX:=agent.targetX;
        oldx:=agent.targetX;
    end;

    if oldy<>agent.targetY then begin
        agent.setY:=agent.targetY;
        oldy:=agent.targetY;
    end;

    sleep(50);
end.
```

Crane test run

It should now be possible to test the model by pressing run in CMAS. The runtime tree should look like this.

In the runtime tree expand the *Crane* and the *Variables* folder and select the *targetX* variable. In the *Properties* window you should see the value for current position of the *Crane* (crane default position at start is 500). Change the value to 300 and press the *[TAB]* key. The crane should now start to move to 300 in the x direction.

Note

The values within the runtime viewer are not updated continuously. This is to prevent overloading the user interface with new values and to avoid slowing down agents. However, the values are updated each time you select a new entity.

Press stop to stop the runtime environment.

Crane transport Interface

In the next step we need to create an interface that can be used by other agents to command the crane to do transportation. The planner can also utilise the crane if we create a specialised transportation interface. To achieve this create a *InterfaceTransport* inside *Interfaces* and rename the new interface to **InterfaceCrane**. The new interface will include three *Specialisation* variables, **FromLocation**, **ToLocation** and **Step**. We must now make sure that the from and to variables are compatible with our location variables. Go to the **Templates** folder and expand *Items* inside the variable **Location**. Then drag **x** and **y** to the **FromLocation** variable in the new interface and drop them on *Items*. Now the **FromLocation** have a **x** and a **y** variable. Do the same for the **ToLocation**.

The **Step** variable is an integer that is used to detect if the transportation is approaching a pick (*Step* =1) or a leave position (*Step* =2).

Modify the transport skill

Now select the **Transport** skill inside *Skills* and rename it to **transportPart**. Double click it to open the skill editor.

The code now includes a function called **onNegotiate**. This function is invoked when the planner needs to verify whether the skills can manage transportation from the **FromLocation** to the **ToLocation**. It must return an **integer** value representing the cost to do the transportation, and **-1** if unable. By examining the coordinates on **remoteinterface.Specialisation.FromLocation** and **remoteinterface.Specialisation.ToLocation**, logic can be written to consider crane reachability. However, for this example, we can leave it blank as long as we return a cost of 0 (we can also use *this.cost* to get the cost of the skill). To observe the negotiation process, a **writeln** statement can be added inside the **onNegotiate** function.

```
function onNegotiate(Planning : boolean): integer;
begin
    writeln('onNegotiate for ', agent.name, '.transportPart() from ',
    remoteinterface.Specialisation.FromLocation,
    ' to ',
    remoteinterface.Specialisation.ToLocation);

    onNegotiate:=0;
end.
```

The next step is to create functions to support the reading of the signals. Enter the following functions in the skill. To make these functions available for all other functions write them at the top directly after the *program* statement.

```
function isMowing(): boolean;
begin
    isMowing:=NOT (agent.targetX=agent.atX AND agent.targetY=agent.atY);
end.

function isAtPosition(x, y:integer): boolean;
begin
    isAtPosition:= x=agent.atX AND y=agent.atY;
end.

function isAtTarget(): boolean;
begin
    isAtTarget:= agent.targetX=agent.atX AND agent.targetY=agent.atY;
end.
```

The next step is to create the transportation skill. In this example, a **SAFE_HEIGHT** parameter is used to ensure the crane's motion is safe and free from collisions. Note that there are many ways to implement the transportation skill.

Regardless of the method, it is crucial to remember that the **OnRun** function should not take too long to execute due to its real-time nature, as this could compromise real-time performance. Therefore, we need a strategy to execute the skill, as the crane requires time to complete the transportation. If the skill requires additional time to complete, **OnRun** should continuously return *processplan.running* until the task is finished. CMAS will then call the

function cyclically. Once the skill is complete, **OnRun** can return *processplan.done* to indicate the task's completion.

In this example, the function **pickOrPlace** tracks the sequence of actions and performs only one step at a time. This allows the **OnRun** function to iteratively call **pickOrPlace**, maintaining real-time performance.

```
Program transportPart;

var constant SAFE_HEIGHT:=150 : integer;

function isMowing(): boolean;
begin
    isMowing:=NOT (agent.targetX=agent.atX AND agent.targetY=agent.atY);
end.

function isAtPosition(x,y:integer): boolean;
begin
    isAtPosition:= x=agent.atX AND y=agent.atY;
end.

function isAtTarget(): boolean;
begin
    isAtTarget:= agent.targetX=agent.atX AND agent.targetY=agent.atY;
end.

function onNegotiate(Planning : boolean): integer;
begin
    writeln('onNegotiate for ', agent.name, '.transportPart() from ',
        remoteinterface.Specialisation.FromLocation,
        ' to ',
        remoteinterface.Specialisation.ToLocation);

    onNegotiate:=0;
end.

var picStep:=0:integer;
function picorplace(posX, posY :integer; vacuum:boolean) : boolean;
begin
    picorplace:=false; // Default value

    case picStep of
    0:
        if isMowing() then error 'crane is mowing';

        // Check if we need to move to a safe height or not
        if agent.atY>=SAFE_HEIGHT then
            picStep:=30;
        else begin
            agent.targetY:=SAFE_HEIGHT;
            picStep:=10;
        end;

    10: // Wait until moving
        if isMowing() then picStep:=20;

    20: // Check if we are at a safe height
        if isAtTarget() then picStep:=30;

    30: // Start to move in x direction
        if isAtPosition(posX, agent.atY) then
            picStep:=70;
        else begin
            agent.targetX:=posX;
            agent.targetY:=agent.atY;
            picStep:=50;
        end;

    40: // Wait until crane starts to move
        if isMowing() then picStep:=50;

    50: // Wait until move x is ready
        if isAtTarget() then picStep:=70;
```

```

70: // Check if we need to move y direction
    if isAtPosition(agent.atX, posY) then
        picStep:=100;
    else
        picStep:=80;

80: // Move in y direction
    agent.targetX:=agent.atX;
    agent.targetY:=posY;
    if isMowing() then picStep:=90;

90: // Wait until move y is ready
    if isAtTarget() then picStep:=100;

100: // Activate vacuum
    agent.vacuum:=booleantoint(vacuum);
    picStep:=110;

110: // Check if we need to move to a safe height or not
    if agent.atY>=SAFE_HEIGHT then
        picStep:=140;
    else
        picStep:=120;

120: // Move to a safe height before we end
    agent.targetY:=SAFE_HEIGHT;
    if isMowing() then picStep:=130;

130: // Check if we are at a safe height
    if isAtTarget() then picStep:=140;

140: // We are done
    picorplace:=true;
    picStep:=0;
else:
    error 'wrong state: ' + inttostring(picStep);
end;
end.

function onRun() : integer;
begin
    onRun:=processplan.failed;
    if interface.Specialisation.step=1 then begin
        if picorplace(interface.Specialisation.FromLocation.x,
interface.Specialisation.FromLocation.y, true) then
            onRun:=processplan.done;
        else
            onRun:=processplan.running;
    end;
    else if interface.Specialisation.step=2 then begin
        if picorplace(interface.Specialisation.ToLocation.x,
interface.Specialisation.ToLocation.y, false) then
            onRun:=processplan.done;
        else
            onRun:=processplan.running;
    end;
end.

```

Create Processes

Inside the *Resources* folder create a new *Resource* agent named **Process1**. Now, add three agent variables as boolean,

1. the proximity sensor for process 1, name it **proximitySensor** and connect it to address 21
2. the start signal for process 1, name it **startProcess** and connect it to address 4
3. the running indicator for process 1, name it **processRunning** and connect it to address 19

Interfaces

Process 1 need to be compatible with the two products and to do that we need to add an interface. Select *InterfacePart* for *Source1* and right click in the modelling tree and select *copy* . Now select *Interfaces* inside *Process1* and right click in the tree and select *paste* and rename it to **InterfaceProcess** .

The next step is to create a skill on *Process1* that will start the process. Select *Skills* and create a new *Skill Structured Text* and name it **runProcess1** . Edit the *skill* and enter the following program.

```
Program runProcess1;

var processState:=0: integer;

function onRun() : integer;
begin
    // Set onRun return value to running to indicate that we want more time
    // to execute the skill. This will be used as the default return value.
    onRun:=processplan.running;
    sleep(50);

    // Execute the sequence of the skill
    case processState of
        0: // Check that it is ok to run
            writeln('Start to run ' + agent.name);
            if agent.proximitySensor and not agent.processRunning then begin
                agent.startProcess:=true;
                processState:=10;
            end;
            else
                error 'process not ready missing a part';

        10: // Wait for feedback that the process is running
            if agent.processRunning then begin
                agent.startProcess:=false;
                processState:=20;
            end;

        20: // Wait until process is done
            if not agent.processRunning then processState:=30;

        30: // We are done
            processState:=0;
            writeln(agent.name + ' is done');
            // Change the default return value to done to signal that the skill is ready
            onRun:=processplan.done;

            else:
                error 'wrong state: ' + inttostring(processState);
            end;
    end.
end.
```

To be able to trigger the built in AI planner to start we need to add a requirement to the *skill* . Select the folder *Requirements* in the skill **runProcess1** and add a new. Name the new requirement **MustBeAttached** and change the *WorkFlowType* to *ATTACHED* . This indicates that any one that want to run the skill **runProcess1** must first be attached to the interface to do so.

Now create **Process2** in the same way and map it to the simulation.

Create Sink

Inside the **Resources** folder create a new *Resource* agent named **Sink**. For the **Sink** create a new *InterfaceSpecialisation* named **InterfaceSink**. In the same way we did it for the source or processes change the property *SpecialisationName* to *Location* and drag the template variable **Location** from the **Template** folder to the new interface folder called *Specialisation* and select *Copy*. Also make sure that you have a pick and place skills in the same way as before.

Now, create a new *Skill* on the interface and name it **packing**. Since the sink won't perform any actions, we can leave the skill unchanged. However, do not forget to add an *ATTACHED* requirement on the **packing** skill.

Set location values

The model is almost done but we need to set coordinates for all our locations. The locations are used by the planner to command the crane to do transports. Start by opening the **Location Specialisation** variable on the **Source1** interface **InterfaceSource** and change the **x** value to 55 and the **y** value to 82.

Entity	Variable	Value
Source1	Location.x	55
Source1	Location.y	82
Process1	Location.x	440
Process1	Location.y	82
Process2	Location.x	640
Process2	Location.y	82
Sink	Location.x	945
Sink	Location.y	82

Goals

Double-click on **ProductA** to open the **Goal Chart** view. Now create three goals, right-click in the top-left box and select **Add New Goal**.

For each goal, set the following values:

Name	Interface name	Process plan name
GoalProcess1	InterfaceProcess	runProcess1
GoalProcess2	InterfaceProcess	runProcess2
GoalPacking	InterfaceSink	packing

Now create a similar goal chart for ProductB.

Note

We have now created process plans for **ProductA** and **ProductB** that focus solely on the objectives related to the products themselves. This approach allows us to disregard skills, such as transportation, that are not relevant from the product's perspective.

Running the example

When you press **Run** to start the model, CMAS will connect to the simulation based on the variables you have defined. If you place a **ProductA** in **Source1** by clicking the **[Generate]** button in the simulation, the **Source1** agent in CMAS will detect that and deploy a new **ProductA** agent. This agent will then begin executing its goals sequentially.

For example, the first goal, **GoalProcess1**, will initiate a negotiation process to find a resource capable of completing the task **runProcess1**. In this case, **Process1** will respond, "I can do it, but I have a requirement that you must first be physically attached to me".

This requirement will trigger the planner to search for a solution to attach **ProductA** to **Process1**. The planner begins by creating a map of all possible movements for products, materials, and resources. This map is then used to identify a solution that satisfies the requirement.

You can view the map generated by the planner in the console by enabling **PathFinder.ShowMapResults** in the preferences. Additionally, you can see the solution identified by the planner by enabling **ShowPathFinderResults**. This can be useful for identifying errors or mistakes in the model.

Variable adapters

Introduction

Variables can be used to handle values inside an agent. With a variable adapter it is possible to access external data. Example of external data might be a sensor value from a transducer. An Adapter, is a low level implementation, to be able to handle different communication protocols.

Supported adapters are:

- **OPC UA**
- **REST**
- **FILE**
- **MODBUS**
- **SOCKET**
- **WEBSOCKET**
- **ZEROMQ**

Any variable can be connected to an adapter by specifying the **Source** and the **Address** property. **Source** is the path to the device to connect to and has a format of [ADAPTER TYPE] : {optional parameters}

Note

An adapter is shared among many variables if they have the same **Source**.

Address is the address to the specific data to read from a **Source**. An address shall not be reused or shared between variables.

If the first parameter before adapter type is **DEBUG:**, e.g. debug:opc.tcp:localhost, it is possible to get more debug information. This is useful if a connection is not working correctly.

OPC UA

For an OPC UA adapter the specific format for **Source** is:

`opc.tcp:[HOST]:[PORT]:[NAME]`

- `opc.tcp` is static for OPC UA
- `HOST` is the name or TCP/IP address to the server
- `PORT` is the port number to the server

- NAME is the server name instance

OPC UA Source example:

```
opc.tcp:localhost:53530:OPCUA/SimulationServer
```

The **Address** indicates which variable to read. If **Address** is empty or set to AUTO the name of the variable will be used to find the variable inside the **Source**. Otherwise, **Address** must be a valid node name.

OPC UA address example:

```
ns=3;i=1006
```

REST

For adapter type of REST the specific format is:
REST.[type]:[HTTP]:[HOST]:[PORT]:[AUTH]:[USER]:[PASW]
 eg,
REST:http:localhost:80:DIGEST:Default

- REST is static for the rest protocol
- TYPE is optional (default non) can be set to .ASSOCIATE
- HTTP http or https
- HOST is the host name or TCP/IP address to the server
- PORT is the port number to the server
- AUTH authorization level, BASIC, DIGEST or NONE (default value NONE)
- USER user name
- PASW password

By default, a REST adapter uses static references for the Source and Address properties. However, you can associate the local available variable value with the request before performing a read operation. To enable this behavior, use the REST.ASSOCIATE type.

Example: To connect to an ABB robot

```
REST:http:localhost:80:DIGEST:Default User:robotics
```

The **Address** indicates which variable to read and should normally start with a slash /

Example:

```
/rw/rapid/symbol/data/RAPID/T_ROB1/MainModule/
```

Example 2: In this example, CMAS connects to an Ollama server via REST to access a large language model. A string variable, named llama, exists in the Agent Variable relation with the following source and address settings:

```
Source = REST.ASSOCIATED:HTTP:localhost:11434
Address = /api/generate
```

To use the ASSOCIATE behavior, we need to set the variable's value. An onRun procedure for the agent can look like this:

```
{ Rest associated variable example with ollama REST server }
Program Example;

// Create a struct type for ollama
struct REQUEST;
```

```

begin
    var model : string;
    var prompt : string;
    var stream : boolean;
end;

// Initialize a variable by the REQUEST type
var aRequest:=(model:='llama3.2', prompt:='', stream:=false) : REQUEST;

procedure onRun;
var
    result, response: string;
begin
    // Set prompt value
    aRequest.prompt:='Hello World!';

    // Associate a JSON object to the llama variable
    // This associated value will be used when reading from the REST adapter later on
    agent.llama := structtoj(aRequest);

    // Read the value from the REST server by associating the value
    // This value is a JSON object
    result := agent.llama;

    // Get the response value from the JSON object
    response := GetJElement(result, 'response');

    if len(response) = 0 then
        writeln('Response is empty: ' + result);
    else
        writeln(response);
end.

```

FILE

For a FILE adapter the specific format for **Source** is:

`file:path`

- file is static
- path is a directory

Example:

`file:C:\Users\Foo\Desktop\communication\`

The **address** indicates which file to open without any path information given

Example:

`forwardA.qx`

MODBUS

For a MODBUS adapter the specific format for **Source** is:

`modbus.<TYPE>:[HOST|SERIALPARAMETERS]:[PORT]`

<TYPE> can be:

- tcp
- udp
- serial

HOST is the name or TCP/IP address to the server (default 127.0.0.1)

PORT is the port number to the server (default 502)

SERIALPARAMETERS is the configuration parameters for a modbus serial connection type. It has the following format:
<SERIALPORT>,<BAUDRATE>,<DATABITS>,<STOPBITS>,<PARITY>,<FLOWCONTROLIN>,<FLOWCONTROLOUT>

- <SERIALPORT> is the name of the serial port.
- <BAUDRATE> (default 9600) is the baud rate for the connection.
- <DATABITS> (default 8) number of data bits.
- <STOPBITS> (default 1) number of stop bits.
- <PARITY> (default 0) 0 (non) / 1 (odd) / 2 (even) / 3 (mark) / 4 (space)
- <FLOWCONTROLIN> (default) 0 (disabled)
- <FLOWCONTROLOUT> (default) 0 (disabled)

MODBUS Source example for a TCP connection:

```
modbus.tcp:127.0.0.1:502
```

MODBUS Source example for a serial port:

```
modbus.serial:com1,9600,8,1
```

The **Address** indicates which variable to read. <unit>:<address>:<type>
<unit> is optional (default 1) and is the device unit number. <address> is the address number to read or write. <type> is optional (default holdingregister) and state which type of register to read <type> can be: holdingregister or 4 inputregister or 3 discreteinput or 2 coil or 1

MODBUS address example to read a holding register from unit 1 at address 4:

```
1:4:holdingregister
```

By the use of default values the same register as above can be simplified to:

```
4
```

SOCKET

For a SOCKET adapter the specific format for **Source** is:

```
socket.<PROTOCOL>.<TYPE>.<RW>:[ SERVER ]:[ PORT ]:[ TIMEOUT ]
```

<PROTOCOL> can be:

- tcp
- udp

<TYPE> can be:

- ascii (default value)
- binary

<RW> can be: RW for read and write (default value) R for read only W for write only

SERVER is the name or address to the server (default 127.0.0.1)

PORT is the port number to the server (default 80)

TIMEOUT is the time out in ms for reading value from the socket (default 2000)

\

SOCKET Source example for a TCP connection with read and write access:

```
socket.tcp.ascii:127.0.0.1:80
```

The **Address** indicates how to read and write values and is a formating string. Note, the address field is only used for ASCII based sockets.

Format Specifier	Data Type	Output
a	floating point	Hex output of floating point number
b	Any type	True or False
c	character	Unicode character
d	integer	Decimal Integer
e	floating point	a decimal number in scientific notation
f	floating point	decimal number
g	floating point	decimal number, possibly in scientific notation depending on the precision and value
o	integer	Octal number
s	Any type	String value
t	Date/Time	t is the prefix for Date/Time conversions.
x	integer	Hex string

The format string receives two values: Name and Value, derived from the variable object.

Example 1: Integer Variable For an integer variable A, if the address is:

```
set %1$s = %2$i
```

And the variable A is set to 1, the following will be sent:

```
set A = 1
```

A read operation will generate a request, provided the address is not empty. The request is assembled in the same way as for a write operation, and it is sent before the read begins. Note, The read operation will always wait until a response string is received or a timeout occurs. Therefore, a read operation may cause the application to freeze if no response is received.

Example 2:

```
get %2$s
```

WEBSOCKET

Web Socket - not implemented

ZeroMQ

For adapter type of ZeroMQ the specific format is: ZMQ.type:[TCP]:[//host]:[port] eg, ZMQ:tcp://*:5555

Type can one of: DEALER, PAIR, PUB, PULL, PUSH, REP, REQ, ROUTER, STREAM, SUB, XPUB, XSUB

Agent action

An agent, e.g. a resource agent, is associated with a property *AgentState* (from *AgentAction*). The *AgentState* indicates which state an Agent is at during runtime. The main states are;

State name	Description
Depot	Agent is ready to be deployed
Running	Agent is running
Failed	Agent has failed
Finished	Agent is finished

These states are visualised in the runtime viewer tree during runtime. Note, that there exist more states (see state chart figure below). When entering a state, a pulse (P) is generated and a procedure is called once, e.g, `onDepot()`. These procedures do not exist as default but can be defined inside *AgentAction* as a user-defined procedure.

```
procedure onDeploy;
begin
    writeln('Resource ', agent.Name, ' is now deployed.');
end.
```

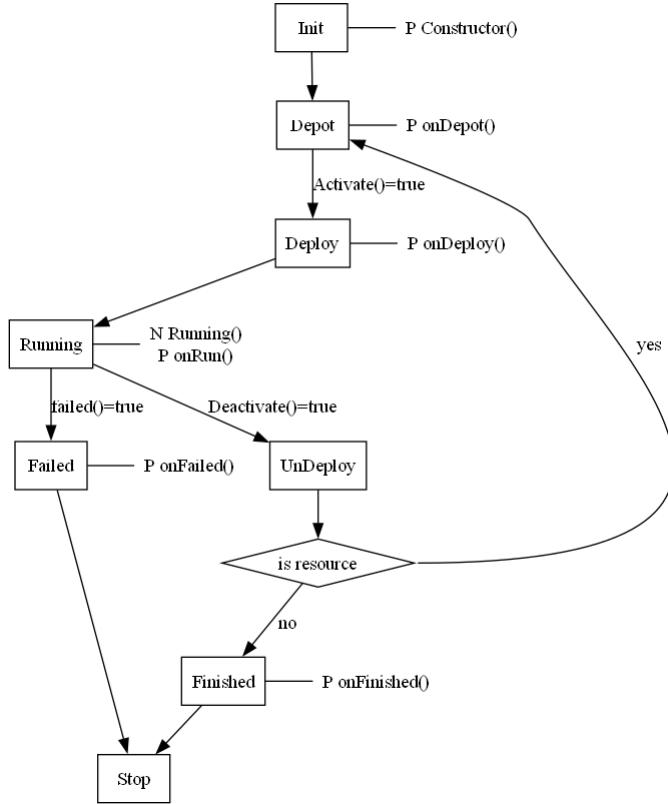
The procedure `Running()` is a normal (N) procedure, a procedure that is constantly called, i.e., for each cycle the procedure is called and is expected to return immediately.

Note

It is not a good approach to use functions, such as `writeln()`, to continuously produce output from the procedure `Running()` since it will overload the system with information to display.

The following example will use the `Running` to indicate that the agent is running. In this case a better approach to write a single message might be to utilise the (P) procedure `Run()` instead.

```
var runOnce:=false : boolean;
procedure Running;
begin
    if not runOnce then
        begin
            writeln('Running');
            runOnce:=true;
        end;
end.
```



Deployment

Automating deployment

There are several strategies to automate deployment in CMAS.

1. **Batches**
2. **Deployment Agents**
3. **User defined logics for deployment**

Creating and Using Batch Objects in CMAS

In CMAS, you can create a **batch** object to specify which agents to deploy. You can also define whether each deployed agent should be **attached** to a specific interface during deployment.

Attachments are particularly useful when using the **Planner** agent, as the planner utilizes these attachments to coordinate transport planning.

How to Create a Batch Object

1. Open the **Modelling** tab.
2. In the **Others** group, click **Batch** object.

This will create a new batch object in the modelling tree.

Opening the Batch Editor

- To open the **Batch Editor**, double-click the new batch object in the modelling tree, or
- Select the batch object and choose **Batch Set** in the **Property Viewer**.

Using the Batch Editor

- **Left Side** : Displays available agents for deployment.
- **Right Side** : Displays the current batch configuration.

To add an agent to the batch:

1. Drag an agent from the left side and drop it into the batch on the right.
2. You can add the same agent multiple times to create multiple instances.

To specify an attachment:

- Double-click an agent in the batch list to open its settings and assign an attachment.

Running

When CMAS is in **Run** mode, you can deploy a batch object from the **Depot**. If the batch contains multiple agents, all of them will be deployed. If any attachments were specified during setup, those agents will be attached accordingly.

Note: You can deploy a batch as many times as needed. Each deployment creates new instances of the agents defined in the batch.

Creating and Using Deployment Agents in CMAS

Deployment Agents provide more options and control than a standard **Batch** object.

How to Create a Deployment Agent

1. Open the **Modelling** tab.
2. In the **Agents** group, click the **DeploymentAgent** object.

This will add a new deployment agent to the modelling tree.

Working with Batches in a Deployment Agent

A deployment agent has a relation called **Batches**. Expand the deployment agent node in the modelling tree to view and manage its associated batches.

You can create as many batches as needed here. Batches are created the same way as described in the **batches section**.

Note

a deployment agent must at least have one associated batch object to work.

Deployment Agent Properties

Deployment behavior is controlled using properties available in the **Property Viewer**.

NumberOfTimes

Defines how many times the deployment process should be repeated across all batches. Once the specified number of repetitions is completed, the deployment agent will automatically undeploy itself.

- 0 means infinite repetitions. In this case, the deployment will continue until the deployment agent is either manually undeployed or stopped.

VariableTrigger

The name of an agent variable that triggers deployment.

- The variable can be of type boolean, integer, or real.
- Deployment is triggered when the variable's value changes to something other than false (for booleans) or 0 (for numeric types).
- If left empty, the variable trigger will be ignored.

DeploymentTime

The deployment time is relative to the start or the previous deployment (if multiple).

- If time is < 0, the time is ignored and only the **VariableTrigger** is used.
- By default, the time value is interpreted in **milliseconds**. You can also specify units:
 - Use h for hours (e.g., 1h)
 - Use m for minutes (e.g., 30m)
 - Use s for seconds (e.g., 1s)

DeployAt

Determines when deployment should occur. Available options:

- **TIME** – Deployment waits for the specified time or trigger condition before each repetition (up to *NumberOfTimes*).
- **INTERVAL** – Deployment happens immediately, then repeats based on the time or condition (up to *NumberOfTimes*).

Note

Deployment of **DeploymentAgents** is controlled by the RunTime\DeployDeploymentAgentsOnStart setting in the preferences. If this value is set to false, DeploymentAgents will not be deployed automatically at startup and must be deployed manually or by some logic.

User defined logics for deployment

If the default behavior of the **Deployment Agent** is not sufficient, there are several ways to implement custom deployment logic. It is hard to cover all possibilities but two main strategies are explained:

- **Centralized**, managed by a dedicated deployment agent that coordinates deployment across other agents.
- **Local**, applied directly to a specific agent's behavior.

Centralized Deployment Agent

By creating a deployment agent with an agent variable and setting **VariableTrigger** to that variable, you can control deployment directly from within the agent's user-defined functions.

Example: Create a deployment agent with a boolean agent variable named `trigger`, and set **VariableTrigger** to `trigger`.

To disable the default deployment logic, set **DeploymentTime** to 0 and **NumberOfTimes** to 0.

Next, add agent actions by double-clicking the deployment agent. The following example will ask the user for the batch size and by default triggers five deployments, each separated by 2 seconds. This replicates the behavior of setting **DeploymentTime** to 2s and **NumberOfTimes** to 5. However, using a variable trigger allows for more complex, customized deployment logic.

Note: The trigger variable can also be connected to an adapter to receive signals from external systems.

```
program DeploymentAgent;

procedure onRun;
var i, size : integer;
var result : string;
begin

    result:=InputDialog('Batch size', '5');
    if not StringIsEmpty(result) and StringIsInteger(result) then
        size:=StringToInt(result);
    else
        error 'Wrong batch size';

    for i:=1 to size do
    begin
        agent.trigger:=true;
        agent.trigger:=false;
        sleep(2000);
        writeln('deployed');
    end;
end;
```

```
// Make sure to undeploy the deployment agent
undeploybyinstanceid(this.InstanceID);
end.
```

Local Deployment

The simplest way to apply local deployment behavior is to open an agent's **Actions Editor** and implement the `onDeploy` procedure.

Example 1: In this example, we will implement a local deployment behavior on a 'Part' agent. Whenever a Part agent is placed in the depot, the `onDepot` procedure is triggered. The agent begins by waiting for 2 seconds, then deploys itself by using the `deployByInstanceID()` function. When a part is deployed, a new part is created in the depot, which repeats the process resulting in continuous, timed deployments.

```
program Part;

procedure onDepot;
begin
    sleep(2000);
    deployByInstanceID(this.InstanceID);
end.
```

Example 2: In this example, a dialog is created when the Part agent is placed in the depot. If the user presses the Deploy button, the agent will deploy itself and close the dialog. A new agent will then be created automatically, which in turn shows a new dialog continuing the cycle. If the user presses Stop, the entire runtime environment will shut down.

```
program Part;

procedure onDepot;
var Selection:=-1:integer;
begin
    OptionDialog(1, Selection, 'Press a button', 'Selection', 'Deploy', 'Stop');

    // Wait for user input
    repeat
        sleep(50);
    until Selection=-1;

    // Close dialog
    OptionDialog(1, false);

    if Selection=1 then
        deploybyinstanceid(this.InstanceID);
    else if Selection=2 then
        stopsystem();
end.
```

Interfaces

Every agent possesses interfaces through which all interactions between agents occur. An interface comprises variables and skills. Agents can have multiple interfaces, and the purpose of an interface is to facilitate communication solely with compatible agents. In other words, for two agents to interact, they must define an identical basic interface, with matching variable names. For example, a robot might have an interface with three variables: `x`, `y`, and `z`,

along with two skills: pickAtLocation and leaveAtLocation. Any other agent that intends to utilize these skills must provide a compatible interface that defines x, y, and z. The compatibility of variables is determined by matching the name, variable type, and range. By incorporating the range, it becomes feasible to restrict x, y, and z to the robot's workspace. The treatment of variables across all interfaces can be fine-tuned. To achieve this, select the interface in the modeling tree and access the properties view to modify the "VariableMatching" property. The following options are available:

Option	Description
Required	All variables must exist and match to be compatible
Optional	If a variable exists on both interfaces it must match, otherwise it is ignored
Ignored	All variables are ignored

Note

When matching two interfaces A and B, both sides are taken into account. Consequently, if interface A is set to "Ignore" while interface B is set to "Required," the "Required" option will be considered for the matching process.

Defining goals is crucial for a part agent, and each goal serves as an interface to interact with resources. Hence, there might be a need to include variables in a goal and establish specific values for those variables. The name of the goal interface can be set in the properties view or in the goal chart viewer.

Note

A goal on a part cannot define a skill.

When a goal is defined with three variables (x=1, y=2, and z=3) and requires the skill "pickAtLocation," the part agent will then search for all compatible process plans that match these criteria. If more than one process plan is found the one with the lowest cost is selected.

For more information see:

- **Interface action**
- **Negotiation**
- **Variable bounds**
- **Goal**

Interface action

When an agent requires assistance from another agent at runtime, a search is conducted to locate a suitable remote agent interface. To ensure the appropriateness and compatibility of the sought agent interface, a negotiation process starts between the agents. This negotiation process entails agents striving to establish an agreement on how to effectively cooperate and synchronize their respective skills. An interface serves as a means of depicting compatibility, i.e., when two interfaces match they can be treated as compatible.

The functions of an interface, all are optional except onRun():

Function	Description
onRun() : integer	Called to execute the process plan
onNegotiate(planning:boolean) : integer	Called during negotiation, if planning is true this is done during planning and all runtime variables might not be updated
onBook() : boolean	Called when asked to book

onUnbook()	Called when asked to unbook
------------	-----------------------------

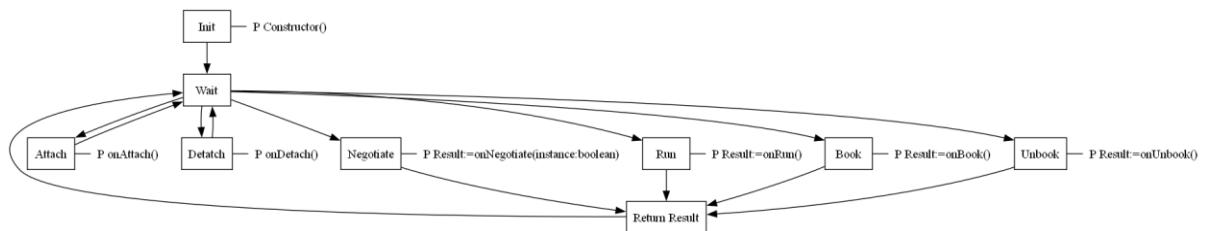
Note

onNegotiate on an interface is called before onNegotiate is called on each **skill** on the same interface. If onNegotiate does not return a value ≥ 0 the call to each skill is ignored. The returned value from onNegotiate for an interface is included in total cost during negotiation.

The procedures of an interface (they must be declared as procedures) and are optional:

Procedure	Description
onAttach	Called when an interface has been attached
onDetach	Called when an interface has bee detached

Each function or procedure can be overloaded to replace the built in default mechanism.



In the following example a process plan is created to make two holes. It also demonstrate how to make loop where each iteration must return back the control to the system to avoid time out.

```

Program Processplan;
abstract aResource : interface;
var steps: integer;

function onNegotiate(planning:boolean) : integer;
begin
  if remoteinterface.diameter=5.0 then
    onNegotiate:=this.cost;
  else
    onNegotiate:=-1; // refuse by returning a negative cost
end.

function onRun() : integer;
begin
  if steps=0 then
  begin
    aResource.x:=10.0;
    aResource.MakeAHole();
    onRun:=processplan.running;
  end;
  else if steps=1 then
  begin
    aResource.x:=20.0;
  end;
end.
  
```

```

        aResource.MakeAHole();
        onRun:=processplan.running;
    end;
else
begin
    steps=0;
    onRun:=processplan.done;
end;
steps:=steps+1;
end.

// Constructor
begin
    steps:=0;
end.

```

Negotiation

Interfaces

A skill represents an action or task executable by a resource agent during runtime. Typically, a skill demands additional information, represented through variables. In CMAS, an agent's **interface** has the role of maintaining both a set of variables and a set of skills as one unit. This unit serves as a connection point and all collaborations are performed through **interfaces**.

For an interface to be operational it must at least possess one skill.

Note

An interface serves as a means of depicting compatibility, i.e., when two interfaces match, they can be treated as compatible.

Collaboration

When an agent requires assistance from another agent at runtime, a search is conducted to locate a suitable remote agent interface. To ensure the appropriateness and compatibility of the sought agent interface, a negotiation process starts between the agents. This negotiation process entails agents striving to establish an agreement on how to effectively cooperate and synchronize their respective skills. During the modelling phase, the ability to reference the utilization of compatible remote interfaces is essential, this without the need to specify any particular agent or interface. This is achieved through abstract interfaces in CMAS. Hence, an abstract interface represents a desired remote interface during the modelling process and lacks any direct connection to any specific remote interface. However, during runtime, it will be linked with the best available remote interface and transformed into a real interface during the negotiation process. A linked interface can then be used to interact with other agents, e.g., to request a skill to start and send variable values.

The following abstract interface types exist:

Typ	Description
Abstract	A pure abstract interface. All local interfaces and all remote interfaces are considered during the negotiation process to find the best pair of one local and one remote interface.
Define	Only one local interface is used (the defined one) to find the best compatible remote

	interface
Static	A remote interface is static mapped to any local interface. The remote interface must be known by ID at the modelling phase.

The negotiation process

A negotiation is performed the first time a variable on a declared abstract interface is accessed (this step is ignored if handle is specified for the abstract interface) or if a skill is called on an abstract interface. It is also possible to do it manually by calling the negotiate() function on an abstract interface. For more details see, [how to define and use abstract interfaces](#).

The negotiation process is conducted through the following sequential steps, for all local interfaces:

1. Compile a ACL message of required skills and the corresponding interface variables.
2. Generate a list of all remote interfaces that are compatible with any of the local interfaces through **interface matching**.
3. If ignore busy is true remove all busy interfaces from the list. Ignore busy is default set to true but can be change by the function **ignorebusy**.
4. If ignore booked is true remove all booked interfaces from the list. Ignore booked is default set to true but can be change by the function **ignorebooked**.
5. Dispatch an ACL query, "negotiate", to all compatible remote interfaces, enclosing the list of obligatory skills and variables. This will trigger the **onNegotiate()** call on all remote interfaces.
6. In the event of an interface responds with a value ≥ 0 (AGREE with a cost) that interface is appended to a list of potential interface combinations, i.e., saving the local and remote interfaces as a pair with a cost.
7. If an interface responds with a negative number (DISAGREE) it is disregarded.
8. If CheckForSafetyOverlaps is true, try to book all overlapping interfaces. If it fails for an interface unbook it and ignore it.
9. From the list of potential interface pairs (local, remote) select the pair that generates the lowest cost.

Note

If the interface is a `defined` abstract interface the negotiation process is only conducted on the `define` interface and the rest of the local interface will be ignored.

Interface matching

The process of identifying matching interfaces is carried out through the following steps:

1. Gather a list of active and deployed agents within the domain.
2. For each active agent, extract all interfaces that share the same name as the local interface.
3. Verify the alignment of variables (**variable matching**) between the local and remote interfaces; any mismatched interface is removed from the list.

Variable matching

The procedure for checking the compatibility of two variables within an interface is performed as follows:

1. Verify that the names are identical.
2. Validate that the types match.
3. Ensure that the local value falls within the bounds of the remote variable.

Note

Regarding the third step for bound verification:

- This verification is ignored if both upper and lower bounds are empty.
- It can be fixed to a specific value by setting the upper and lower bounds to the same value, e.g., 5.
- Alternatively, it can be defined within a range by setting distinct upper and lower bounds, e.g., 1 and 5.
- Another option is to establish a **bound expression**.
- The behaviour is also influenced by the "**VariableMatching**" property of the interface.

Details

- For an abstract interface the following appliesIf the program that declares the abstract interface is called as a skill on an interface: all interface values will be copied (if possible) to the abstract interface as an automatic transfer of data. This step is ignored if the abstract interface is declared as handle.
- A negotiation will be triggered if a variable on the declared abstract interface is accessed. This step is ignored if handle is specified for the abstract interface.
- If a skill is called on the abstract interface and the interface is unnegotiated a negotiation process is triggered.
- If an integer variable is given as a parameter to a skill call on an abstract interface the skill will be started in parallel and the program will not wait for the result. Instead, the integer parameter will reflect the status of the skill call.
- If handle is specified all abstract interfaces remain static mapped after the first negotiation this can be avoided by calling the negotiate() function on the abstract interface.
- It is possible to trigger a negotiation by calling <interface>.negotiate() or <interface>.agree() A call to negotiate() will try to negotiate with available resources to find a suitable match if it fail the execution will stop with an error. If agree() is called instead, the same negotiation process will start and it will retry until an valid agreement or search forever.
- It is possible to book or unbook an interface by calling the <interface>._book() or <interface>._unbook(). A booked interface will remain booked until it is unbooked.
- If a skill is called and that skill fails the execution will be aborted and an error is thrown. This step is ignored if the interface is declared as handle and instead a return value is returned.
- It is possible to combine negotiation and booking in one call by <interface>.negotiateandbook() or <interface>.agreeandbook()

For an abstract interface the following static local functions exist:

Function	Book
negotiate()	no
negotiateandbook()	yes
agree()	no
agreeandbook()	yes

The negotiate() or negotiateandbook() will make a single attempt to locate a compatible remote interface from the accessible resources. If this proves unsuccessful, the execution will halt with an error unless it was declared with the modifier handle. When handle is specified the return value will be failed or done. The agree() or agreeandbook() will try forever to locate a compatible remote interface from the accessible resources. This is good approach if resources can be plugged in or out as in the Plug & Produce concept. When an interface is booked it must be unbooked when no longer needed. To unbook an interface call the _unbook() function on the abstract interface. The behaviour of negotiate and agree on an interface can be

modified with **ignorebusy()** or **ignorebooked()**.

Note

If a remote skill is called on an uninitialised abstract interface an agree() is automatically triggered. If the interface is declare with the handle modifier a negotiate() call is triggered instead. This can be avoided by calling any of negotiate() or agree() functions before. It is also possible to re-negotiate an abstract interface later on by calling negotiate() or agree() again.

Negotiation and Booking

It is possible to use an abstract interface directly. In the following example an abstract interface a is declared. When the skill call doSomeThing() is called it will trigger a negotiation. That negotiation process will try to map any local interface to a compatible remote interface before executing the skill.

```
Program example;
abstract a : interface;

function onRun() : integer;
begin
    a.doeSomeThing();
    a.doeSomeMore();
end.
```

During the call to doSomeThing() the remote interface will be busy and no other calls can be made to the same interface during that time. However, it is possible for some one else to access the remote interface before we reach the doeSomeMore() call. If we want to avoid anyone else to access the interface in between we need to book the interface. We can keep the remote interface booked as long as we want to but it is necessary to unbook it when we are done to let other agents to use the same interface.

```
Program example;
abstract a : interface;

function onRun() : integer;
begin
    a.agreeandbook(); // Will negotiate and book (will never fail, but might take some
time)
    a.doeSomeThing();
    a.doeSomeMore();
    a._unbook();
end.
```

For an abstract interface mapped to a remote interface all remote skills are available. A call to a remote skill is considered as asynchronous and is running in parallel. However, a remote interface also have some static synchronous functions:

Function	Description
book()	Will try to book the remote interface
unbook()	Will try to unbook the remote interface
_isbookedbyme()	Will return done if the remote interface is booked by the interface
_isbooked()	Will return done if the remote interface is booked by any one

It is possible to call _book() manually to try to book an interface. Usually it is a better

approach to use the local negotiateandbook() or agreeandbook() function to book instead. However, if _book is used it is possible to write your own logic, e.g., how many retries and when to give up. The global parameter RetryOnBusy and the abstract interface modifier handle will affect the behaviour of _book(). RetryOnBusy can be found in the Preferences of CMAS or can be read or altered by **preferencevalue()**.

Function	RetryOnBusy	Handle	Description
_book()	true	*	will never fail (see note)
_book()	false	yes	will return done or failed
_book()	false	no	will fail and stop the execution on an error

Note

a call to _book() can fail if the negotiation process fails. To overcome this it is possible to first call negotiate() and check if it is done before calling _book().

Example:

```
Program Processplan;

define handle RemoteInterface : interface;
begin
    var A : integer;
end;

procedure BookAndAgree();
var negotiated:=false, booked:=false : boolean;
var counter:integer;
begin
    writeln('RetryOnBusy=', preferencevalue('RetryOnBusy', false));

    repeat
        // Will try to negotiate forever with all compatible interfaces
        repeat
            if RemoteInterface.negotiate()=processplan.done then
                negotiated:=true;
            else
                sleep(500); // Wait some time before we try again
        until not negotiated;

        // Will try to book five times after that give up
        for counter:=1 to 5 do
        begin
            if RemoteInterface._book()=processplan.done then begin
                booked:=true;
                break;
            end;
            else
                sleep(500);
        end;
        until not booked;
    end.

    function onRun() : integer;
begin
    BookAndAgree();

    RemoteInterface.a:=2;
    RemoteInterface.s2(); // Call a skill on the remote interfcae
    RemoteInterface._unbook();

```

```

    onRun:=processplan.done;
end.
```

Variable bounds

Variables can be used to handle values inside an agent. With variable bounds it is possible to limit the values of a variable of type integer, real and string.

Any variable in the modelling tree can have a range expressed as upper and lower bounds. This behaviour is useful when the variable is part of an interface since the range is part of the negotiation process. If two agents negotiate a variable value must match the remote variable range.

Integer and Real

If upper bound is set to 5 and lower bound to 1 the value can only be within the range 1-5, i.e. $1 \leq x \leq 5$, and the variable will only accept a value in that range during negotiation.

Note

At creation of a variable, the lower and upper bounds are empty and there is no limitation on the variable.

It is possible to write more complex expressions for upper and lower bound.

The basic syntax for a range expression:

Example	Name	Description
1	fixed value	only 1 is accepted
<100	open range lt	all values lower than 100 are accepted
>50	open range gt	all values greater than 50 are accepted
!2	not	all values except 2 is accepted
[>100 <200]	AND	the set of all the elements inside [] must be true
1 2 3 >10	OR	one of the element in the set separate by must be true

Example 1:

```
[<100 >50 !70] Will give a range less than 100 AND greater than 50 AND NOT the value 70)
```

Example 2:

```
[<10 >5] | [<20 >15] A dual range example, value must be between 5 AND 10 OR 15 AND 20
```

String

A string can only handle the or | operator when it comes to range expressions.

Example:

```
start | stop | pause Will only accept strings with a value of 'start', 'stop' or 'pause'
```

Boolean

Boolean does not implement complex expressions, however it is possible to limit the value by setting upper or lower bound to true or false.

Goal

In CMAS, a Goal serves a dual purpose:

- Internal Purpose: It represents a objective that an agent seeks to fulfill through collaboration with other agents.
- Interactive Purpose: It also acts as an interface for interacting with other agents. This means that a goal must have an interface name and process plan name to determine how it can be achieved.

Each goal includes built-in behavior that defines a default strategy for how the goal seeks to be fulfilled. This strategy is based on an abstract interface, which enables the agent to initiate negotiations with other agents to find the most suitable remote interface for achieving the goal. During the negotiation process, the agent uses the goal interface name, the desired process plan name, and the goal variables to match the abstract interface with a compatible implementation on another agent. Once a suitable match is found and negotiated, the requested process plan is executed remotely via the mapped interface.

This behavior is executed locally by the agent and below is the logic for the built-in behavior:

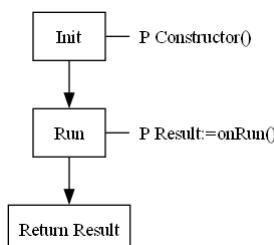
```
Program ProcessplanGoal;
abstract resource : interface;

function onRun() : integer;
begin
    resource.agreeandbook();
    tonRun:=resource.getProcessPlanName();
    resource._unbook();
end.
```

Note

The onRun function can be overridden to replace the built-in default mechanism. It must return the integer value processplan.DONE (2) when the goal is successfully achieved. However, in most cases, the default behavior is sufficient and recommended.

Function	Description
onRun() : integer	Called to execute the goal

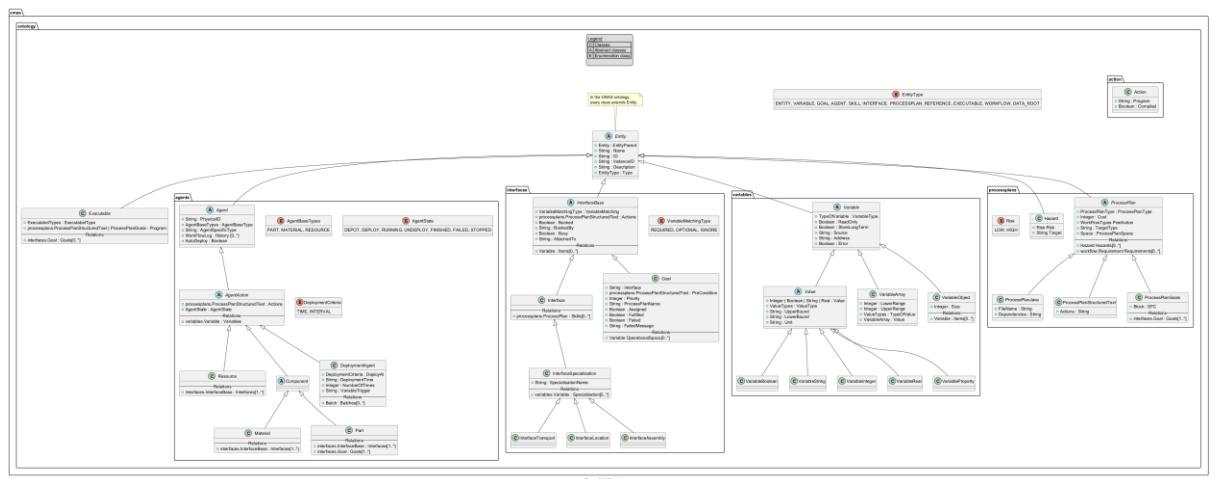


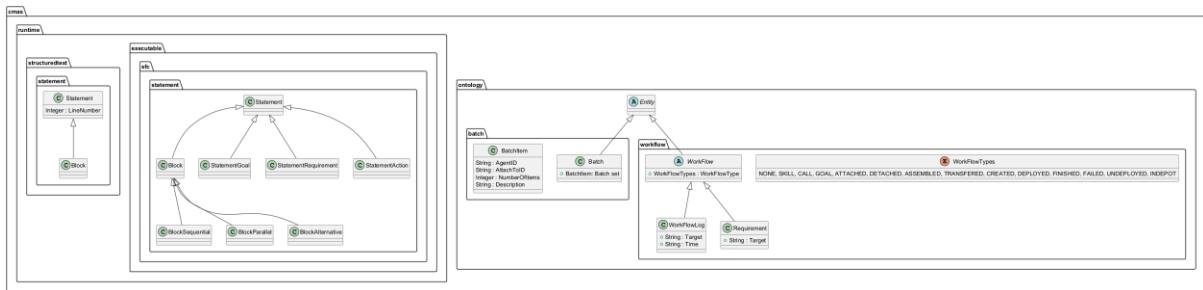
Ontology

Introduction

Object-oriented data modelling is a technique used in CMAS to represent and organize data as a collection of objects. Overall, object-oriented data modelling is a powerful approach for designing systems that closely align with real-world entities and relationships. Every object within CMAS comprises a combination of properties and relationships, alongside inherited properties and relationships. As an example, if you have an Agent object in the CMAS system, it will have all the properties and relationships that are defined for the more general Entity object. This is a common concept in object-oriented data modelling, where specialised objects (like Agent) inherit attributes from more general classes (like "Entity"). A property represents a value that can be used by the object it self or the user, while a relation denotes a link to another object. Relations can handle new relations dynamically if they are compatible with the specific type of connection. A relation can be a direct relation to a child object or a reference relation to any object in the system. All objects in a model are visible in the Modelling tree viewer in CMAS and all properties are visible in the properties viewer. Relations categories are visible as sub-folders in the modelling tree holding child objects. The available objects in CMAS is based on a classes defined in the CMAS ontology. A class serves as a blueprint or template for creating objects and it defines the structure that objects of that class will have.

For a brief overview of the CMAS ontology see the figure below (or see the inheritance diagram for classes **Entity**).





cmas.ontology Objects

All objects in the CMAS ontology and their specific properties are described in the following section.

Entity

All objects in CMAS ontology are based on the general object **Entity**.

Table Entity class attributes

Attribute	Description	Type
Entity::EntityParent	A parent entity object. If parent is set it is possible to inherit property values from that parent. When parent is set to a valid entity a check box will be visible for all properties that can be inherited. If the check box is checked the local value will be override by the parent value.	Entity
Entity::Name	Name is a user friendly name of an entity. It is possible to refer to an entity by name, however, a name must not be unique two entities can share the same name.	String
Entity::ID	An ID is a unique identifier of an entity during modelling time. The ID is automatically generated by CMAS when a new entity is created. However, it is possible to set the ID to a more readable form to be able to refer an ID but note that it must be unique.	String
Entity::InstanceID	An InstanceID is an	ReadOnly String

	identifier of an entity during runtime and must be unique in the system. The InstanceID is automatically generated by CMAS during runtime.	
Entity::Description	A user defined description of the entity. The description is not used by the system but it is visualised in many dialogs in CMAS and can help to better document specific entities.	String
Entity::Type	The base type of the object.	ReadOnly EntityType

cmas.ontology.processplans

ProcessPlan

All process plans in CMAS are based on the ProcessPlan object.

Table ProcessPlan class attributes

Attribute	Description	Type
ProcessPlan::ProcessPlantype	Process plan type.	processplans.ProcessPlanType
ProcessPlan::Cost	The cost to execute the plan.	Integer
ProcessPlan::PostAction	The outcome of executing the process plan, this is used by the built in planner.	workflow.WorkFlowTypes
ProcessPlan::TargetType	If there are any risks what is the target type.	String
ProcessPlan::Space	Required space.	ProcessPlanSpace

ProcessPlanGoals

A ProcessPlanGoals represents a process plan that primarily consists of goals. These goals can be arranged in a goal chart or an SFC-like (Sequential Function Chart) diagram. Each goal has precondition logic that must be true before the goal can start. It is also possible to add actions and other SFC elements, such as jumps, conditions, and so on—to the chart.

Table ProcessPlanGoals class attributes

Attribute	Description	Type
ProcessPlanGoals::SFC	Process plan type.	runtime.executable.sfc.state ment.Block

ProcessPlanStructuredText

A ProcessPlanStructuredText defines a process plan written in Structured Text, a high-level programming language. Compared to ProcessPlanGoals, Structured Text offers more advanced and flexible programming features, enabling the implementation of more complex process logic.

Table ProcessPlanStructuredText class attributes

Attribute	Description	Type
ProcessStructuredText::actions	A structured text program.	String

Hazard

A hazard object represents a potential hazard associated with a process plan. Each hazard has a target, which is a user-defined name.

Table Hazard class attributes

Attribute	Description	Type
Hazard::Risk	A risk.	Risk
Hazard::Target	A user defined target for the risk.	String

cmas.ontology.agent

All agents in CMAS ontology are based on the object Agent.

Table Agent class attributes

Attribute	Description	Type
Agent::PhysicalID	A user defined physical ID of the hardware that the agent represents. The PhysicalID is not used by the system but can be used for identifications or in programs.	String
Agent::AgentBaseType	The agent base type.	ReadOnly AgentBaseTypes
Agent::AgentSpecificType	The agent specific type is user defined string that can be used to determine what type of agent its. This property is not used by the system but can be utilised in a program to make	String

	decisions.	
Agent::AutoDeploy	If this is false the agent will not be deployed at run or when deployed all is pressed.	Boolean

AgentAction

AgentAction implements an overall controller and states for all agents according to **actions**. Agent action has a relation named Variables that accept all ValueTypes.

Table AgentAction class attributes

Attribute	Description	Type
AgentAction::Actions	A structure text program that implements actions .	processplans.ProcessPlanStructuredText
AgentAction::AgentState	Agent current state during runtime, i.e., INIT, DEPOT, DEPLOY, RUNNING, UNDEPLOY, FINISHED, FAILED, STOPPED.	AgentState

Resource

A Resource agent is a passive agent that can be used as a resource. A Resource agent has a relation named Interfaces that accept all InterfaceTypes.

Table Resource class attributes

Attribute	Description	Type

Material

A Material agent is a passive agent that can be used as a subpart when it comes to assembly. A Material agent has a relation named Interfaces that accept all InterfaceTypes.

Table Material class attributes

Attribute	Description	Type

Part

A Part agent is an active agent that will try to fulfil all goals. A Part agent has a relation named Goals that accept all GoalTypes.

Table Part class attributes

Attribute	Description	Type
Part::ProcessPlan	A goal chart that express the order of the goals.	ProcessPlanGoals

DeploymentAgent

The deployment agent will deploy all batches based on *DeploymentTime* or *VariableTrigger*. If *VariableTrigger* is empty, it will be ignored, and if *DeploymentTime* is less than 0, it will also be ignored. Otherwise, deployment occurs when the time is reached or the trigger variable is changed to true. For more complex trigger logic, you can utilise the Agent action and the Running procedure to modify the trigger variable's value.

Attribute	Description	Type
DeploymentAgent::DeployAt	If set to TIME, the first batch will be deployed after DeploymentTime, with each subsequent batch deployed at intervals of DeploymentTime. If set to INTERVAL the first batch will be deployed at time 0 with each subsequent batch deployed at intervals of DeploymentTime.	DeploymentCriteria
DeploymentAgent::DeploymentTime	The time used by DeployAt. If no suffix is provided, the time is interpreted in milliseconds (e.g., 500 means 500 milliseconds). Valid suffixes include: h (hour), m (minutes), s (seconds), and ms (milliseconds). For example, 20s represents 20 seconds. If set to a negative value the time will be ignored.	String
DeploymentAgent::NumberOfTimes	Number of times to repeat the deployment, (0 = infinite number of times).	Integer
DeploymentAgent::VariableTrigger	The name of a local agent variable that, when it becomes true, will trigger a deployment. If it is left empty the TriggerVariable will be ignored.	String

cmas.ontology.variables

Variable

Table Variable class attributes

Attribute	Description	Type
Variable::VariableType	Type of variable.	ReadOnly TypeOfVariable
Variable::ReadOnly	If the variable is read only during run time.	Boolean
Variable::StoreLongTerm	If the variable value is stored in an external database or not.	Boolean
Variable::Source	Adapter source field, see adapters .	String
Variable::Address	Adapter address field, see adapters .	String
Variable::Error	If an error has occurred in the adapter during runtime.	ReadOnly Boolean

Value

Table Value class attributes

Attribute	Description	Type
Value::Value	The variable value.	Integer Boolean String Real
Value::ValueType	The variable value type.	ValueTypes
Value::UpperBound	The value UpperBound, see variable bounds .	String
Value::LowerBound	The value LowerBound, see variable bounds .	String
Value::Unit	The value unit of measurement.	String

VariableBoolean

A boolean variable.

Table VariableBoolean class attributes

Attribute	Description	Type

VariableInteger

An integer variable.

Table VariableInteger attributes		
Attribute	Description	Type

VariableReal

A real variable.

Table VariableReal attributes		
Attribute	Description	Type

VariableString

A string variable.

Table VariableString attributes		
Attribute	Description	Type

VariableArray

An array variable.

Table VariableArray attributes		
Attribute	Description	Type
VariableArray::LowerRange	The lower range of the array.	Integer
VariableArray::UpperRange	The upper range of the array.	Integer

VariableObject

An variable object is a user defined type that has a relation named Items that accept all ValueTypes.

Table VariableObject attributes		
Attribute	Description	Type
VariableObject::Size	Number of items in the relation Items.	ReadOnly Integer

cmas.ontology.interfaces

InterfaceBase

Table InterfaceBase attributes

Attribute	Description	Type
InterfaceBase::VariableMatching	Variable matching for the interface, see interfaces .	VariableMatchingType
InterfaceBase::Actions	Interface actions, see interface action .	processplans.ProcessPlanStructuredText
InterfaceBase::Booked	Runtime property.	ReadOnly Boolean
InterfaceBase::BookedBy	Runtime property.	ReadOnly String
InterfaceBase::Busy	Runtime property.	ReadOnly Boolean
InterfaceBase::AttachedTo	Runtime property.	ReadOnly String

Interface

An interface object that has a relation named Skills that accept all SkillTypes.

Table Interface attributes

Attribute	Description	Type

Goal

A goal object.

Table Goal attributes

Attribute	Description	Type
Goal::Interface	A interface name used for a goal during negotiation.	String
Goal::PreCondition	Precondition to enable the goal during runtime.	processplans.ProcessPlanStructuredText
Goal::Priority	Goal priority.	Integer
Goal::ProcessPlanName	The name of a process plan (or skill) that can assist to reach the goal.	String
Goal::Assigned	Runtime variable to indicate if the goal is assigned to a process plan instance tha can solve the goal.	Boolean
Goal::Fulfilled	Runtime variable true if	Boolean

	goal is fulfilled.	
Goal::Failed	Runtime variable true if goal failed.	Boolean
Goal::FailedMessage	Runtime variable a string explaining the reason for a failure.	ReadOnly String

InterfaceSpecialisation

An interface object that can be used to implement specialisation tasks.

Table Interface attributes

Attribute	Description	Type
InterfaceSpecialisation::SpecialisationName	Name of the specialisation.	String

InterfaceSpecialisation templates

The listed InterfaceSpecialisation objects below are not new object types but rather templates based on the InterfaceSpecialisation class. These templates are designed to simplify the modeling of specific specialisations by predefining certain relations and property values.

- InterfaceLocation
- InterfaceTransport
- InterfaceAssembly

cmas.ontology.workflow

WorkFlow

An abstract WorkFlow object.

Table WorkFlow attributes

Attribute	Description	Type
WorkFlow::SpecificType	The specific workflow type.	ReadOnly SpecificTypes
WorkFlow::WorkFlowType	The workflow type.	WorkFlowTypes

WorkFlowLog

A WorkFlowLog object to log the work flow on agents.

Table WorkFlowLog attributes

Attribute	Description	Type
WorkFlowLog::Target	Target for the log.	String
WorkFlowLog::Time	Time stamp for the log.	String

Requirement

A Requirement object to add requirements on skills.

Table Requirement attributes

Attribute	Description	Type
Requirement::Target	Target for the requirement.	String

cmas.ontology.batch

Batch

A batch represents a set of entities that will be deployed at the same time.

Table Batch attributes

Attribute	Description	Type
Batch::Batch set	The set of entities that will be deployed	BatchItem

BatchItem

Table BatchItem attributes

Attribute	Description	Type
BatchItem::AgentID	An ID of an agent that will be deployed	String
BatchItem::AttachToID	An ID of an interface to attach to	String
BatchItem::NumberOfItems	Number of items to deploy	Integer
BatchItem::Description	Description	String

Extended Ontology

It is possible to extend the existing ontology in CMAS by creating new objects based on the existing ones. During the startup of CMAS, if the file **cmas-ontology.json** exists, it is read and interpreted.

Note

Normally, an extension should be achieved by simply adding new variables or modifying existing ones, rather than defining a new ontology with new object types.

This JSON file can contain instructions on how to extend the current ontology. The file consists of an array of object definitions, where each object must include the following information:

- **Name** : The name of the new object to define.
- **Description** : A description of the object.
- **Category** : The object's category.
- **Extends** : The base class the object extends from. Must be an existing CMAS object.

You can also define new properties that are added to the object, with the supported types being Integer, String, Boolean, and Real. A property must include:

- **Name** : The property's name.
- **Type** : The property's type (Integer, Real, Boolean or String).
- **Description** : A description of what the property represents.
- **LocalValue** : A default value for the property.
- **ClassName** : The fully qualified class name of the object the property belongs to. This is typically the base class domain plus the object name.
- **CanInherit** : A boolean indicating whether the property value can be inherited from a parent
- **Override** : A boolean specifying whether a inherited property value is overridden by a local value, must be false
- **ReadOnly** : A boolean indicating whether the property is read-only
- **ParentID** : An identifier of the parent object, must an empty string
- **.PropertyType** : How the property can be utilized in the property viewer.

Additionally, you can define new relations for the object. A relation object must include:

- **Name** : The name of the relation.
- **AcceptedObjects** : An array of object names in CMAS that the relation can host.

Example of two new objects Temperature and Operator in the **cmas-ontology.json** file:

```
[  
  {  
    "Name": "Temperature",      /* Object Name */  
    "Extends": "VariableReal", /* Base class in CMAS */  
    "Description": "",         /* Description of the Object */  
    "Category": "Value",       /* Object category */  
  
    "Properties": [],          /* No new properties */  
    "Relations": []            /* No new relations */  
  },  
  
  {  
    "Name": "Operator",        /* Object Name */  
    "Extends": "Resource",     /* Base class in CMAS */  
    "Description": "",         /* Description of the Object */  
    "Category": "Agent",       /* Object category */  
  
    "Properties": [           /* An array of new properties */  
      {  
        "Name": "Time",          /* Property name */  
        "Type": "Integer",        /* Property type */  
        "Description": "The max time to work",  
        "LocalValue": 10,          /* Default value */  
  
        "ClassName": "cmas.ontology.agents.Agent",  
        "CanInherit": true,  
        "ReadOnly": false,  
        "ParentID": "",  
        "Override": true,  
      }  
    ]  
  }  
]
```

```

        "PropertyType": "[]"
    },
],
"Relations" : [ /* An array of new relations */
    {
        "Name": "Demands", /* Name of the relation */
        "AcceptedObjects" : ["Temperature"] /* Name of the accepted objects that
the relation can host */
    }
]
]

```

The category can be one of the following values:

- "value"
- "skills"
- "interface"
- "goal"
- "executable"
- "hazard"
- "processplan"
- "agent"
- "batch"
- "deployment"
- "workflow"
- "requirement"

The PropertyType can be a combination of:

- INHERITANCE Possible to set inheritance
- REFERENCEFILE Only show if a reference file
- PARENTEDITOR Possible to edit parent
- IDEEDIT Opens an ID editor
- TEXTEDITOR Open a text editor for the value
- STRINGEDIT Open a simple string editor for the value
- VARIABLEEDITOR Open an array editor
- SFCEditor Open graphical SFC editor
- PROCESSPLANEDITOR Possible to select a process plan
- BATCHEDITOR Open an editor for batches
- INTERFACEBROWSER Possible to browse for interface name
- ADDRESSBROWSE Possible to browse for an adapter address
- HISTORYSHOW Open a window to show the history log
- HIDE Don't show at all
- RUNTIME Only show during runtime
- MODELLING Only show during modelling
- HIDE BASIC Only show during normal

Planner

Introduction

Assume we need to run process A for a product. First we need a method to locate the nearest resource capable of executing it. This can be achieved through negotiation with abstract

interfaces to identify a suitable resource. However, once a resource is selected, we must determine a path through the system and identify a suitable transportation resource to reach it before we can run process A. This can be an extensive task in the end.

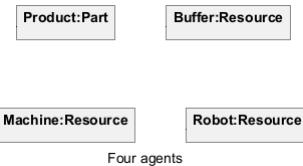
The Planner Agent in CMAS can automatically generate solutions to achieve a goal or meet a requirement. This will help the modeler to avoid writing complex transportation programs. The key is to model the system in a way that the Planner Agent can understand.

The planner will work according to the following algorithm:

```
PathFinder(Domain, targetInterface, destinationInterface)
    map := GenerateMap(Domain);
    solver := Solver( SMT | PDDL)
    solver.generateProblem(Map, targetInterface, destinationInterface)
    Plan := solver.findBestPlan()
    execute(Plan)
```

Example 1 - one robot

In this example we have four Agents, one Part called Product and three resources Buffer, Machine and Robot.



Let's create a model that can make use of the Planning Agent.

Machine

We start by building a **Resource** agent to represent the Machine. This resource will serve as a provider for process A . To model process A , we need to establish an interface. This interface will function as a destination for the part, meaning the part must be connected to the interface's location in order for process A to run.

To achieve this, we create a **LocationInterface** , which is essentially a standard interface with a specialization to act as a location as well. We can name this interface InterfaceProcessA. If we expand the interface, we will notice it already includes two skills that were automatically generated when the location interface was created: Detach and Attach. These skills can be filled with logic to define how to attach and detach from the interface. For instance, attaching might require clamping. However, for now, we can leave these skills empty as long as they return the value *processplan.done* . Note, that these two skills also have the **PostAction** property set to *ATTACHED* and *DETACHED* .

Additionally, we will find a Specialization folder within the interface that contains a **VariableObject** named *Location* . This variable represents the geometrical location of the interface, but it does not contain any specific variables. It's up to the designer to determine how a location will be represented in the system. You could use real variables like *x*, *y*, *z*, *rx*, *ry*, *rz* for a full robot representation, or simply use an integer *position* to represent discrete locations, such as location 1, 2, 3, etc. Whatever approach we choose, it's crucial that all agents in the system follow the same notation for location. In this example, a discrete number will suffice. We can achieve this by expanding the Position object and add a new **VariableInteger** named *Position* and give it a value of 2. This value will be a known

position for the system to represent process interface A. We also make it a read only value by setting **ReadOnly** to true. Now we can create process *A* by adding a new **processplannstructuredtext** to **Skills** on the interface and the skill can look like this:

```
Program ProcessplanA;
function onRun() : integer;
begin
    writeln('Running process A');
    onRun:=processplan.done;
end.
```

To run process A, we have a requirement that the part requesting the process must first be attached to it. We can model this requirement by selecting the **Requirements** folder for skill *A* and adding a new **Requirement**. Name this requirement *RequirementAttached* and set the **WorkFlowType** property to *ATTACHED*. This ensures that anyone wanting to execute skill *A* must first be attached to the interface.

Buffer

Just like we created the *Machine*, we can now create the *Buffer* as a **Resource**. For the *Buffer*, we also need a **LocationInterface**, but this time, we can set the *position* to 1 instead. In this example, no additional specific skills are required on the *Buffer*, except for the one that was automatically created for the specialization. We can name the interface **InterfacePart**. To make it easier to distinguish the buffer skill, we can rename the skill **Attach** to **AttachOnBuffer**. We will need this later.

Robot

Now, let's create a **Resource** agent and name it *Robot*. On the *Robot*, create a new specialization interface called **InterfacePart**. This interface will be specialized for transportation tasks.

We can name the interface *InterfacePart*.

On a transportation interface, we will find three predefined variables: **FromLocation**, **ToLocation**, and **Step**. For both the **FromLocation** and **ToLocation**, we need to add an integer variable called **Position**. However, we won't set a specific value for this variable, nor will we make it read only.

Now open the skill predefined skill **Transport**.

```
program ProcessplanTransport;
function onNegotiate(Planning : boolean): integer;
begin
    // Only accept moving forward
    if remoteinterface.specialisation.fromlocation.position >
        remoteinterface.specialisation.tolocation.position then
        onNegotiate:=-1; // reject
    else
        onNegotiate:=this.cost; // use cost value
end.

function onRun() : integer;
begin
    if interface.specialisation.step=1 then
        writeln('Robot will pick up from ',
            interface.specialisation.fromlocation,
            ' and transport to ',
            interface.specialisation.tolocation);
    else
        writeln('Robot will place at ',
            interface.specialisation.tolocation);
```

```

    onRun:=processplan.done;
end.
```

Part

Now, create a **Part** agent and name it *Part*. On the **Part**, create a new **Interface** and name it **InterfacePart**. This will make the **Part** compatible with the **buffer**, **robot**, and **machine**. When the **Part** is deployed, it needs to be attached to the **buffer** in the system. To accomplish this, we need to define the logic for the deployment process. Select the **Part** and open the **Actions** editor. Here, we can use the dedicated skill **AttachOnBuffer** to ensure the **Part** attaches to the buffer, rather than to the machine or robot.

```

program Part;

abstract aResource : interface;

procedure onDeploy;
begin
    aResource.AttachOnBuffer();
end.
```

Now, open the process plan editor for the **Part** and create a new **Goal**. Edit the **Goal** by setting the process plan name to *A* and the interface to *InterfacePart*. Next, drag the goal to the goal chart and drop it on the start. The **Part** is now ready.

Example 2 - adding an operator

Now, we would like to add an operator to the system to handle situations when the robot is out of order, but only in those cases. We begin by adding a new **Resource** and naming it *Operator*. For the operator, we duplicate the interface of the *Robot*, which means creating an **InterfaceLocation** and naming it **InterfacePart**. Next, we update the two variables, *FromLocation* and *ToLocation*, by adding an integer variable named *position*, in the same way we did earlier for the *Robot*. Now, we need to implement the *Transport* skill. We start by renaming the default *Transport* skill to *ManualTransport*. This helps distinguish it as a manual task. Then, we set the cost for the *ManualTransport* skill, for example, 5. As long as the cost is higher than the robot's, any value will work. After that, we can implement the logic for the *ManualTransport* skill :

```

program ProcessplanManualTransport;

function onNegotiate(Planning : boolean): integer;
begin
    // Only accept moving forward
    if remoteinterface.specialisation.fromlocation.position >
        remoteinterface.specialisation.tolocation.position then
        onNegotiate:=-1; // reject
    else
        onNegotiate:=this.cost; // use cost value
end.

function onRun() : integer;
begin
    // Write a message for the operator on the HMI
    // This can be done with, e.g., OPC UA connected variables
    if interface.specialisation.step=1 then
        writeln('Operator must pick up from ',
            interface.specialisation.fromlocation,
            ' and transport to ',
            interface.specialisation.tolocation);
```

```

        else
            writeln('Operator must place at ',
                   interface.specialisation.tolocation);

        onRun:=processplan.done;
end.
```

Updated robot

Now, we need to update the *Robot* agent to handle errors. We begin by adding an agent variable to the *Robot*. This can be a boolean variable named *RobotError*. At this point, it's possible to map it to a physical signal using adapters, but for this example, we will leave it as is. The *Transport* skill must now check the *RobotError* signal and refuse to participate by rejecting any negotiation attempts when *RobotError* is true. This can be easily done by adding an **OR** *RobotError* condition to the rejection part of the *onNegotiate* function.

```

program ProcessplanTransport;

function onNegotiate(Planning : boolean): integer;
begin
    // Only accept moving forward
    if remoteinterface.specialisation.fromlocation.position >
        remoteinterface.specialisation.tolocation.position OR
        agent.RobotError
    then
        onNegotiate:=-1; // reject
    else
        onNegotiate:=this.cost; // use cost value
end.

function onRun() : integer;
begin
    if interface.specialisation.step=1 then
        writeln('Robot will pick up from ',
               interface.specialisation.fromlocation,
               ' and transport to ',
               interface.specialisation.tolocation);
    else
        writeln('Robot will place at ',
               interface.specialisation.tolocation);

    onRun:=processplan.done;
end.
```

Running Example 2

When running the example and deploying a part, we can see in the console that the planner selects the **Robot** for transportation. This is because the robot's cost is lower than the operator's.

```

Robot will pick up from {"Position":"1"} and transport to {"Position":"2"}
Robot will place at {"Position":"2"}
Running process A on Machine
```

If we deploy several parts, the system will always select the robot. However, if we open the running tree, access the robot agent, and change the *RobotError* signal to true during runtime, we will see that the next part is planned with the operator instead. For more complex scenarios, note that the planner will always attempt to optimize the total cost of the plan. We can observe the same effect if we undeploy the robot because we no longer need it. To do this, simply select the robot in the running tree and click "Undeploy." When we want to use the robot again, we can deploy it again from the depot.

Processplans

In CMAS there exist something called process plans. Process plans can be created for different purposes. The flexibility of CMAS comes when you're putting together all process plans in one system. A process plan can be expressed as a goal chart (GC) or as **structured text (ST)**. A process plan is executed according to the actions described in the section **process plan actions** and can be defined for the following entities:

Entity	Description	Modelling language
Part	a description on how a part can reach all its goals to become finalised	GC
Resource	a description of a skill on an interface available for other agents to utilize	ST or GC
Global Process Plan	defined in the root of the modelling tree and can be used by parts or resources	ST or GC

Part Agent

Before commencing production, process planning is essential, outlining the necessary steps to manufacture or assemble the end product. This crucial stage establishes the prerequisites and the sequence of operations. This must be expressed as a sequence in goal chart for a part agent. A goal serves as a bridge between product design and manufacturing and describes something that we need to reach. An example of a goal might be to get hole in the centre of the part. The process requirements are reflected in the order or sequence of the goals. For example, it might be more useful to first create a hole and then deburr it, rather than doing it the other way around. In CMAS all part agents must have at least one goal and a process plan that utilise at least one goal. To reach the part process plan select the part in the modelling tree and in the property window select processplan Edit button you can also double click on the part in the tree to open the process plan.

Resource Agent

A resource will publish available skills through **interfaces**. An interface describes requirements as variables and skills as process plans. A skill is a process plan describing the required steps needed to perform the skill. A skill can be used to fulfil a goal or as a required step in another skill. A well-crafted skill allows for adaptability and flexibility. To edit a skill double click on it in the modelling tree,

Global process plans

A global process plan is defined in the root of the modelling tree and can be used by parts when they are searching for a way to fulfil a goal. When used to fulfil a goal the process plan will be executed by the part agent it self.

It is also possible to use global process plans to describe skills on resources. To do this drag the global process plan to a resource interface and drop it and select reference.

For more information see:

- [Structured text](#)
- [Process plan actions](#)
- [Interfaces](#)

The hierarchical structure: Goal to Execution

This section outlines the structure and hierarchy of elements involved in task execution and planning. Each element plays a distinct role in organizing and coordinating activities to achieve system goals efficiently. The framework is divided into five primary categories: Goals, Operations, Process Plans, Skills, and Actions, each serving specific functions and working together to enable seamless execution of complex tasks.

1. Goal or requirement (System objective level)
 - Role: Defines what the system must achieve.
 - Scope: Broad, user-defined outcome, originates from the design.
 - Properties: Triggers planning or execution processes. It may be decomposed into one or more operations / skills. Goals are typically defined for parts and requirements for resource skills.
2. Operations (Abstract task)
 - Role: Represent abstract tasks such as "transport item to station".
 - Scope: Broad in scope, they are domain-relevant and understandable at the planning level.
 - Properties: Serve as units (edges) of planning in high-level graphs and can not be created manually.
3. Process Plans (General Executable capabilities)
 - Role: Implement general operations by organizing the logic and flow of execution.
 - Scope: Can include actions and delegation to remote skills (recursive structure).
 - Properties: Serve as reusable, parameterizable execution strategies that is not associated with any specific agent. The agent that executes the Process Plan will be consider the target agent. Define coordination logic (e.g., sequences, selections, iterations, synchronizations).
4. Skills (Agent specific Executable capabilities)
 - Role: Implement agent associated operations by organizing the logic and flow of execution.
 - Scope: Can include local actions and delegation to remote skills (recursive structure).
 - Properties: Serve as reusable, parameterizable execution strategies. Define coordination logic (e.g., sequences, selections, iterations, synchronizations).
5. Actions (Low-Level, Atomic Behaviors)
 - Role: Directly executable local functions or commands (e.g., "move arm", "start conveyor motor").
 - Scope: Minimal scope; cannot be broken down further.
 - Properties: Require no external coordination or resources beyond the executing agent.
 - Must be deterministic, safe, and bounded in execution.

Structured text

Introduction

Structured text (or Pascal) can be used to describe skills and process plans. We will start with the famous Hello World in structured text.

```

Program HelloWorld;

{ Program to write Hello World on the console }
Begin
  writeln('Hello World');
End.

```

The overall structure in a program

A program consist of the following sections:

- **Program name** (optional)
- **Directives** (optional)
- **Struct type** (optional)
- **Interfaces** (optional)
- **Global variables** declarations (optional)
- **Comments** (optional)
- **Functions declarations** (optional)
- **Procedures declarations** (optional)
- **Main program block** or constructor (optional)

Program

Section: Program name

Syntax: Program <Name>;

Description: <Name> is a descriptive text string that defines the name of the program.

Example:

```

Program HelloWorld;

{ Program to write Hello World on the console }
Begin
  writeln('Hello World');
End.

```

Directives

The following directive exists

Directive	Value	Description
\$I	<files_string>	Include file
\$INCLUDE	<file_string>	Include file
\$U	<ID>	Include resource by ID
\$USE	<ID>	Include resource by ID
\$ERROR	<message_string>	Stop compiler and give an error

Include

Section: directives

Syntax: \$include <filename> | \$i <filename>

Description: include will read another file and include it in the compilation. If

<filename> starts with *system*: a predefined system file can be included, see more about system files in **system include files**.

Example:

```
Program IncludeFile;

{ Directive to include the file standard.inc }
$INCLUDE 'standard.inc'

{ Will include a system file }
$INCLUDE 'system:file.inc'

Begin
  // Will print the object MODE defined in system:file.inc
  writeln(MODE);
End.
```

Use

Section: directives

Syntax: \$use <id> | \$u <id>

Description: use will include another entity based on the id. The id is defined in the modeling tree.

Example:

```
Program IncludeResource;

{ Directive to include a program defined in another entity }
$USE 'aResource'

Begin
  // Will call a function defined in another entity
  start();
End.
```

Error

Section: directives

Syntax: \$error <message>;

Description: will stop the compiler and show the error message.

Example:

```
Program DoNotCompile;

Begin
  // Will prevent any one to compile and use this resource
  error 'Do not use this resource yet it is not ready!';
End.
```

Note

\$error will generate an error when compiling for run time errors use **error** instead

Struct type

Section: Struct type

Syntax: struct <name> [extends <base>]; Begin <variable declarations> End;

Description: Definition of new structure types. It is possible to extend a previously defined structure by the use of extends.

Example:

```
Program StructExample;

{ Structure definition }
Struct Pos2D;
Begin
    Var x, y : Real;
End;

{ Structure definition }
Struct Pos3D extends Pos2D;
Begin
    Var z : Real;
End;

{ Variable declaration using a structure definition }
Var a: Pos2D;

// Variable declaration with an initialisation
Var b:=(x:=1.0, y:=0.0): Pos2D;
Var c:=(x:=1.0, y:=0.0, z:=1.0): Pos3D;

Begin
    a.x:=0.0;
    a.y:=1.0;
    writeln('Struct a = ', a);
    writeln('Struct b = ', b);

    writeln('Struct c = ', c);
End.
```

Interface

It is possible to declare interfaces in a program. This must be done if it is impossible to model the interface or if the interface is abstract. These interfaces can be used to interact with other agents, e.g., to request a skill to start and send variable values. The following interface types exist:

- **Abstract**
- **Define**
- **Static**

For more details see the section **interfaces and negotiation**.

Abstract interface declarations

Section: Abstract interface declarations

Syntax: abstract [handle] <name1> [, <name2>] [, ...] : interface;

Description: Declaration of abstract interfaces. Abstract interfaces are mapped during runtime through a negotiation process. The negotiation process consider variables and used skills. If there are many interfaces that accept the negotiation the one with the lowest cost is selected. The negotiation process starts automatically when an item of an abstract interface is accessed for the first time. It is possible to force a negotiation for an interface by calling the negotiation() action on the abstract interface.

Example 1 :

```
Program AbstractExample;
{ Abstract remote interfaces }
abstract Robot, Resource : interface;

Begin
    Robot.MoveTo();

    Resource.Use();
    Resource.Signal:=TRUE;
    Resource.Motor();
End.
```

Example 2 :

```
Program AbstractHandleExample;
{ Abstract remote interfaces that is handled by onRun()  }
abstract handle transport : interface;
var returnValue:integer;

function onRun() : integer;
begin
    transport.negotiateandbook();
    transport.moveRight(returnValue);

    // Wait until moveRight skill call is done
    // this is necessary since the call is done with an result variable
    repeat
        sleep(100);
    until returnValue=processplan.running;

    transport._unbook();
    onRun:=returnValue;
end.
```

Note

The negotiate() call will make sure that the abstract interface is renegotiated on each run. If negotiate() is not called the system will make sure that a negotiation is made the first time and then the interface will remain static and no further negotiations are made. The repeat loop will wait for the skill moveRight() to end.

Define interface declaration

Section: Define a new local interface syntax 1

Syntax1: define [handle] <name> : interface; Begin <variable declarations> <function definitions>End;

Description: Declaration of a new local interface for use to find compatible remote interfaces. This can be used when it is not possible define it in the configuration. The declared name of the interface is the name used for matching with remote interfaces. All declared variables together with all skill calls used on the abstract interface is automatically added to the requirement list for matching. It is also possible to add skills manually by defining function names in the interface.

Example:

```
Program DefineExample1;

{ Define a new local interface }
define aRobot : interface;
begin
    var x, y, z : real;
    function move;
end;
```

```

function onRun() : integer;
begin
    aRobot.x:=2;
    aRobot.y:=3;
    aRobot.z:=0;

    // Will search for a remote interface aRobot and run the skill move() on the remote
    // interface.
    aRobot.move();

    // Return value - running, done, failed or interrupted
    onRun:=processplan.done;
end.

```

Section: Define an existing local interface syntax 2

Syntax2: define [handle] <name> :=<interface name>' interface;

Description: Declaration of an existing local interface, named <interface name> for use to find compatible remote interfaces. This can be used when you do not want to search locally for an interface. Note, <name> will be used as interface name during the negotiation process.

Example:

```

Program DefineExample2;

{ Define an existing local interface robot to use as an abstract interface }
define aRobot:='robot' : interface;

function onRun() : integer;
begin
    aRobot.x:=2;
    aRobot.y:=3;
    aRobot.z:=0;

    // Will use the local interface 'robot' to search for a remote interface named aRobot
    // and run the skill move() on the remote interface.
    aRobot.move();

    // Return value - running, done, failed or interrupted
    onRun:=processplan.done;
end.

```

Static interface declarations

Section: Static interface declarations

Syntax: static [handle] <name1:='ID1'> [, <name2:='ID2'>] [, ...] : interface;

Description: Declaration of static interfaces. The remote interface is mapped based on entity ID.

Example:

```

Program StaticExample;
{ Abstract remote interfaces }
static Robot:='interfaceID', Resource:='124' : interface;

Begin
    Robot.Position:=1;
    Robot.MoveTo();
    Resource.Signal:=TRUE;
    Resource.Motor();
End.

```

Global variable declarations

Section: Global variable declarations

Syntax:

```
var [constant] <name1> [, <name2>] [, ...] : {real | integer |  
boolean | string | structtype};
```

or

```
var [constant] <name1> [, <name2>] [, ...] : array [<startindex>  
.. <endindex>] of {real | integer | boolean | string | structtype  
};
```

Description: Declaration of variables or arrays that are accessible in the entire program and all functions. It is possible to declare several variables based on the same type on one line or it is possible to use several lines. During declaration is also possible to assign a default value. For structures the default value must be surrounded by () and all values must be paired with variable names.

Example:

```
var Ready : boolean;  
var Counter, Number : integer;  
var Data : array [1 .. 10] of real;  
var constant PI:=3.1415 of real;  
var pos:=(x:=1.0, y:=2.0) of Position;
```

Comment

Section: Comment

Syntax 1: { <multi line text> } | (* <multi line text> *)

Syntax 2: // <single line text>

Description: Comments are ignored by the compiler and are optional but can improve the readability of the code. Comment can be single line // that ends when the line ends or multiline { } that end at the end of the comment character }. Note that comments cannot be nested, i.e. a comment cannot contain another comment. The comment type { } is equivalent to (* *).

Example:

```
{ Program to write Hello World on the console  
The program will show the basic idea of structured text  
}  
  
// This comment is one line only!  
  
(* Multiline comment again *)
```

Function declaration

Section: Function declaration

Syntax 1 - Parameters:

```

    function <name>([<parameter 1> [, <parameter n>]:<parameter
type>] [; <parameter n>:<parameter type>] [; ...]) : { real | integer
| boolean | string | structtype};

[local variable declaration]
begin
    [function code]
end.

```

Syntax 2 – No parameters:

```

    function <name> : {real | integer | boolean | string |
structtype};

[local variable declaration]
begin
    [function code]
end.

```

Description: Functions can be declared to encapsulate a block of code to perform a specific action. Functions can be called from other functions or procedures and have a return value. The return value is handled by assigning the function name a value. A function can accept 0 or n number of parameters where <parameterx>\ represents parameter x name and <parameter typ> is a known type or struct. A function with no parameters can be called in two ways, <name>; or by <name>(); .e.g., f; or f();.

Example 1:

```

Program SquareFunction;
{ Function with return value }
function Square(Value: integer) : integer;
begin
    Square:=Value*Value; // Calculate and set return value
end.
Begin
    writeln('The square of 4 is ', Square(4));
End.

```

Example 2:

```

Program PI;
{ Function with no parameters }
function Pi : real;
begin
    Pi:=3.1415; // Constant expression
end.
Begin
    writeln('Pi=', Pi);
End.

```

Procedure declaration

Section: Procedure declaration

Syntax 1 - Parameters:

```

procedure <name>([<parameter 1>[, <parameter 2>]:<parameter
type>] [; <parameter n>:<parameter type>] [; ...]);

[local variable declaration]
begin

```

```
[procedure code]
end.
```

Syntax 2 - No parameters:

```
procedure <name>;
[local variable declaration]
begin
    [procedure code]
end.
```

Description: Procedures can be declared to encapsulate a block of code to perform a specific action. Procedures can be called from other functions or procedures but have no return value. A procedure can accept 0 or n number of parameters where <parameter x> represents parameter x name and <parameter type> is a known type or struct. A procedure with no parameters can be called in two ways, <name>; or by <name>(); .e.g., f; or f();.

Example:

```
program ProcedureExample;

procedure WriteSquare(Value: integer);
var Squared : integer;
begin
    writeln('The value is ', Value);
    Squared:=Value*Value;
    writeln('and the value squared is ', Squared);
end.
Begin
    WriteSquare(5); // Print the square of 5
End.
```

Main program block

Section: Main program block

Syntax:

```
begin
    [code]
end.
```

Description: Each program must have the main program block. The main program block is where the program starts and must be at the end of the program file. The main program block starts at begin and ends at end. Note that the main program block ends with a period (.).

Example:

```
Program HelloWorld;
{ Program to write Hello World! on the console }
Begin
    writeln('Hello World!');
End.
```

Control structures

The following control structures exist:

- **while**
- **repeat until**
- **for**
- **foreach**
- **for downto**
- **if ifelse else**
- **case**
- **exit**
- **break**
- **error**
- **label**

while

Control iterator: while

Syntax 1 - block:

```
while <expression> do
begin
    [code block]
end;
```

Syntax 2 – no block:

```
while <expression> do
    [single statement]
```

Description: Iterates the code block or statement as long as the expression evaluates to true.

Example:

```
Program WhileIterator;
var Counter :integer;
Begin
    Counter:=5;
    While Counter>0 do
        Begin
            writeln('Countdown ', Counter);
            Counter:=Counter-1;
        end;
End.
```

repeat

Control iterator: repeat

Syntax:

```
repeat
    [code block]
until <expression>;
```

Description: Iterates the code block as long as the expression evaluates to true.

Note, the expression is evaluated after the code block is executed. **Example:**

```
Program WhileIterator;
var Counter :integer;
```

```

Begin
  Counter:=5;
  repeat
    writeln('Countdown ', Counter);
    Counter:=Counter-1;
  until Counter>0;
End.

```

for

Control iterator: for to

Syntax 1 - block:

```

for <variable>:=<fromexpression> to <toexpression> do
begin
  [code block]
end;

```

Syntax 2 – no block:

```

for <variable>:=<fromexpression> to <toexpression> do
  [single statement]

```

Description: Iterates the code block or the statement n times with an integer variable as a counter. The iteration starts from <fromexpression> and counts up to <toexpression>.

Example:

```

Program ForIterator;
var Counter :integer;
Begin
  for Counter:=1 to 5 do
  begin
    writeln('Count up ', Counter);
  end;
End.

```

foreach

Control iterator: foreach in

Syntax 1 - block:

```

foreach <variable_name> in <iterablevariable> do
begin
  [code block]
end;

```

Description: Iterates the code block or the statement over all elements in the <iterablevariable>. The <variable_name> will be a name that can be used to reference each element in the <iterablevariable>.

Note

The <iterablevariable> must be of a type that can be iterable, e.g, arrays, structs or strings.

Example:

```

Program ForEachIterator;
var anArray: array [ 1 .. 4 ] of integer;

procedure onRun;
begin

```

```

{ Initialize the array to -1}
foreach i in anArray do
begin
    i:=-1;
end;

writeln(anArray);
end.

```

for dwonto

Control iterator: for dwonto

Syntax 1 - block:

```

for <variable>:=<fromexpression> downto <toexpression> do
begin
    [code block]
end;

```

Syntax 2 - no block:

```

for <variable>:=<fromexpression> downto <toexpression> do
[single statement]

```

Description: Iterates the code block or the statement n times with an integer variable as a counter. The iteration starts from fromexpression and countdown to toexpression.

Example:

```

Program ForDowntoIterator;
var Counter :integer;
Begin
    for Counter:=5 downto 1 do
    begin
        writeln('Countdown ', Counter);
    end;
End.

```

if else if else

Control selection: if ifelse else

Syntax 1:

```

if <expression1> then
begin
    [code block]
end;

[else if <expression2> then
begin
    [code block]
end;]

[else if <expression_n> then
...
]
```

```

[else
begin
  [code block]
end;]

Syntax 2:
  if <expression1> then
    <single statement>
  [else if <expression2> then
    <single statement>]
  [else if <expression_n> then
    ...
  ]
  [else
    <single statement>]

```

Description: Selection control.

Example:

```

Program IfElseIFElse;
var a, b : integer;
Begin
  writeln('Enter two integers');
  readln(a, b);
  if a<b then
    begin
      writeln('a<b');
    end;
  else if a>b then
    begin
      writeln('a>b');
    end;
  else
    begin
      writeln('a=b');
    end;
End.

```

Note

For syntax 2 each single line expression ends with a semicolon ;

However, it can be safer to work with blocks as in syntax 1 for complex or nested if/else sections.

```

if a>0 then
  writeln('Greater');
else if a=0 then
  writeln('Zero');
else
  writeln('Less');

```

case

Control selection: case

Syntax:

```

  case <expression> of
    label_1: [code block]

```

```

...
[label_n:] [code block]
[else:] [code block]
end;

```

Description: Case executes a block of code based on the value of an <expression>.

The **else** clause is optional and can be used to handle values not matched by any label.

Note: <expression> must be of type **integer**, **boolean**, or **string**. Labels must be constants, such as 1, 2, 3, etc. Alternatively, a label can reference a constant variable whose value will be used.

Example:

```

program CaseExample;

var Input:=0: integer;

begin
  writeln('Press 1 to start and 2 to stop:');
  readln(Input);
  case Input of
    1:
      writeln('Start');
    2:
      writeln('Stop');
    else:
      writeln('Unknown option');
  end;
end.

```

exit

Control sequence: exit

Description: Exit exits the current procedure or function and returns control to the caller. If invoked in the main program block, exit stops the program.

Example:

```

Program ExitEarly;
var a : integer;
Begin
  for a:=1 to 10 do
    begin
      if a=5 then
        begin
          exit;
        end;
      writeln('a = ', a);
    end;
End.

```

break

Control sequence: break

Description: Break breaks the current loop and continues to execute after the end of the loop.

Example:

```

Program BreakEarly;
var a : integer;
Begin
    for a:=1 to 10 do
    begin
        if a=5 then
        begin
            break;
        end;
        WriteLn('a = ', a);
    end;
    WriteLn('Loop is ready');
End.

```

error

Control sequence: error

Syntax: error <message>

Description: Error will generate a user error and stops the current program to display a message.

Example:

```

Program Error_Example;

Var a, b : Integer;

Begin
    a:=random(5);
    b:=random(5);

    if b=0 then
        error 'cannot divide by zero';

    writeln(a,'/',b,'=',inttoreal(a)/inttoreal(b));
End.

```

label

Control sequence: label

Syntax: <name>:

Description: A label is a constant value used in a case statement to identify a specific code block. Labels must be constants, such as 1, 2, 3, etc. Alternatively, a label can reference a constant variable, in which case its value is used.

Example:

```

program CaseExample;

var Input:=0: integer;

var CONSTANT SEQ_1:=1, SEQ_2:=2: Integer;

begin
    Input=getValue();

    case Input of
    SEQ_1:
        writeln('Will start sequence ', SEQ_1);
    SEQ_2:
        writeln('Will start sequence ', SEQ_2);
    else:
        writeln('Unknown sequence');
    end;
end.

```

Operators

The following operators exist:

Arithmetic

Operator	Notation
Division	/
Addition	+
Subtraction	-
Multiplication	*
Negative (unary minus)	-
Modulus	mod

Relations

Operator	Notation
Equeal	=
Less equal	<=
Less	<
More equal	>=
More	>
Not equal	\diamond

Logical

Operator	Notation
Or	or
Xor	xor
And	and
Not	not

Note

Will only operate on booleans.

Bitwise

Operator	Notation
Or	or
Xor	xor
And	and
Complement	not

Shift left	<<
Shift right	>>

Note

Will only operate on integers.

Object

Operator	Notation
Access member operator	.

Array

Operator	Notation
Index operator	[]

Types

The following types exist:

- **REAL**
- **INTEGER**
- **BOOLEAN**
- **STRING**
- **STRUCT**
- **OBJECT**

Real

Real syntax: [-] number [.number]

Example:

```
Var aVariable : real;
aVariable := -123.456;
```

Integer

Integer syntax: [-] number

Example:

```
Var aVariable: integer;
aVariable:=-1234;
```

Boolean

Boolean syntax: [true | false]

Example:

```
Var aVariable: boolean;  
aVariable:=true;
```

String

String syntax: 'text'

Example:

```
var aVariable : string;  
avariable:='Text message';
```

Struct

Struct type variables are data structures based on a user defined type. User defined types are global and act as templates to be used when variables are declared. A **Struct type** consist of base types (integer, real, boolean or string) or other structs.

Objects

Objects cannot be defined within a program. They exist from the start and they are instances of structs. All agents have an object defined called agent. All interfaces are defined as objects one per interface and share the name of the interface. It is possible to refer to interfaces on other agents by declaring abstract interfaces. Use the dot operator to access any member of an object, variables and skills. If the agent has goals the goal variables are defined under the object called goal.

The following objects exists:

this

The entity representing the program. Example of an entity is a goal.

agent

Agent is an object representing the agent running the code.

interface

Interface is an object that represents an interface on the current agent (local)

remoteinterface

Remoteinterface is an object that represents an interface on an remote agent during negotiation.

Goal

Goal is an object that represents a goal, only process plans can reach a goal objects

ProcessPlan

Processplan is an object that contains possible return values for an onRun() function. Processplan has the following members:

Member	Value	Description
running	1	Function is still running and must be called next cycle
done	2	Function is done
failed	3	Function has failed and cannot continue
interrupted	4	Function was interrupted during the execution
replan	5	Function has run in to a problem and the entire solution must be replanned might trigger a new negotiation
redo	6	Function failed and want to be restarted

Example:

```
Abstract Interface anInterface;

{ will change the value of variable MyVariable defined on agent }
agent.MyVariable:=1;

{ Will change the value of variable A defined on a local interface }
interface.A:=true;

{ will change the value of variable Run defined on a remote abstract interface referred
to as anInterface }
anInterface.Run:=true;

{ Will run the skill Execute on a remote abstract interface and at the same time set the
Run variable to true }
anInterface.Execute();

{ Will print the name of the entity }
writeln(this.Name);

{ For process plans it is possible to access variables defined for goals }
WriteLn('Goal position is ', goal.Pos);

{ Will set return value to done }
onRun:=processplan.done;
```

Procedures

Input and output

writeln

Procedure: writeln

Syntax: writeln([<expression>] [, <expression>] [, ...])

Description: Write with new line

All expressions will be written to the console and a new line will be added to the end.
Example:

```
{ Hello World to the console }
writeln('Hello World!');

{ Empty will only output a new line }
writeln();

{ Multi expressions }
writeln('The value of a is ', a, ' and b is ', b);
```

write

Procedure: write

Syntax: write([<expression>] [, <expression>] [, ...])

Description: Write without a new line

All expressions will be written to the console. **Example:**

```
{ The two write() outputs will be written on the same line }
write('The value of a is ', a);
write(' and b is ', b);
```

readln

Procedure: readln

Syntax: readln([<variable_1>] [, <variable_2>] [, ...])

Description: Will read input from the console window and store the input in the given variables.

Example:

```
program Readln_Example;

function Menu() : integer;
var input:string;
begin
    writeln(' 1 - Write hello');
    writeln(' 2 - Write world');
    write('Selection: ');

    // Repeat until a valid integer is read
    repeat
        readln(input); // Read user input from console
        until not stringisinteger(input);
        Menu:=stringtoint(input);
    end.

var m:integer;
begin
    m:=Menu();
    if m=1 then
        writeln('hello');
    else if m=2 then
        writeln('world');
    else
        writeln('wrong menu');
end.
```

Abstract Interface Functions

abstractname

Procedure: abstractname

Syntax: abstractname(<interface>] [, <string_name>]) : string

Description: Will get the <interface> name and if <string_name> is given also change the name.

Return value: the new name if changed otherwise current name.

Example:

```
define handle example : interface;
begin
    var x, y : integer;
end;

begin

    example.negotiate(); // Will negotiate with the name example
    example.run();

    abstractname(example, 'process'); Will negotiate with the name process
    example.run();
end.
```

abstractcost

Procedure: abstractcost

Syntax: abstractcost(<interface>]) : integer

Description: Will get the total cost to call all planed skills on the abstract interface <interface>. The cost will only include the skills used by the abstract <interface>. If the <interface> is not negotiated the cost returned is 0.

Return value: the total cost.

Example:

```
abstract aResource : interface;

begin
    aResource.S1();
    aResource.S2();
    writeln('The cost to execute S1 and S2 is ', abstractcost(aResource));

end.
```

abstractshare

Procedure: abstractshare

Syntax: abstractshare(<string_name>, <interface>) : boolean

Description: Will add the <interface> as a global abstract interface available for all skills on the same agent. The global abstract interface can be found with the help of the <string_name>

Return value: true if it was possible to add the abstract interface.

Example:

```
define handle example : interface;
begin
    var x, y : integer;
end;
```

```

begin
    example.negotiateandbook();
    example.run();
    abstractshare('example', example); // Share the interface
end.

```

abstractfind

Procedure: abstractfind

Syntax: abstractfind(<string_name>, <interface>) : boolean

Description: Will search for a shared abstract interface by the name <string_name>. If an interface is found it is associated with the <interface>. All variables and functions for the global interface will be available on the local defined interface after a successful find.

Return value: true if it was possible to find a global interface.

Example:

```

define handle example : interface;
begin
end;

begin
    // Find an existing abstract interface
    if not abstractfind('example', example) then error 'unable to find interface';

    example.x:=1;
    example.y:=2;
    example.run();
end.

```

ignorebooked

Procedure: ignorebooked

Syntax: ignorebooked(<interface>] [, <boolean_value>]) : boolean

Description: Will get the <interface> property IgnoreBooked and if <boolean_value> is given also change the value. This property affects negotiation process and how to handle remote interfaces.

Return value: the new vaule if changed otherwise current value.

Example:

```

define handle example : interface;
begin
    var x, y : integer;
end;

begin

    example.ignoreBooked(example, false);
    example.negotiate(); // Will negotiate with the name example
    example.run();
end.

```

ignorebusy

Procedure: ignorebusy

Syntax: ignorebusy(<interface>] [, <boolean_value>]) : boolean

Description: Will get the <interface> property IgnoreBusy and if <boolean_value> is given also change the value.

This property affects negotiation process and how to handle remote interfaces.

Return value: the new vaule if changed otherwise current value.

Example:

```
define handle example : interface;
begin
    var x, y : integer;
end;

begin
    example.ignoreBusy(example, false);
    example.negotiate(); // Will negotiate with the name example
    example.run();
end.
```

Test Functions

Procedure: isreal | isboolean | isinteger | isstring

Syntax: is<type>(<expression>) : boolean

Description: Will test if the <expression> is of the type <type>.

Return value: true if <expression> is of specific type <type> otherwise false.

Example:

```
Var a:=3.14: real;

begin
    { Will write true }
    writeln(isreal(7.3));
    writeln(isreal(a));
    writeln(isstring('Hello'));

    { Will write false }
    writeln(isreal('Hello'));
    writeln(isstring(7.3));
end.
```

Procedure: stringisreal | stringisboolean | stringisinteger

Syntax: stringis<type>(<string>) : boolean

Description: Will test if the content of the <string> can be interpreted and converted to the type <type>.

Return value: true if <string> can be converted otherwise false.

Example:

```
{ Will write true }
writeln(stringisreal('7.3'));
writeln(stringisreal('-7'));

{ Will write false }
writeln(stringisinteger('Hello'));
writeln(stringisinteger('7.3'));
```

Procedure: stringisempty

Syntax: stringisempty(<string>) : boolean

Description: Will test if the <string> is empty or not.

Return value: true if <string> is empty otherwise false .

Example:

```
{ Will write true }
writeln(stringisempty(''));

{ Will write false }
writeln(stringisempty('Hello'));
```

Converting Functions

Procedure: <type_from>to<type_to>

Syntax: <type_from>to<type_to>(<expression>) : <type_to>

Description: Will convert the <expression> value from the <type_from> to <type_to>.

Return value: The <expression> value converted to <type_to>.

Note

The following combinations exists

- realtoint()
- inttoint()
- booleantoint()
- stringtoint()
- realtoreal()
- inttoreal()
- booleantoreal()
- stringtoreal()
- realtoboolan()
- inttoboolan()
- booleantoboolan()
- stringtoboolan()
- realtostring()
- inttostring()
- booleantostring()
- stringtostring()

Example:

```
Program Example_Convert;

Var a:=1:Integer;
Var b:=3.1:Real;
Var c:=false:Boolean;
Var d:='Hello':String;

Begin
    writeln('a is ', a,
            ' b is ', b,
            ' c is ', c,
            ' and d is ', d);

    c:=inttoboolan(a);
    d:=realtostring(b);
    a:=realtoint(b);
    b:=booleantoreal(c);

    writeln('and now a is ', a,
            ' b is ', b,
            ' c is ', c,
            ' and d is ', d);
End.
```

String Functions

len

Procedure: len

Syntax: len(<string>) : integer

Description: Will return the length of the <string>.

Return value: <string> length as an integer.

Example:

```
Program Example_String_Functions;
Var s:='Hello World' : String;
Begin
    writeln('string s is of length ', len(s));           // Will print 11
    writeln('the first 5 letters of s ', left(s, 5));   // Will print Hello
    writeln('the last 5 letters of s ', right(s, 5));   // Will print World
    writeln('the mid part of s ', mid(s, 5 ,3));       // Will print lo Wo
End.
```

left

Procedure: left

Syntax: left(<string>, <integer>) : string

Description: Will return a new string based on <string> containing <integer> number of characters from the left side of the <string>.

Return value: A new string.

Example:

```
Program Example_String_Functions;
Var s:='Hello World' : String;
Begin
    writeln('string s is of length ', len(s));           // Will print 11
    writeln('the first 5 letters of s ', left(s, 5));   // Will print Hello
    writeln('the last 5 letters of s ', right(s, 5));   // Will print World
    writeln('the mid part of s ', mid(s, 5 ,3));       // Will print lo Wo
End.
```

right

Procedure: right

Syntax: right(<string>, <integer>) : string

Description: Will return a new string based on <string> containing <integer> number of characters from the right side of the <string>.

Return value: A new string.

Example:

```
Program Example_String_Functions;
Var s:='Hello World' : String;
Begin
    writeln('string s is of length ', len(s));           // Will print 11
    writeln('the first 5 letters of s ', left(s, 5));   // Will print Hello
```

```

writeln('the last 5 letters of s ', right(s, 5)); // Will print World
writeln('the mid part of s ', mid(s, 5 ,3)); // Will print lo Wo
End.

```

mid

Procedure: mid

Syntax: mid(<string>, <integer_start>, <integer_length>) : string

Description: Will return a new string based on <string> containing <integer_length> number of characters starting from <integer_start> of the <string> (zero based index).

Return value: A new string.

Example:

```

Program Example_String_Functions;

Var s:='Hello World' : String;
Begin
    writeln('string s is of length ', len(s)); // Will print 11
    writeln('the first 5 letters of s ', left(s, 5)); // Will print Hello
    writeln('the last 5 letters of s ', right(s, 5)); // Will print World
    writeln('the mid part of s ', mid(s, 5 ,3)); // Will print lo Wo
End.

```

trim

Procedure: trim

Syntax: trim(<string>) : string

Description: Will return a new string without any leading or trailing whitespace based on <string>.

Return value: a trimmed string without leading or trailing whitespace.

Example:

```

Program Example_String_Functions;

Var s:='Hello World ' : String;
Begin
    writeln('-', s, '-'); // Will print - Hello World -
    writeln('-', trim(s), '-'); // Will print - Hello World -
End.

```

replace

Procedure: replace

Syntax: replace(<string_input>, <string_find>, <string_replace>) : string

Description: Will return a new string based on <string_input> where all occurrences of <string_find> is replaced with <string_replace>. If <string_replace> is empty all occurrences of <string_find> will be deleted.

Return value: a new string that does not contain <string_find>.

Example:

```

Program Example_String_Functions;

Var s:='Hello World' : String;
Begin

```

```

writeln(replace(s, 'l', '.'));           // Will print He..o Wor.d
writeln(replace(s, 'l', ''));            // Will print Heo Word
End.

```

contains

Procedure: contains

Syntax: contains(<string_input>, <string_find>) : boolean

Description: Check if <string_input> contains <string_find>.

Return value: True if <string_find> is a part of <string_input>.

Example:

```

Program Example_String_Functions;

Var s:='Hello World' : String;
Begin
    writeln(contains(s, 'Hello'));           // Will print true
    writeln(contains(s, 'hello'));           // Will print false
End.

```

tolowercase

Procedure: tolowercase

Syntax: tolowercase(<string>) : string

Description: Converts all of the characters in <string> to lowercase.

Return value: a new string with all character in lowercase.

Example:

```

Program Example_String_Functions;

Var s:='Hello World' : String;
Begin
    writeln(tolowercase(s));               // Will print hello world
End.

```

touppercase

Procedure: touppercase

Syntax: touppercase(<string>) : string

Description: Converts all of the characters in <string> to uppercase.

Return value: a new string with all character in uppercase.

Example:

```

Program Example_String_Functions;

Var s:='Hello World' : String;
Begin
    writeln(touppercase(s));              // Will print HELLO WORLD
End.

```

split

Procedure: split

Syntax: split(<string>, <string_find>) : string

Description: Will split <string> if the <string_find> is found and return the first part.

Return value: a new string.

Example:

```
Program Example_String_Functions;
Var s:='Hello World' : String;
Begin
  writeln(split(s, ' '));           // Will print Hello
End.
```

tostring

Procedure: tostring

Syntax: tostring(<expression>) : string

Description: Will return a string based on the <expression>.

Return value: <expression> converted to a string.

Example:

```
Program Example_String_Functions;
Var value:=4 : Integer;
Var strvalue : String;
Begin
  strvalue:=toString(value*2);      // Convert expression to a string
  writeln('strvalue=', strvalue);   // Will print 8
End.
```

JSON functions

GetJElement

Procedure: getjelement

Syntax: getjelement(<string_json>, <string_element>) : string

Description: Will get <string_elemt> value from a JSON string <string_json>.

Return value: the value as a string.

Example:

```
Program Example_JSON;
Var JSONString:'{"value": "Hello World"}' : String;
Begin
  WriteLn(getjelement(JSONString, 'value'));           // Will print Hello World
End.
```

SetJElement

Procedure: setjelement

Syntax: setjelement (<string_json>, <string_element>, <string_value>) : string

Description: Will set <string_element> to <string_value> in the JSON string <string_json>.

Return value: a new JSON string with the new value.

Example:

```
Program Example_JSON;
Var JSONString:='{"value": "Hello World"}' : String;
Begin
    WriteLn(setjelement(JSONString, 'value', 'HI!'));           // Will print
 {"value": "HI!"}
End.
```

JToStruct

Function: jtostruct

Syntax: jtostruct (<string json>) : struct

Description: Will return a new struct based on the JSON string <string_json>.

Return value: a new struct with the values based on `<string json>`.

Example:

```
Program Example_JSON;
Var JSONString:='{"value": "Hello World"}' : String;

struct object;
begin
    var value : string;
end;

var data:object;

Begin
    data:=jtostruct(JSONString);
    WriteLn(data);           // Will print {"value":"Hello World!"}
End.
```

StructToJ

Function: structtoj

Syntax: structtoij (<struct>) : string

Description: Will return a new JSON string based on the <struct>.

Return value: a new JSON string.

Example:

```
Program Example_JSON;  
struct object;  
begin  
    var value : string;
```

```

end;

var data:=(value:='hello world!'):object;
var str:string;

Begin
    str:=structtoj(data);
    WriteLn(str);           // Will print {"value":"Hello World!"}
End.

```

IsJson

Function: isjson

Syntax: isjson(<string>) : boolean

Description: Will return true if string is a valid JSON description.

Return value: true if string is valid otherwise false.

Example:

```

Program Example_JSON;

Begin
    writeln(isjson('{"x":1}'));           // Will print TRUE
End.

```

Math functions

Procedure: <math_function>

Syntax: <math_function>(<expression>) : integer | real

Description: The <math function> accept a real or integer <expression>. The math functions will return a value based on the input type, i.e., if the <expression> input type is integer the output will be of type integer. Note that, the min and max functions requires two expressions that must match by type.

Return value: An integer or a real number.

Example:

```

Program Math;

Begin
    WriteLn( abs( -1) );   { Will write 1 }
    WriteLn( abs( -1.0) ); { Will write 1.0 }

    WriteLn( random(10) ); { Will write a random number between 0-9 }
    WriteLn( random(1.0) ); { Will write a random number between 0.0 and 1.0 }

    WriteLn( min(3, 2) );  { Will write 2 }
    WriteLn( max(3, 2) );  { Will write 3 }

End.

```

List of all math functions:

Function name	Description
abs	Absolute value of <expression>

ceil	Round <expression> up to nearest integer value.
floor	Round <expression> down to nearest integer value.
round	Round <expression> to the nearest integer value.
max	Returns the greater value of <expression1> and <expression2>
min	Returns the smaller value of <expression1> and <expression2>

Function name	Description
exp	e raised to the power of <expression>
log	natural logarithm of <expression>
log10	10 logarithm of <expression>
sqrt	square root of <expression>

Function name	Description
sin	trigonometric sine of an angle in radians
cos	trigonometric cosine of an angle in radians
tan	trigonometric tangent of an angle in radians
asin	arc sine in radians
acos	arc cosine in radians
atan	arc tangent in radians
sinh	hyperbolic sine
cosh	hyperbolic cosine
tanh	hyperbolic tangent
todegrees	convert radians to degrees
toradians	convert degrees to radians

Function name	Description
random	generate a pseudo random number between 0 and <expression>

Array Functions

length

Procedure: length

Syntax: length (<any_array>) : integer

Description: Will return the length of an array.

Return value: Array length.

Example:

```
Program Example_Array;

var MyArray : array [1 .. 3] of real;
var Sum:=0.0 : real;
var i : integer;
var Input : string;

Begin
  for i:=startindex(MyArray) to endindex(MyArray) do
  begin
    Input:=InputDialog('Give input for index ' + inttostring(i), '1');
    if stringisreal(Input) then
      MyArray[i]:=stringtoreal(Input);
  end;

  for i:=startindex(MyArray) to endindex(MyArray) do
  begin
    Sum:=Sum+MyArray[i];
  end;

  writeln('The sum of the array is ', Sum, ' and the average value ',
  Sum/inttoreal(length(MyArray)));
End.
```

startindex

Procedure: startindex

Syntax: startindex(<any_array>) : integer

Description: Will return the first index of an array.

Return value: First index.

Example: see length.

endindex

Procedure: endindex

Syntax: endindex(<any_array>) : integer

Description: Will return the last index of an array.

Return value: Last index.

Example: see length.

HMI

messagebox

Procedure: messagebox

Syntax: messagebox(<string_message>, [<string_title>], [<integer_options>]) : integer

Description: Will show an message box with a user message <string_message>. It is possible, but optional, to give the dialog a title <string_title>. <integer_option> gives a possibility to change the type of message and number of buttons, see example for the values.

Return value: The button pressed, see example for values.

Note

The constants used in the example, e.g, DIALOG_SHOW_YES_NO, are also defined in the system file messagebox.inc. An easier approach than defining all constants is to included all by the use of \$include 'system:messagebox.inc'

Example:

```
{ An example program how to show HMI messages }
Program MessageBox_Example;

{ Return value constants for messagebox }
Var CONSTANT
    DIALOG_ANSWER_YES:=0,
    DIALOG_ANSWER_NO:=1,
    DIALOG_ANSWER_CANCEL:=2,
    DIALOG_ANSWER_CLOSED:=-1 :Integer;

{ Option constants for type of messagebox to show }
Var CONSTANT
    DIALOG_SHOW_YES_NO:=0,
    DIALOG_SHOW_YES_NO_CANCEL:=1,
    DIALOG_SHOW_OK_CANCEL:=2,
    DIALOG_ERROR_MESSAGE:=3,
    DIALOG_INFORMATION_MESSAGE:=4,
    DIALOG_WARNING_MESSAGE:=5,
    DIALOG_QUESTION_MESSAGE:=6,
    DIALOG_PLAIN_MESSAGE:=-1 :Integer;

{ Main procedure }
Var Result : Integer;
Begin
    Result:=MessageBox('Press a button', 'Title', DIALOG_SHOW_YES_NO_CANCEL);

    if Result=DIALOG_ANSWER_YES then
    begin
        WriteLn('You pressed Yes');
    end
    else if Result=DIALOG_ANSWER_NO then
    begin
        WriteLn('You pressed No');
    end
    else if Result=DIALOG_ANSWER_CANCEL then
    begin
        WriteLn('You pressed Cancel');
    end
    else
    begin
        WriteLn('You closed the window');
    end;
End.
```

inputdialog

Procedure: inputdialog

Syntax: inputdialog(<string_prompt>, [<string_default_value>]) : string

Description: Will show an input dialog window containing a message <string_prompt> and

an input field with a default value of <string_default_value>. The user can input any string value in the dialog. **Return value:** The value entered by the user or an empty string if the user clicks cancel.

Example:

```
{ An example program how to get user input strings from HMI }
Program InputDialog_Example;

Var Result : String;
Var DefaultValue:='3' : String;
Var Iterator: Integer;

{ Main procedure }
Begin
    Result:=InputDialog('How many iterations?', DefaultValue);

    if not StringIsEmpty(Result) and StringIsInteger(Result) then
    begin
        Iterator:=StringToInt(Result);

        while Iterator>0 do
        begin
            WriteLn('Loop - ', Iterator);
            Iterator:=Iterator-1;
        end;
    end;
End.
```

optiondialog

Procedure: optiondialog

Syntax 1 : optiondialog(<integer_handle>, <integer_value>, <string_message>, <string_title>, <string_option_1> [<string_option_2>] [, ...]) : boolean

Syntax 2 : optiondialog(<integer_handle>, <boolean_condition>) : boolean

Description: Will show a message box with a user message <string_message> and a title <string_title>. <string_option_1> is a string associated with a button 1. It is possible to give many <string_options> to create many buttons. The integer handle must be unique to create a new dialog. A handle can be reused if the dialog is closed in between or if you want to refer to an already open dialog.

Note

The OptionDialog call is MODELESS, i.e., the execution of the code will not stop and wait for an answer. The code must handle this by constantly checking the <integer_value> to get the result.

For syntax 2 it is possible to close an open OptionDialog by setting the <boolean_condition> to false. Further, if the handle is -1 all open OptionDialogs will be closed. If <boolean_condition> is true it is possible to check the return value to see if the dialog is open or closed.

Note

An attempt to close an already closed OptionDialog will generate an error.

Return value: For syntax 1 True if the dialog was created as a new dialog, false if not. For syntax 2 True if the dialog is open or false if closed.

Example:

```
Program ExampleOptionDialog;
Var Selection : Integer;
{ Main procedure }
Begin
    Selection:=-1;
    OptionDialog(1, Selection, 'Press a button', 'Selection', 'Button 1', 'Button 2',
'Button 3');

    While Selection=-1 do
    Begin
        sleep(50);
    End;

    writeln('You selected button number ', Selection);

    // Check if dialog is still open
    if optiondialog(1, true) then
        optiondialog(1, false); // Close the dialog
End.
```

System Functions

Time

sleep

Procedure: sleep

Syntax 1: sleep(<integer_time>)

Syntax 2: sleep(<string_time>)

Description: Will causes the execution to sleep for the specified number of milliseconds <integer_time>. For syntax 2 a string is accepted that must include the time with an optional suffix. Valid suffixes include: h (hour), m (minutes), s (seconds), and ms (milliseconds). For example the string, '2s' represents 2 seconds.

Example:

```
Program Sleep_Example;
Begin
    WriteLn('Wait for it ...');
    sleep(3000);
    WriteLn('wait for it ...');
    sleep('3s');
    WriteLn('legendary!');
End.
```

time

Procedure: time

Syntax: time() : integer

Description: Will return the time since start as an integer.

Return value: <integer>

Example:

```
Program Time_Example;
Var t:integer;
Begin
  t=time();
  sleep(100);
  writeln(time()-t); // Will write approx. 100
End.
```

Preferences

preferencevalue

Procedure: preferencevalue

Syntax: preferencevalue(<string_name>, [<value>>]) : string

Description: Will set or get a global preference value. All available values can be found in the Preferences. **Return value:** <string>

Example:

```
Program Example;
Begin
  writeln('RetryOnBusy=', preferencevalue('RetryOnBusy', false));
End.
```

Debug

threadmessage

Procedure: threadmessage

Syntax: threadmessage([<string_message>]) : string

Description: Will set or get the user message for current agent thread. A thread user message can be seen in the Agent thread viewer during runtime and the user message is ignored by the system. It can be used to display various information for e.g. debug purposes.

Return value: <string>

Example:

```
Program Example;
Begin
  threadmessage('Init of agent - running constructor');
End.
```

File Functions

open

Procedure: open

Syntax: open(<string_filename>, <integer_filemode>) : integer

Description: Will open the <string_filename> file. The <integer_filemode> decides if the file is opened for reading or writing.

Return value: a file handle (integer) that can be used to access the file. If file handle<0 an error occurred and the file cannot be opened.

Example:

```
program FileExample;
$include 'system:file.inc'

var fileHandle:integer;

begin
    fileHandle:=open('test.txt', MODE.WRITE_TEXT);
    writeln(fileHandle, 'Hello World!');
    writeln(fileHandle, 1);
    writeln(fileHandle, true);
    close(fileHandle);
end.
```

MODE

Mode is an object that contains possible values for open() and is defined in the system include file file.inc. Use \$include 'system:file.inc' to include the object. Mode has the following members:

Member	Value	Description
WRITE_TEXT	1	Create a new text file for writing.
READ_TEXT	2	Open an existing text file for reading.

close

Procedure: close

Syntax: close(<integer_filehandle>) : boolean

Description: Will close the <integer_filehandle> file. The <integer_handle> must be a valid file handle returned from an open.

Return value: true if the file was closed otherwise false.

Example:

```
program FileExample;
$include 'system:file.inc'

var fileHandle:integer;

begin
    fileHandle:=open('test.txt', MODE.WRITE_TEXT);
    writeln(fileHandle, 'Hello World!');
    writeln(fileHandle, 1);
    writeln(fileHandle, true);
    close(fileHandle);
end.
```

fwriteln

Procedure: writeln

Procedure: fwrite

Syntax: writeln(<integer_filehandle>, [<expression>] [, <expression>] [, ...])

Description: Will print all <expression> to the <integer_filehandle> file. The <integer_handle> must be a valid file handle. writeln will add a new line after each write and fwrite will not add any new line.

Example:

```
program FileExample;
$include 'system:file.inc'

var fileHandle:integer;

begin
    fileHandle:=open('test.txt', MODE.WRITE_TEXT);
    writeln(fileHandle, 'Hello World!');
    writeln(fileHandle, 1);
    writeln(fileHandle, true);
    close(fileHandle);
end.
```

freadln

Procedure: readln

Syntax: readln(<integer_filehandle>, [<variable_1>] [, <variable_2>] [, ...]) : boolean

Description: Will read values from the <integer_filehandle> file and store the result in <variable>. The <integer_handle> must be a valid file handle returned from an open.

Return value: Number of <variable> assigned a value.

Example:

```
program FileExample;
$include 'system:file.inc'

var fileHandle : integer;
var s:string;
var i:integer;
var b:boolean;

procedure readFromFile1;
begin
    fileHandle:=open('test.txt', MODE.READ_TEXT);

    // Read each line in the file and convert to a specific variable type
    readln(fileHandle, s);
    readln(fileHandle, i);
    readln(fileHandle, b);

    writeln(s);
    writeln(i);
    writeln(b);
    close(fileHandle);
end.

procedure readFromFile2;
begin
    fileHandle:=open('test.txt', MODE.READ_TEXT);

    // Treat each line in the file as a string
    while readln(fileHandle, s)>0 do
```

```

begin
    writeln(s);
end;

close(fileHandle);
end.

begin
    // Will generate the same output on the screen
    readFromFile1();
    readFromFile2();
end.

```

Server API Functions

Pause

pauseAll

Procedure: pauseAll

Syntax: pauseAll()

Description: Will pause all running agents.

Example:

```

Begin
    pauseAll();
End.

```

pausebyname

Procedure: pauseByName

Procedure: pauseById

Procedure: pauseByInstanceId

Syntax: pauseByName(<string_ID>)

Syntax: pauseByID(<string_ID>)

Syntax: pauseByInstanceId(<string_ID>)

Description: Will pause all running agents that match <string_ID>.

Example:

```

Begin
    // Will pause all drill agents
    pauseByName('drill');

    // Will pause all agent with a ID=1
    pauseByID('1');
End.

```

Console and log

clearconsole

Procedure: clearconsole

Syntax: clearconsole()

Description: Will clear the console window.

Example:

```
Begin
    clearconsole();
    writeln('Hello World!')
End.
```

clearlog

Procedure: clearlog

Syntax: clearlog()

Description: Will clear the log window.

Example:

```
Begin
    clearlog();
End.
```

Runtime control

stopsystem

Procedure: stopsystem

Syntax: stopsystem()

Description: Will stop the runtime environment.

Example:

```
Begin
    stopsystem();
End.
```

restartsystem

Procedure: restartsystem

Syntax: restartsystem()

Description: Will restart the runtime environment.

Example:

```
Begin
    restartsystem();
End.
```

getagentsbyname

Procedure: getagentsbyname

Syntax: getagentsbyname (<list>, <string_id>)

Description: Will get all running agents that match <string_ID> and store all instance ids in the string array <list>.

Example:

```
program Example_GetAgents;
Var list: array [ 1 .. 2 ] of String;
procedure onDeploy;
begin
    getagentsbyname(list, 'drill');
    writeln(list);
End.
```

getagentsbyid

Procedure: getagentsbyid

Syntax: getagentsbyid(<list>, <string_id>)

Description: Will get all running agents that match <string_ID> and store all instance ids in the string array <list>.

Example:

```
program Example_GetAgents;
Var list: array [ 1 .. 2 ] of String;
procedure onDeploy;
begin
    getagentsByID(list, 'drill');
    writeln(list);
End.
```

deploybyandattach

Procedure: deployandattach

Syntax 1: `deployandattach(<string_name>, <string_deployinterfacename>, <string_localinterfaceid>) : boolean`

Description: Will deploy an agent that match the name with <string_name> and attach the newly deployed agent interface named <string_deployinterfacename> to a local interface with an ID of <string_localinterfaceid>.

Syntax 2: `deployandattach(<string_name>) : boolean`

Description: Will deploy an agent that match <string_name> and attach the new agent interface to an compatible interface of the calling agent. If the deployed agent is a part the goals are also searched to check for compatible interfaces.

Return value: **true** if any agent was deployed and attached otherwise **false**.

Example:

```
program Example;
```

```
begin
    DeployAndAttach('drill');
End.
```

deploybyinstanceid

deploybyid

deploybyname

Procedure: deploybyname

Syntax: deploybyname(<string_id>) : boolean

Syntax: deploybyinstanceid(<string_id>) : boolean

Syntax: deploybyid(<string_id>) : boolean

Description: Will deploy agents that match <string_ID>.

Return value: true if any agent was deployed otherwise false .

Example:

```
program Example;
begin
    DeployByName('drill');
End.
```

deployall

Procedure: deployall

Syntax: deployall() : integer

Description: Will deploy all agents in the depot.

Return value: number of agents deployed.

Example:

```
program Example;
var n:integer;
begin
    n:=DeployAll();
    writeln('deployed ', n, ' agents.');
End.
```

undeploybyname

Procedure: undeploybyname

Syntax: undeploybyname(<string_id>) : boolean

Description: Will undeploy agents that match <string_ID>.

Return value: true if any agent was undeployed otherwise false .

Example:

```
program Example;
begin
    UnDeployByName('drill');
End.
```

undeploybyid

Procedure: undeploybyid

Syntax: undeploybyid(<string_id>) : boolean

Description: Will undeploy agents that match <string_ID>.

Return value: true if any agent was undeployed otherwise false .

Example:

```
program Example;
begin
    UnDeployByID('idResource');
End.
```

undeploybyinstanceid

Procedure: undeploybyinstanceid

Syntax: undeploybyinstanceid(<string_id>) : boolean

Description: Will undeploy agents that match <string_ID>.

Return value: true if any agent was undeployed otherwise false .

Example:

```
program Example;
begin
    UnDeployByInstanceID(agent.InstanceID);
End.
```

undeployall

Procedure: undeployall

Syntax: undeployall() : integer

Description: Will undeploy all agents.

Return value: number of agents undeployed.

Example:

```
program Example;
var n:integer;
begin
    n:=UnDeployAll();
    writeln('undeployed ', n, ' agents.');
End.
```

getnumberofrunningparts

Procedure: getnumberofrunningparts

Syntax: getnumberofrunningparts() : integer

Description: Will count number of running part agents.

Return value: number of running parts.

Example:

```
program Example;
var n:integer;
begin
  n:=getnumberofrunningparts();
  writeln('Number of parts ', n);
End.
```

getnumberofrunningresources

Procedure: getnumberofrunningresources

Syntax: getnumberofrunningresources() : integer

Description: Will count number of running resource agents.

Return value: number of running resources.

Example:

```
program Example;
var n:integer;
begin
  n:=getnumberofrunningresources();
  writeln('Number of resources ', n);
End.
```

attach

Procedure: attach

Syntax 1: attach(string_id1, string_id2) : boolean

Description: Will attach two interfaces together based on the IDs.

Note: Two interface that can be attached will always be attached by this method regardless of their position. **Return value:** true on a successful attachment.

Syntax 2: attach(abstract_interface, string_target) : boolean

Description: Will try to attach two interfaces together by invoking the planner to find a solution on how to do the attachment. string_target is a name of a remote interface.

Return value: true on a successful attachment.

Example:

```
program Example;
begin
  attach('Interface1ID', 'Interface2ID');
End.
```

assembly

Procedure: assembly

Syntax: assembly(*abstract_interface*, *string_name*) : integer

Description: Will try to assemble *string_name* through and abstract interface *abstract_interface* by invoking a planner to find a way to do assembly.

Return value: process return value.

Example:

```
program Example;
abstract product : interface;

begin
    { Will search for Product2 and if compatible it will be assembled through product
    interface with us.
    Note, that after assembly Product2 will be marked FINISHED and removed. }
    assembly(product, 'Product2');
End.
```

Object access

getobject

Procedure: getobject

Syntax: getobject(*string_id*, *struct_var*) : boolean

Description: Will map the CMAS object identified by *string_id* and map that to the *struct_var*.

Return value: true if mapped.

Example:

```
{ An example program }
Program Example;

struct Entity;
begin
    var Name : String;
    var Id : String;
    var InstanceId : String;
    var Description : String;
end;

var anObject : Entity;

procedure onRun;
begin
    if not getobject('id', anObject) then
        error 'unable to map object';

    // Set a new run time name
    anObject.name:='New name';
    writeln(anObject);
end.
```

getrelation

Procedure: getrelation

Syntax: getrelation(*string_id*, *array_var*, *string_relation*) : boolean

Description: Will map a relation named string_relation in a CMAS object and associate that with the array array_var. The CMAS object is identified by string_id.

Return value: true if mapped.

Example:

```
{ An example program }
Program Example;

$include 'system:ontology.inc'

var items: array [ 1 .. 2 ] of Hazard;
var risk :integer;

procedure onRun;
begin
    // Get my own relation Hazards and map that to the items array
    getrelation(this.ID, items, 'Hazards');

    // Loop for all Hazards
    foreach i in items do
        risk:=risk+i.risk;

    writeln('Total risk as a sum = ', risk);
end.
```

Examples

Example 1 - Arrays

```
{ An example program how to handle arrays }
Program Example_array;

{Global variables }
var MyArray : array [1 .. 10] of integer;
var i : integer;

Procedure PrintArray(Size: integer);
var Counter : integer;
Begin
    for Counter:=1 to Size do
    begin
        writeln('Value ', Counter, '=', MyArray[Counter] );
    end;
End.

{ Main procedure }
Begin
    for i:=startindex(MyArray) to endindex(MyArray) do
    begin
        MyArray[i]:=11 - i;
    end; { End for }

    { Print array with a procedure }
    PrintArray(length(MyArray));

    { Print array with writeln }
    writeln(MyArray);
End.
```

â€ƒ

Example 2 - Interfaces and objects

```

{ An example program how to handle interfaces and agent objects }
Program Example_Skill;

{ Abstract must be global }
abstract Interface1 : interface;

{ Main procedure }
Begin
  { Set global agent variable }
  agent.running:=true;

  {Execute skill on remote interface }
  Interface1.Start:=true;
  Interface1.RemoteSkill(); // Call remote skill

End.

```

Example 3 - Structures

```

{ An example program how to use structures }
Program Pose_Example;

{ Define a new struct for a Pose in 3D }
struct Pose3D;
Begin
  Var x, y, z: real; // Location
  Var rx, ry, rz: real; // Rotation
End;

function Diff(a, b : Pose3D) : Pose3D;
begin
  Diff.x:= a.x-b.x;
  Diff.y:= a.y-b.y;
  Diff.z:= a.z-b.z;
end.

{ Declaration of Pose3D variables Pos1 and Pos2}
Var Pos1:=(x:=1.0, y:=2.0,z:=3.0, rx:=0.0, ry:=0.0, rz:=0.0): Pose3D; // Init. of x, y and
z in structure.
Var Pos2: Pose3D;

{ Main procedure }
Begin
  Pos2.x:=3.0;
  Pos2.y:=2.0;
  Pos2.z:=1.0;

  WriteLn('Pos diff=', Diff(Pos1, Pos2));
End.

```

Example 4 - MessageBox User Information

```

{ An example program how to show HMI messages }
Program MessageBox_Example;

$include 'system:messagebox.inc'

{ Main procedure }
Var Result : Integer;
Begin
  Result:=MessageBox('Press a button', 'Title', DIALOG_SHOW_YES_NO_CANCEL);

  if Result=DIALOG_ANSWER_YES then
  begin
    WriteLn('You pressed Yes');
  end;
  else if Result=DIALOG_ANSWER_NO then
  begin
    WriteLn('You pressed No');
  end;
End.

```

```

    end;
else if Result=DIALOG_ANSWER_CANCEL then
begin
    WriteLn('You pressed Cancel');
end;
else
begin
    WriteLn('You closed the window');
end;
End.

```

Example 5 - InputDialog

```

{ An example program how to get user input strings from HMI }
Program InputDialog_Example;

Var Result : String;
Var DefaultValue:='3' : String;
Var Iterator: Integer;

{ Main procedure }
Begin
    Result:=InputDialog('How many iterations?', DefaultValue);

    if not StringIsEmpty(Result) and StringIsInteger(Result) then
    begin
        Iterator:=StringToInt(Result);

        while Iterator>0 do
        begin
            WriteLn('Loop - ', Iterator);
            Iterator:=Iterator-1;
        end;
    end;
End.

```

Example 6 - OptionDialog

```

{ An example program how to create a HMI stop button for a calculation.
The program will calculate PI by a known series. The series is handled by a for loop
that will end if the user press the stop button or when the Counter reach MAX_COUNT.
}
Program ExampleOptionDialog;
Var Selection:=-1, Counter:=1 : Integer;
Var Odd:=false : Boolean;
Var PI:=1.0 : Real;
Var Constant MAX_COUNT:=10000000 :Integer;

{ Main procedure }
Begin
    { Create an OptionDialog with one stop button associated with handle 1 }
    OptionDialog(1, Selection, 'Calculating PI', 'Selection', 'Stop');

    While Selection=-1 AND Counter<MAX_COUNT do
    Begin
        if Odd then
            PI := PI + 1.0 / ( 2.0 * inttoreal(Counter) + 1.0);
        else
            PI := PI - 1.0 / ( 2.0 * inttoreal(Counter) + 1.0);

        Counter:=Counter+1;
        Odd:=not Odd;
    End;

    { Check if the OptionDialog is still open, if open then close it }
    if OptionDialog(1, true) = true then
    Begin
        OptionDialog(1, false);
    End;

```

```

    WriteLn('PI is ', PI*4.0, ' after ', Counter , ' number of iterations.');
End.
```

Example 7 - JSON Get

```

Program Example_JSON;
Var JSONString:='
{
  "array": [1, 2, 3],
  "string": "Hello World",
  "boolean": true,
  "real": 1.23,
  "integer": 123,
  "object": {
    "a": "b",
    "c": "d"
  }
}' : String;

Var a : String;
Var b : Real;
Var c : Integer;
Var d : Boolean;
Var e : array [1 .. 3] of String;
Var f : String;

Begin
  a:=getjelement(JSONString, 'string');
  WriteLn('a=', a);                                // Will print Hello World

  b:=stringtoreal(getjelement(JSONString, 'real'));
  writeln('b=', b);                               // Will print 1.23

  c:=stringtoint(getjelement(JSONString, 'integer'));
  writeln('c=',c);                               // Will print 123

  d:=stringtoboolan(getjelement(JSONString, 'boolean'));
  writeln('d=', d);                             // Will print true

  e:=getjelement(JSONString, 'array');
  writeln('e=', e[2]);                           // Will print 2

  f:=getjelement(JSONString, 'object');
  writeln('f=', f);                            // Will print {"a":"b","c":"d"}

  writeln('f.a=', getjelement(f, 'a')); // Will print b
End.
```

Example 8 - JSON and Struct mapping

```

Program Example_JSON_Struct;
Var JSONString:='
{
  "a": "Hello World",
  "b": true,
  "c": 1.23,
  "d": 123,
  "e": [1, 2, 3]
}' : String;

Struct JObj;
Begin
  Var a : String;
  Var b : Boolean;
  Var c : Real;
  Var d : Integer;
  Var e : array [1 .. 3] of Integer;
```

```

End;

Var Data : JObj;

Begin
  Data:=jtostruct(JSONString);      // Convert JSON to struct
  WriteLn(Data);                  // Print struct as a JSON

  Data.c:=3.21;                   // Change value in struct
  Data.d:=4;
  Data.b:=false;
  Data.e[1]:=3;

  JSONString:=structtoj(Data);    // Convert Struct back to JSON
  Writeln(JSONString);           // Print the canged JSON
End.

```

Example 9 - JSON Set

```

Program Example_JSON;

Var JSONString:='
{
  "array": [1, 2, 3],
  "string": "Hello World",
  "boolean": true,
  "real": 1.23,
  "integer": 123,
  "object": {
    "a": "b",
    "c": "d"
  }
}' : String;

Struct JObj;
Begin
  Var a : String;
  Var c : Real;
End;

Var a : Integer;
Var b : array [1 .. 3] of Integer;
Var c : JObj;

Begin
  { Get value for a from JSON object and add 1 }
  a:=StringToInt(GetJElement(JSONString, 'integer')) + 1;

  b[1]:=3;
  b[2]:=2;
  b[3]:=1;

  c.a:='HI!';
  c.c:=3.14;

  { Write new values to JSON object }
  JSONString:=SetJElement(JSONString, 'integer', a);
  JSONString:=SetJElement(JSONString, 'boolean', false);
  JSONString:=SetJElement(JSONString, 'object', c);
  JSONString:=SetJElement(JSONString, 'array', b);

  { Write JSON object to screen }
  Writeln(JSONString);
End.

```

Example 10 - Sort

```

{ An example program }
Program Example;

```

```

{ Array size }
Var Constant Size:=5 : Integer;

{ Data structure type }
Struct tData;
Begin
    Var Number:=0 : Integer;
    Var Index:=0 : Integer;
End;

{ Global data array }
Var data : Array [1 .. Size] of tData;

{ Procedure to initialize the data with random numbers }
Procedure Init;
    Var Counter : Integer;
Begin
    For Counter:=StartIndex(data) to EndIndex(data) do
        Begin
            Data[Counter].Number:=random(100);
            Data[Counter].Index:=Counter;
        End;
End.

{ Procedure to write all data to screen }
Procedure Display(Message : String);
    Var Counter : Integer;
Begin
    writeln();
    writeln(Message);

    For Counter:=StartIndex(data) to EndIndex(data) do
        Begin
            Write('Item ', Counter, ' ');
            Write('Number = ', Data[Counter].Number, ' ');
            Writeln('Index = ', Data[Counter].Index);
        End;
End.

{ Sorting algorithm }
Procedure BubbleSort;
Var Counter : Integer;
Var UnSorted:=true : Boolean;
Var SwapTmp : tData;
Begin

    { Iterate until sorted }
    While UnSorted=true do
        Begin
            { Iterate over the data array }
            UnSorted:=false;
            For Counter:=StartIndex(data)+1 to EndIndex(data) do
                Begin

                    { Check if unsorted data }
                    if data[Counter].Number < data[Counter-1].Number then
                        Begin
                            SwapTmp:=data[Counter];
                            data[Counter]:=data[Counter-1];
                            data[Counter-1]:=SwapTmp;
                            UnSorted:=true;
                        End;
                End;
            End;
        End;
End.

{ Main procedure }
Begin
    Init;
    Display('Random Data');
    BubbleSort;
    Display('Sorted Data');
End.

```

Example 11 - Foreach

```
{ An example program }
Program Example;

var anArray: array [ 1 .. 2, 1 .. 2] of integer;
Var aString:='hello world': string;

Struct Object;
Begin
    Var x, y : Real;
End;

var anObject:=(x:=1.0, y:=1.0):object;

procedure onRun;
begin
    { Will iterate over the array }
    foreach sub in anArray do
    begin
        foreach ii in sub do
            ii:=-1;
    end;
    writeln(anArray);

    { Will iterate over the string.
    Note that si is read only when it comes to strings
    }
    foreach si in aString do
    begin
        write(si);
    end;
    writeln();

    { Iterate over an object note that all fields in the
    object must match oi to be able to use foreach }
    foreach oi in anObject do
    begin
        oi:=1.1;
    end;
    writeln(anObject);
end.
```

System include files

CMAS provides a set of include files that can be used in any structured text program. These include files can be accessed using the **\$include** directive. To access system include files, the filename must begin with **system:**, e.g. **\$include 'system:ontology.inc'**.

Name	Description
ontology.inc	The most common CMAS objects
file.inc	Constants for file access
messagebox.inc	Constants for messageboxes

ontology.inc

```
//
// Include file for the ontology objects in CMAS
//

struct Entity;
begin
    var Name :          String;
    var Id :           String;
    var InstanceId :   String;
```

```

        var Description :      String;
end;

struct ProcessPlan extends Entity;
begin
    var Cost :              Integer;
    var TargetType :        String;
end;

struct Agent extends Entity;
begin
    var PhysicalID :        String;
    var AgentSpecificType:  String;
    var AutoDeploy :         Boolean;
end;

struct DeploymentAgent extends Agent;
begin
    var DeploymentTime:     String;
    var NumberOfItems :     Integer;
    var VariableTrigger:   String;
end;

struct Part extends Agent;
begin
end;

struct Resource extends Agent;
begin
end;

struct Material extends Agent;
begin
end;

struct InterfaceBase extends Entity;
begin
end;

struct InterfaceSpecialisation extends InterfaceBase;
begin
    var SpecialisationName :  String;
end;

struct Goal extends InterfaceBase;
begin
    var InterfaceName :      String;
    var Priority :           Integer;
    var ProcessPlanName :    String;
end;

struct Value extends Entity;
begin
    var UpperBound :         String;
    var LowerBound :         String;
    var Unit :               String;
end;

struct Variable extends Value;
begin
    var Source :             String;
    var Address :            String;
    var ReadOnly :           Boolean;
    var StoreLongTerm :      Boolean;
end;

struct VariableInteger extends Variable;
begin
    var Value :              Integer;
end;

struct VariableString extends Variable;
begin
    var Value :              String;
end;

struct VariableBoolean extends Variable;

```

```

begin
    var Value :          Boolean;
end;

struct VariableReal extends Variable;
begin
    var Value :          Real;
end;

struct VariableObject extends Variable;
begin
    var Size :           Integer;
end;

struct VariableArray extends Variable;
begin
    var LowerRange :     Integer;
    var UpperRange: Integer;
end;

struct Hazard extends Entity;
begin
    var Risk :           Integer;
    var Frequency :     Integer;
    var TargetType :    String;
end;

```

file.inc

```

// Include file for file handling

struct typeMODE;
begin
    var constant WRITE_TEXT:=1 : integer;
    var constant READ_TEXT:=2 : integer;
end;

var MODE:typeMODE;

```

messagebox.inc

```

// Include file for messagebox

// Return value constants for messagebox
Var CONSTANT
    DIALOG_ANSWER_YES:=0,
    DIALOG ANSWER NO:=1,
    DIALOG_ANSWER_CANCEL:=2,
    DIALOG_ANSWER_CLOSED:=-1 :Integer;

// Option constants for type of messagebox to show
Var CONSTANT
    DIALOG SHOW YES NO:=0,
    DIALOG_SHOW_YES_NO_CANCEL:=1,
    DIALOG_SHOW_OK_CANCEL:=2,
    DIALOG_ERROR_MESSAGE:=3,
    DIALOG_INFORMATION_MESSAGE:=4,
    DIALOG_WARNING_MESSAGE:=5,
    DIALOG QUESTION MESSAGE:=6,
    DIALOG_PLAIN_MESSAGE:=-1 :Integer;

```

Processplan action

A process plan will be executed according to the following steps:

State name	Description	Overload
Init	Constructor is called	optional
Negotiate	onNegotiate is called	onNegotiate is optional but if defined it must return an integer
Running	onRun is called	onRun must be defined and return an integer
Finished	onFinished is called	onFinished is optional

The function onRun() is a user-defined function that must always exist. The onRun function must be of type *integer* and the return values from the function must be one of the following:

Integer value	Predefined constant	Explanation
1	processplan.running	Still running, the onRun() function will be called again at the next cycle
2	processplan.done	Ongoing activities are done
3	processplan.failed	Indicates a failure and ongoing running will be aborted
4	processplan.interrupted	Indicates an interruption

Note

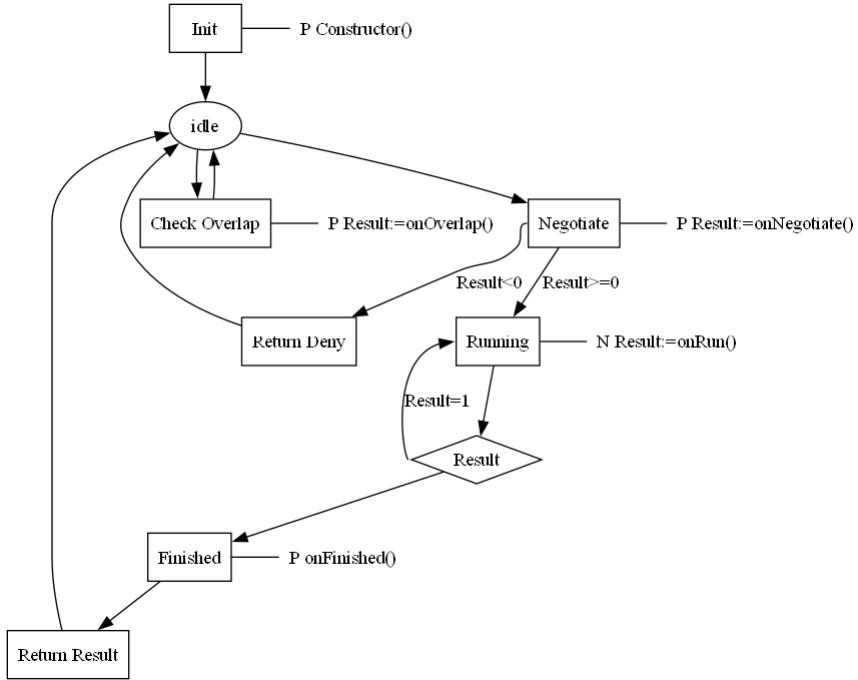
For a goal chart process plan it is only possible to describe the onRun part with the goal chart.

Note

It is not a good approach to block the onRun() function with an extensive loop or to wait for a specific value. This will generate a time out in the system and the process plan will be aborted. This can be turned off in preferences by setting timeout to 0 (use it for debug purposes only).

Note

onNegotiate is only called if another agent asks for a negotiation. It is possible to return -1 to state that the process plan is not compatible or >=0 to accept, a value >=0 indicates a cost to execute the skill. If onNegotiate is not defined the default behavior is to return the cost property for the process plan as it is.



In the following example a process plan is created to make two holes. It also demonstrate how to make loop where each iteration must return back the control to the system to avoid time out.

```

Program Processplan;
abstract aResource : interface;
var steps: integer;

function onNegotiate(Planning : boolean) : integer;
begin
  if remoteinterface.diameter=5.0 then
    onNegotiate:=this.cost; // Accept by using the value in the cost property
  else
    onNegotiate:=-1; // Refuse
end.

function onOverlap() : boolean;
begin
  onOverlap:=true;
end.

function onRun() : integer;
begin
  if steps=0 then
  begin
    aResource.x:=10.0;
    aResource.MakeAHole();
    onRun:=processplan.running;
    steps:=steps+1;
  end;
  else if steps=1 then
  begin
    aResource.x:=20.0;
    aResource.MakeAHole();
    onRun:=processplan.running;
    steps:=steps+1;
  end;
  else
  begin
    steps=0;
  end;
end;

```

```

        onRun:=processplan.done;
    end;
end.

// Constructor
begin
    steps:=0;
end.

```

Requirements

Introduction

Requirements can serve as criteria representing necessary conditions that must be met prior to the execution of a skill or a goal. For instance, if a robot possesses the skill "MoveRobot," an ATTACHED requirement can specify that a part must be affixed to the robot before it can start to move. CMAS has a built in AI planner that can autonomously generate plans outlining how to fulfil requirements and then execute them.

Example requirement on a skill

In the following example, a part is placed in a buffer. The part has only one goal to run a process. The process is located on a machine that requires the part to be there. In between the buffer and the machine, there's a robot that can do transportation. By modelling the system with a requirement and specialised interfaces CMAS can find a path for the part to reach the process without explicitly setting it as goals.

Model

The object illustration below demonstrates how the example can be modelled. The crucial elements in the model are the specialised interfaces paired with relevant skills and their associated PostActions, the AI planner will utilise these to find solutions.

- In this example we will use two templates for interface specialisation: InterfaceTransport - An interface that can do transportation
- InterfaceLocation - An interface that can be a destination or a start position

Buffer:Resource						
Name	Buffer					
Interfaces	1	Name	BufferInterface			
		SpecialisationName	Location			
		Skills	1	Name	pickBuffer	
				PostAction	DETACHED	
			2	Name	placeBuffer	
				PostAction	ATTACHED	
		Specialisation	Location	Name	Position	
				ValueType	Integer	
				Value	1	
				ReadOnly	true	

Machine:Resource						
Name	Machine					
Interfaces	1	Name	ProcessInterface			
		SpecialisationName	Location			
		Skills	1	Name	placeProcess	
				PostAction	ATTACHED	
			2	Name	runProcess	
				Requirements	1 Name MustBeAttached	
		Specialisation	Location	Name	Position	
				ValueType	Integer	
				Value	2	
				ReadOnly	true	

Robot:Resource						
Name	Robot					
Interfaces	1	Name	GripperInterface			
		SpecialisationName	Transport			
		Skills	1	Name	Transport	
				PostAction	TRANSFERRED	
		Specialisation	1	FromLocation	Name Position	
				ValueType	Integer	
				Value	Object	
			2	ToLocation	Name Position	
				ValueType	Integer	
				Value	Object	
			3	Step	Name Position	
				ValueType	Integer	

ProductA:Part						
Name	ProductA					
Goals	1	Name	GoalGetProcessed			
		ProcessPlan	runProcess			
		Interface	ProcessInterface			
Interfaces	1	Name	BufferInterface			
	2	Name	GripperInterface			

Step-by-step instructions:

1. Buffer

- 1.1 Create a resource and name it Buffer. The buffer will serve as a source for our parts later on.
- 1.1. On Buffer create a new InterfaceLocation and name it BufferInterface. An InterfaceLocation is a specialized interface that can serve as a location, e.g. for pick or place.
- 1.2. On BufferInterface there is a folder called Specialisation that has one item called Location. Location is a variable object. Open the location folder and add an VariableInteger to items. Name the new integer Position, and set the value to 1 and mark ReadOnly. This variable is now our location it is possible to add more variables as x, y and z to represent more complex locations if required.
- 1.3. Now create a new Skill as Structured Text and name it pickBuffer.
- 1.4. Select the new skill and change PostAction to DETACHED. This indicates that if you run this skill you will be detached.
- 1.5. Create another Structured Text skill and name it placeBuffer.
- 1.6. Select the new skill and change PostAction to ATTACHED.
-

2. Machine

- 2.1 Create a resource and name it Machine.
- 2.2. On Machine create a new InterfaceLocation and name it ProcessInterface.
- 2.3 In the same way as for BufferInterface add a Position variable and set the value to 2 and read only.
- 2.4 Create a Structure Text skill, name it placeProcess and set PostAction to ATTACHED.
- 2.5 Create a Structure Text skill and name it runProcess.
- 2.6 Open the folder for runProcess and add a new Requirement. Name it MustBeAttached.
- 2.7 Select MustBeAttached and change WorkflowType to ATTACHED. This requirement will indicate that a Part that wants to run the skill must be attached to the interface first.
- 2.8 Open Actions for placeProcess skill and add the following line to onRun() function:

```
writeln('Running the process');
```

3. Robot

- 3.1 Create a resource and name it Robot.
- 3.2 Create a new InterfaceTransport and name it GripperInterface. This will serve as a interface that can do transports.
- 3.3 Open the Specialisation folder.
- 3.4 On FromLocation add a new IntegerVariable named Position. Do not make it read-only.
- 3.5 On ToLocation add a new IntegerVariable named Position. Do not make it read-only.
- 3.6 Double-click on the existing skill Transport to edit.
- 3.7 In side the onRun() function add

```
writeln('Robot transport from ',
      remoteinterface.Specialisation.FromLocation.Position,
      ' to ',
      remoteinterface.Specialisation.ToLocation.Position,
      ' Step ', remoteinterface.Specialisation.Step);
```
-

4. ProductA

- 4.1 Add a new Part
- 4.2 Double-click on the new part.
- 4.3 Right click in the top left window and select Add new goal.
- 4.4 Double-click on the new goal.
- 4.5 Set interface name to ProcessInterface.
- 4.6 Set process plan name for the goal to runProcess.
- 4.7 Close the goal.
- 4.8 Drag the goal and drop it on Start.
- 4.9 Close the editor.
- 4.10 Add a new interface and name it BuffertInterface. This ensures that the part can be placed in a buffer.
- 4.11 Add one more interface and name it GripperInterface. This ensures that the part can be placed in a gripper.
- 4.12 Select the part and click on Actions.
- 4.13 Inside the empty editor write the following program (this program will put the part inside a buffer when deployed):

```
program Part;
abstract aBuffer : interface;

procedure onDeploy();
begin
  aBuffer.placeBuffer();
end.
```

-
-

Run

1. Press Run to start the system (if auto deploy is not on deploy everything).
You should see the following output:

```
Robot transport from 1 to 2 Step 1  
Robot transport from 1 to 2 Step 2  
Running the process
```

Example requirements in a process plan

In this example we will look at requirements in process plans for *Parts* .

Assembly

Similar to how we specified ATTACHED as a requirement in the previous example, we can also specify ASSEMBLY as a requirement. While ASSEMBLY is closely related to ATTACHED, there is a key difference: when an agent, typically a material agent, is assembled with another agent, the assembled agent is marked as finished and can no longer be used. The modelling pattern involves representing the main structure as a *Part* agent and all subassemblies as *Material* agents. This allows the planner to generate solutions for how to *assemble materials and attach parts* .

The material to be assembled must have a specialized interface *InterfaceAssembly* . Similarly, the main structure must have a specialized interface *InterfaceLocation* , that is compatible with the assembly interface, to tell where to assembly the material.

InterfaceSpecialisation

An InterfaceSpecialisation resembles a regular Interface but includes an additional relation called Specialisation, which can incorporate variables. It also have a property called SpecialisationName that indicates the purpose of the interface. Unlike the standard interface relation Variables, the Specialisation relation is disregarded during the negotiation process and in determining the compatibility of two interfaces. The specialisation variables are utilised by the internal AI planner to gather information for plans or can be used in applications to introduce variables that don't interfere with the negotiation process.

InterfaceLocation

InterfaceLocation is a template for creating InterfaceSpecialisation for locations. It automatically creates a InterfaceSpecialisation with a VariableObject named Location and SpecialisationName set to Location. It also assigns SpecialisationName to a read only value "Location".

InterfaceTransport

InterfaceTransport is a template for creating an InterfaceSpecialisation tailored for transport. It automatically generates an InterfaceSpecialisation that includes two VariableObjects named FromLocation and ToLocation. Additionally, it creates a Skill named Transport with the PostAction configured as TRANSFERED and assigns SpecialisationName to a read only value "Transport".

InterfaceAssembly

InterfaceAsembly is a template for creating an Interface that can be used for assembly. It automatically creates a InterfaceSpecialisation with the SpecialisationName set to Assembly.

Note

It is possible to manually create a specialisation interface by setting the correct values and adding the VariableObjects manually on an *InterfaceSpecialisation*.

Hierarchical Index

Class Hierarchy

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

cmas.ontology.interfaces.AbstractInterface.AbstractType	130
cmas.ontology.agentcommunication.ACL.ACMessage	131
cmas.ontology.action.Action	132
cmas.ontology.action.ActionInterface	133
cmas.ontology.agents.AgentAction	142
cmas.ontology.agents.Component	156
cmas.ontology.agents.Material	235
cmas.ontology.agents.Part	253
cmas.ontology.agents.DeploymentAgent	172
cmas.ontology.agents.Resource	296
cmas.ontology.processplan.ProcessPlanJava	270
cmas.ontology.processplan.ProcessPlanStructuredText	276
cmas.ontology.action.ActionJava	134
cmas.ontology.adapter.Adapter	135
cmas.ontology.adapter.FILE.FileClient	191
cmas.ontology.adapter.Modbus.ModbusClient	242
cmas.ontology.adapter.OPCUA.OPCUAClientBase	250
cmas.ontology.adapter.OPCUA.OPCUAClient	247
cmas.ontology.adapter.REST.RESTClient	302
cmas.ontology.adapter.ZeroMQ.ZMQClient	390
cmas.ontology.adapter.socket.SocketTCPClient	311
cmas.ontology.adapter.socket.SocketUDPClient	313
cmas.ontology.adapter.websocket.WebSocketClient	379
cmas.ontology.adapter.AdapterFactory	136
cmas.ontology.adapter.Modbus.ModbusClient.ADDRESS_TYPE	137
cmas.ontology.agents.Agent.AgentBaseTypes	147
AgentRunnable	
cmas.ontology.agents.AgentAction	142
cmas.ontology.interfaces.InterfaceBase	213
cmas.ontology.interfaces.Goal	193
cmas.ontology.interfaces.Interface	203
cmas.ontology.interfaces.InterfaceSpecialisation	223
cmas.ontology.interfaces.InterfaceAssembly	208
cmas.ontology.interfaces.InterfaceLocation	218
cmas.ontology.interfaces.InterfaceTransport	229

cmas.ontology.agents.AgentAction.AgentState	148
cmas.ontology.interfaces.Attachment.....	149
cmas.ontology.batch.BatchItem	154
cmas.ontology.agentcommunication.CommunicativeActs	155
cmas.ontology.agents.AgentAction.....	142
cmas.ontology.interfaces.AbstractInterface	128
cmas.ontology.interfaces.InterfaceBase	213
cmas.ontology.processplan.ProcessPlan	259
cmas.ontology.processplan.ProcessPlanGoals	264
cmas.ontology.processplan.ProcessPlanJava	270
cmas.ontology.processplan.ProcessPlanStructuredText	276
cmas.ontology.agentcommunication.jsoncontent.Content	162
cmas.ontology.agentcommunication.jsoncontent.template.ContentBusy	163
cmas.ontology.agentcommunication.jsoncontent.template.ContentDone	164
cmas.ontology.agentcommunication.jsoncontent.template.ContentError	165
cmas.ontology.agentcommunication.jsoncontent.template.ContentInformation.....	166
cmas.ontology.agentcommunication.jsoncontent.template.ContentNegotiate	167
cmas.ontology.agentcommunication.jsoncontent.template.ContentProgress	169
cmas.ontology.agentcommunication.jsoncontent.template.ContentReplan	170
cmas.ontology.agentcommunication.jsoncontent.template.ContentResult	171
cmas.ontology.agentcommunication.jsoncontent.ContentParameter	168
cmas.ontology.agents.DeploymentAgent.DeploymentCriteria	177
cmas.ontology.Entity.EntityType	182
Event	
cmas.ontology.Container.....	161
cmas.ontology.Visualisation	378
cmas.ontology.Entity	178
cmas.ontology.Executable	186
cmas.ontology.Hazard	198
cmas.ontology.Reference.....	286
cmas.ontology.agents.Agent	138
cmas.ontology.agents.AgentAction	142
cmas.ontology.batch.Batch	150
cmas.ontology.interfaces.InterfaceBase.....	213
cmas.ontology.processplan.ProcessPlan	259
cmas.ontology.processplan.SkillCall	307
cmas.ontology.variables.Variable	331
cmas.ontology.variables.Value	323
cmas.ontology.variables.VariableBoolean	323
cmas.ontology.variables.VariableInteger	347
cmas.ontology.variables.VariableProperty	360
cmas.ontology.variables.VariableReal	366
cmas.ontology.variables.VariableString	372
cmas.ontology.variables.VariableArray	335
cmas.ontology.variables.VariableObject	354
cmas.ontology.workflow.WorkFlow	381

cmas.ontology.workflow.Requirement.....	292
cmas.ontology.workflow.WorkFlowLog.....	385
cmas.ontology.Property< T >.....	283
cmas.ontology.agentcommunication.ACL.MessageHandler	240
EventListener	
cmas.ontology.Relations.RelationObject	290
events.Event.EventListener	
cmas.ontology.Reference	286
ExceptionBase	
cmas.ontology.exceptions.ExceptionConstant	183
cmas.ontology.exceptions.ExceptionType	184
cmas.ontology.exceptions.ExceptionVariable.....	185
cmas.ontology.Executable.ExecutableTypes	190
cmas.ontology.interfaces.InterfaceBase.InterfaceState	228
cmas.ontology.variables.Iterator	234
cmas.ontology.adapter.Modbus.ModbusClient.MODBUS_TYPE	241
cmas.ontology.Ontology	244
cmas.ontology.Ontology.OntologyObject.....	245
cmas.ontology.Ontology.OntologyRelation	246
cmas.ontology.adapter.OPCUA.OPCUANode	252
cmas.ontology.agentcommunication.ACL.ACMessage.Performative.....	258
cmas.ontology.processplan.ProcessPlanInterface	269
cmas.ontology.processplan.ProcessPlan	259
cmas.ontology.processplan.ProcessPlan.ProcessPlanReturnValue	275
cmas.ontology.processplan.ProcessPlan.ProcessPlanType	281
cmas.ontology.Properties	282
cmas.ontology.variables.ValueRange.Range	285
cmas.ontology.Relations	291
cmas.ontology.adapter.REST.RESTClient.REST_TYPE	301
cmas.ontology.adapter.socket.SocketTCPClient.SERVER_TYPE.....	305
cmas.ontology.adapter.socket.SocketUDPClient.SERVER_TYPE	306
cmas.ontology.Property< T >.SpecificPropertyType	315
cmas.ontology.workflow.WorkFlow.SpecificTypes	316
cmas.ontology.variables.Variable.TypeOfVariable	317
cmas.ontology.Types.....	318
cmas.ontology.Entity.UID.....	322
URLClassLoader	
cmas.ontology.action.IndependentClassLoader	202
cmas.ontology.variables.ValueRange	329

cmas.ontology.variables.Value.ValueTypes	330
cmas.ontology.interfaces.InterfaceBase.VariableMatchingType	353
cmas.ontology.workflow.WorkFlow.WorkFlowTypes	389

Class Index

Class List

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

<code>cmas.ontology.interfaces.AbstractInterface</code>	128
<code>cmas.ontology.interfaces.AbstractInterface.AbstractType</code>	130
<code>cmas.ontology.agentcommunication.ACMessage</code>	131
<code>cmas.ontology.action.Action</code>	132
<code>cmas.ontology.action.ActionEvent</code>	133
<code>cmas.ontology.action.ActionJava</code>	134
<code>cmas.ontology.adapter.Adapter</code>	135
<code>cmas.ontology.adapter.AdapterFactory</code>	136
<code>cmas.ontology.adapter.Modbus.ModbusClient.ADDRESS_TYPE</code>	137
<code>cmas.ontology.agents.Agent</code>	138
<code>cmas.ontology.agents.AgentAction</code>	142
<code>cmas.ontology.agents.Agent.AgentBaseTypes</code>	147
<code>cmas.ontology.agents.AgentAction.AgentState</code>	148
<code>cmas.ontology.interfaces.Attachment</code>	149
<code>cmas.ontology.batch.Batch</code>	150
<code>cmas.ontology.batch.BatchItem</code>	154
<code>cmas.ontology.agentcommunication.CommunicativeActs</code>	155
<code>cmas.ontology.agents.Component</code>	156
<code>cmas.ontology.Container</code>	161
<code>cmas.ontology.agentcommunication.jsoncontent.Content</code>	162
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentBusy</code>	163
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentDone</code>	164
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentError</code>	165
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentInformation</code>	166
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentNegotiate</code>	167
<code>cmas.ontology.agentcommunication.jsoncontent.ContentParameter</code>	168
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentProgress</code>	169
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentReplan</code>	170
<code>cmas.ontology.agentcommunication.jsoncontent.template.ContentResult</code>	171
<code>cmas.ontology.agents.DeploymentAgent</code>	172
<code>cmas.ontology.agents.DeploymentAgent.DeploymentCriteria</code>	177
<code>cmas.ontology.Entity</code>	178
<code>cmas.ontology.Entity.EntityType</code>	182
<code>cmas.ontology.exceptions.ExceptionConstant</code>	183
<code>cmas.ontology.exceptions.ExceptionType</code>	184
<code>cmas.ontology.exceptions.ExceptionVariable</code>	185
<code>cmas.ontology.Executable</code>	186
<code>cmas.ontology.Executable.ExecutableTypes</code>	190
<code>cmas.ontology.adapter.FILE.FileClient</code>	191
<code>cmas.ontology.interfaces.Goal</code>	193
<code>cmas.ontology.Hazard</code>	198
<code>cmas.ontology.action.IndependentClassLoader</code>	202
<code>cmas.ontology.interfaces.Interface</code>	203

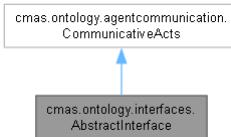
cmas.ontology.interfaces.InterfaceAssembly	208
cmas.ontology.interfaces.InterfaceBase	213
cmas.ontology.interfaces.InterfaceLocation	218
cmas.ontology.interfaces.InterfaceSpecialisation	223
cmas.ontology.interfaces.InterfaceBase.InterfaceState	228
cmas.ontology.interfaces.InterfaceTransport	229
cmas.ontology.variables.Iterator	234
cmas.ontology.agents.Material	235
cmas.ontology.agentcommunication.ACMessageHandler	240
cmas.ontology.adapter.Modbus.ModbusClient.MODBUS_TYPE	241
cmas.ontology.adapter.Modbus.ModbusClient	242
cmas.ontology.Ontology	244
cmas.ontology.Ontology.OntologyObject	245
cmas.ontology.Ontology.OntologyRelation	246
cmas.ontology.adapter.OPCUA.OPCUAClient	247
cmas.ontology.adapter.OPCUA.OPCUAClientBase	250
cmas.ontology.adapter.OPCUA.OPCUANode	252
cmas.ontology.agents.Part	253
cmas.ontology.agentcommunication.ACMessage.Performative	258
cmas.ontology.processplan.ProcessPlan	259
cmas.ontology.processplan.ProcessPlanGoals	264
cmas.ontology.processplan.ProcessPlanInterface	269
cmas.ontology.processplan.ProcessPlanJava	270
cmas.ontology.processplan.ProcessPlanReturnValue	275
cmas.ontology.processplan.ProcessPlanStructuredText	276
cmas.ontology.processplan.ProcessPlan.ProcessPlanType	281
cmas.ontology.Properties	282
cmas.ontology.Property< T >	283
cmas.ontology.variables.ValueRange.Range	285
cmas.ontology.Reference	286
cmas.ontology.Relations.RelationObject	290
cmas.ontology.Relations	291
cmas.ontology.workflow.Requirement	292
cmas.ontology.agents.Resource	296
cmas.ontology.adapter.REST.RESTClient.REST_TYPE	301
cmas.ontology.adapter.REST.RESTClient	302
cmas.ontology.adapter.socket.SocketTCPClient.SERVER_TYPE	305
cmas.ontology.adapter.socket.SocketUDPClient.SERVER_TYPE	306
cmas.ontology.processplan.SkillCall	307
cmas.ontology.adapter.socket.SocketTCPClient	311
cmas.ontology.adapter.socket.SocketUDPClient	313
cmas.ontology.Property< T >.Specific.PropertyType	315
cmas.ontology.workflow.WorkFlow.SpecificTypes	316
cmas.ontology.variables.Variable.TypeOfVariable	317
cmas.ontology.Types	318
cmas.ontology.Entity.UID	322
cmas.ontology.variables.Value	323
cmas.ontology.variables.ValueRange	329
cmas.ontology.variables.Value.ValueTypes	330

cmas.ontology.variables.Variable	331
cmas.ontology.variables.VariableArray	335
cmas.ontology.variables.VariableBoolean	341
cmas.ontology.variables.VariableInteger	347
cmas.ontology.interfaces.InterfaceBase.VariableMatchingType	353
cmas.ontology.variables.VariableObject	354
cmas.ontology.variables.VariableProperty	360
cmas.ontology.variables.VariableReal	366
cmas.ontology.variables.VariableString	372
cmas.ontology.Visualisation	378
cmas.ontology.adapter.websocket.WebSocketClient	379
cmas.ontology.workflow.WorkFlow	381
cmas.ontology.workflow.WorkFlowLog	385
cmas.ontology.workflow.WorkFlow.WorkFlowTypes	389
cmas.ontology.adapter.ZeroMQ.ZMQClient	390

Class Documentation

cmas.ontology.interfaces.AbstractInterface Class Reference

Inheritance diagram for cmas.ontology.interfaces.AbstractInterface:



Classes

enum AbstractTypePublic Member Functions

- **AbstractInterface** (String AbstractName)
- **AbstractInterface** (String AbstractName, String Interface, **AbstractType** type)
- **AbstractInterface** (String AbstractName, **VariableObject** Object)
- ArrayList< String > **getAllSkills** ()
- String **toString** ()
- void **addSkill** (String Name)
- void **addSkills** (ArrayList< String > Names)
- String **getName** ()
- void **setName** (String newName)
- void **addVariable** (String Name)
- void **addVariables** (ArrayList< String > variables)
- void **setRemoteInterface** (Pair< Agent, InterfaceBase > bestInterface)
- Pair< Agent, InterfaceBase > **getRemoteInterface** ()
- Pair< Agent, InterfaceBase > **getLocalInterface** ()
- void **setLocalInterface** (Pair< Agent, InterfaceBase > localInterface)
- int **getCost** ()
- boolean **isStatic** ()
- boolean **isDefined** ()
- void **setIsDefined** ()
- boolean **isLocal** ()
- void **setReference** (AbstractInterface refInterface)
- InterfaceBase **getDefinedInterface** ()
- void **setDefinedInterface** (InterfaceBase localInterface)
- **ProcessPlanReturnValue agree** (boolean book) throws ExceptionRuntime, ExceptionFailedToExecuteStatement
- **ProcessPlanReturnValue negotiate** (boolean book) throws ExceptionRuntime, ExceptionFailedToExecuteStatement
- ArrayList< Requirement > **getRequirement** ()
- void **unbookOverlapInterfaces** ()
- HashSet< String > **getInterfaceTargetList** (InterfaceBase remoteInterface, ArrayList< String > skillList, boolean CheckOnSkillsOnly)
- synchronized **ProcessPlanReturnValue skillCall** (String functionName, **VariableInteger** result) throws InterruptedException, ExceptionRuntime, ExceptionNullValue, **ExceptionType**, **ExceptionConstant**
- synchronized **ProcessPlanReturnValue skillCall** (String functionName) throws InterruptedException, ExceptionRuntime
- synchronized **ProcessPlanReturnValue actionCall** (String functionName) throws InterruptedException, ExceptionRuntime
- Entity **getCommunicative** ()
- void **postMessage** (ACLMessages message)
- void **setReplyToInterface** (Agent agent, InterfaceBase replyInterface)
- boolean **getThrowOnError** ()

- `void setThrowOnError (boolean throwOnError)`
 - `AgentAction getInitAgent ()`
 - `void setInitAgent (AgentAction initAgent)`
 - `InterfaceBase getInitGoal ()`
 - `void setInitGoal (InterfaceBase goal)`
 - `boolean isRunning ()`
 - `boolean isNegotiated ()`
 - `void setNegotiated (boolean value)`
 - `void copyVariable (Block globalBlock)`
 - `boolean getIgnoreBusy ()`
 - `void setIgnoreBusy (boolean ignoreBusy)`
 - `boolean getIgnoreBooked ()`
 - `void setIgnoreBooked (boolean ignoreBooked)`
 - `ArrayList< Pair< Agent, InterfaceBase > > getCompatibleInterfaceList (DataTree root, Agent agent)`
 - `MessageHandler getMessageHandler ()`
 - `ImageIcon getEntityIcon ()`
 - `ProcessPlanReturnValue fulfilLocalRequirements (Requirement requirement) throws ExceptionFailedToExecuteStatement, ExceptionRuntime`
 - default `ACLMessages waitForPerformative (MessageHandler messageHandler, ACLMessages message, Performative answerType)` throws `InterruptedException`
-

Member Function Documentation

Entity `cmas.ontology.interfaces.AbstractInterface.getCommunicative ()`

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

MessageHandler `cmas.ontology.interfaces.AbstractInterface.getMessageHandler ()`

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

void `cmas.ontology.interfaces.AbstractInterface.postMessage (ACLMessages message)`

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

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

- `AbstractInterface.java`

cmas.ontology.interfaces.AbstractInterface.AbstractType Enum Reference

Public Attributes

- DYNAMIC
- STATIC
- LOCAL
- DEFINED

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

- AbstractInterface.java

cmas.ontology.agentcommunication.ACL.ACMessage Class Reference

Classes

enum PerformativePublic Member Functions

- **ACMessage (Performative performative, CommunicativeActs sender, Content content)**
- **ACMessage (ACMessage message)**
- **ACMessage copy ()**
- **Performative getPerformative ()**
- **String getConversationID ()**
- **void setConversationID (String conversationID)**
- **String getOntology ()**
- **void setOntology (String ontology)**
- **Content getContent ()**
- **void setContent (String content)**
- **CommunicativeActs getSender ()**
- **StringProperty senderProperty ()**
- **void setSender (CommunicativeActs sender)**
- **CommunicativeActs getReceiver ()**
- **StringProperty receiverProperty ()**
- **void setReceiver (CommunicativeActs receiver)**
- **String getLanguage ()**
- **void setLanguage (String language)**
- **CommunicativeActs getReplyTo ()**
- **StringProperty replyToProperty ()**
- **void setReplyTo (CommunicativeActs replyTo)**
- **void setPerformative (Performative performative)**
- **String getTimestampSend ()**
- **void setTimestampSend (String timestampSend)**
- **String getTimestampReceived ()**
- **void setTimestampReceived (String timestampReceived)**
- **CommunicativeActs getDelegateTo ()**
- **void setDelegateTo (CommunicativeActs delegate)**
- **boolean isAnswer ()**

Static Public Member Functions

- **static StringProperty getFullName (CommunicativeActs item)**

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

- **ACMessage.java**

cmas.ontology.action.Action Class Reference

Public Member Functions

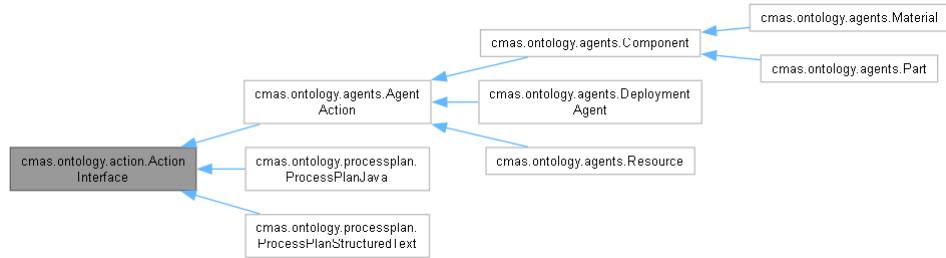
- **Action (Entity Owner)**
- **String getProgram ()**
- **void setProgram (String actions)**
- **Parser getCompiledProgram ()**
- **void compile () throws ExceptionParameters, Exception**
- **void compile (Agent agent, Entity target, Variable ... variables) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **void compile (Agent agent, Entity target, ArrayList<Variable> variables, ArrayList<AbstractInterface> abstractInterfaces) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **boolean isCompiled ()**
- **void run () throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **void run (Agent agent, Entity target, Variable... variables) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **void replaceVariableObject (VariableObject variable)**
- **void addVariable (Variable variable)**
- **void closeDialogs ()**
- **void closeFiles ()**
- **DebugResult BreakPointCallBack (Block globalBlock, Block localBlock, Statement statement, BreakPoint breakpoint)**
- **Object callFunction (String Name) throws ExceptionParameters, Exception**
- **Object callFunction (String Name, Agent agent, Entity target, Vector<Expression> param, Variable... variables) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **boolean callProcedure (String Name) throws ExceptionParameters, Exception**
- **boolean callProcedure (String Name, Agent agent, Entity target, Variable... variables) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **boolean callProcedureForce (String Name, Agent agent, Entity target) throws ExceptionParameters, Exception**
- **void copyParam (ContentParameter[] param, InterfaceBase localInterface)**
- **VariableObject createRemoteInterfaceVariable (InterfaceBase remoteInterface) throws ExceptionVariable, ExceptionType**
- **VariableObject createRemoteInterfaceVariable (ContentParameter[] param, InterfaceBase localInterface) throws ExceptionBase, ExceptionRuntime**

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

- Action.java

cmas.ontology.action.ActionInterface Interface Reference

Inheritance diagram for cmas.ontology.action.ActionInterface:



Public Member Functions

- `void gotoLine (Integer line)`
- `void stopEditor ()`

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

- `ActionInterface.java`

cmas.ontology.action.ActionJava Class Reference

Public Member Functions

- **ActionJava** (String fileName, String dependencies)
- **ProcessPlanInterface getObject ()** throws Exception
- boolean **compile ()** throws ExceptionParser
- String **getFileName ()**
- void **setFileName** (String fileName)
- String **getDependencies ()**
- void **setDependencies** (String dependencies)

Package Attributes

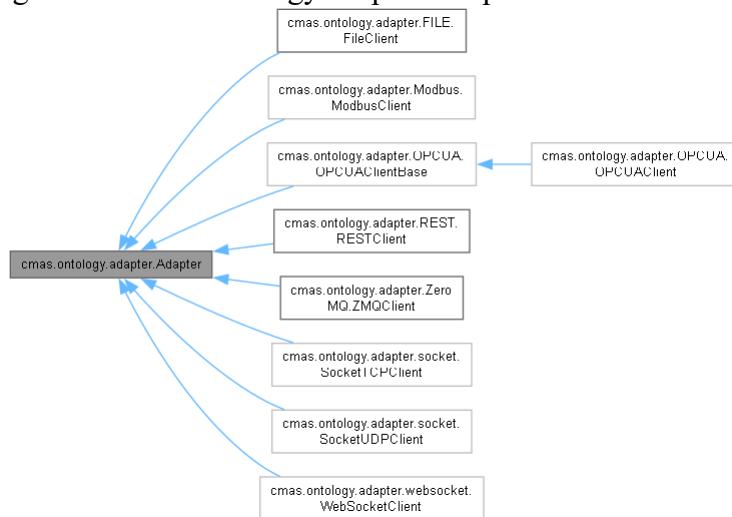
- ProcessPlanInterface **actionObject** = null

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

- ActionJava.java

cmas.ontology.adapter.Adapter Interface Reference

Inheritance diagram for cmas.ontology.adapter.Adapter:



Public Member Functions

- abstract boolean **close** ()
- abstract String **ReadData** (String Address, String Name, Object **Value**) throws Exception
- abstract void **WriteData** (String Address, String Name, Boolean bValue) throws Exception
- abstract void **WriteData** (String Address, String Name, String sValue) throws Exception
- abstract void **WriteData** (String Address, String Name, Integer iValue) throws Exception
- abstract void **WriteData** (String Address, String Name, Double dValue) throws Exception
- abstract void **setDebug** (boolean bValue)
- abstract String **getDefaultValue** ()
- abstract String **getDescription** ()
- abstract boolean **isValid** (String Source)
- abstract **Adapter** **createNew** (String Source)

Member Function Documentation

abstract void cmas.ontology.adapter.Adapter.setDebug (boolean bValue) [abstract]

Implemented in **cmas.ontology.adapter.OPCUA.OPCUAClientBase** (*p.251*).

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

- Adapter.java

cmas.ontology.adapter.AdapterFactory Class Reference

Static Public Member Functions

- static Adapter **getClient** (String Source)
- static String **getType** (String Source)
- static ArrayList< Pair< String, Adapter >> **getAllAdapters** ()
- static void **closeClients** ()

Package Attributes

- OPCUAClient **opcuaClients**

Static Package Attributes

- static ArrayList< Pair< String, Adapter >> **allAdapterInstances** = new ArrayList<Pair<String, Adapter>>()
- static ArrayList< Pair< String, Adapter >> **adapterObjects**

Member Data Documentation

ArrayList<Pair<String, Adapter> >
cmas.ontology.adapter.AdapterFactory.adapterObjects [static], [package]

Initial value:

```
= new ArrayList<>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Pair<>("OPCUA", new OPCUAClient()));
        add(new Pair<>("REST", new RESTClient()));
        add(new Pair<>("FILE", new FileClient()));
        add(new Pair<>("MODBUS", new ModbusClient()));
        add(new Pair<>("SOCKET TCP", new SocketTCPClient()));
        add(new Pair<>("SOCKET UDP", new SocketUDPClient()));
        add(new Pair<>("ZMQ", new ZMQClient()));
        add(new Pair<>("WebSocket", new WebSocketClient()));
    }
}
```

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

- AdapterFactory.java

cmas.ontology.adapter.Modbus.ModbusClient.ADDRESS_TYPE Enum Reference

Public Attributes

- **COIL**
- **DISCRETE_INPUT**
- **INPUT_REGISTER**
- **HOLDING_REGISTER**

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

- ModbusClient.java

cmas.ontology.agents.Agent Class Reference

Inheritance diagram for cmas.ontology.agents.Agent:



Classes

enum AgentBaseTypesPublic Member Functions

- **Agent (AgentBaseTypes basetype)**
- **Agent (Agent entity, boolean instance)**
- **String getPhysicalID ()**
- **void setPhysicalID (String physicalID)**
- **String getAgentSpecificType ()**
- **void setAgentSpecificType (String agentSpecificType)**
- **AgentBaseTypes getAgentBaseType ()**
- **void addAbstractInterface (AbstractInterface abstractInterface, String Name)**
- **void clearAbstractInterfaces ()**
- **AbstractInterface getAbstractInterface (String Name)**
- **RuntimeTree getRuntimeTree ()**
- **void setRuntimeTree (RuntimeTree runtimeTree)**
- **Domain getRuntimeDomain ()**
- **void setRuntimeDomain (Domain runtimeDomain)**
- **void DeSerialize (JsonObject job) throws ExceptionType**
- **JsonObject Serialize ()**
- **LinkedBlockingQueue< WorkFlowLog > getHistory ()**
- **void setHistory (LinkedBlockingQueue< WorkFlowLog > p)**
- **void addToHistory (WorkFlowLog log)**
- **Boolean getAutoDeploy ()**
- **void setAutoDeploy (Boolean autoDeploy)**
- **abstract Entity copy (boolean instance)**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRealtions ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**

- `ArrayList< Entity > getSpecificRelationType (String Name)`
- `void addSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (Entity ent)`
- `void eventReflection (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)`
- `abstract String getEntityDescription ()`
- `abstract String getEntityIconFile ()`
- `ImageView getEntityIcon ()`
- `void copyProperties (Container container, boolean instance)`
- `void setOverride (Boolean value)`
- `Boolean getProtected ()`
- `void setProtected (Boolean protectedValue)`
- `Object getUserData ()`
- `void setUserData (Object userData)`
- `String getClass_Name ()`
- `void setClassName (String newName)`
- `Relations getRealtionsObject ()`
- `ArrayList< BreakPoint > getBreakPointList ()`
- `void setBreakPointList (ArrayList< BreakPoint > breakPointList)`
- `Boolean isInstance ()`

Static Public Member Functions

- `static Entity Factory (JsonObject job) throws ExceptionType`
- `static AgentBaseTypes getAgentType (JsonObject job) throws ExceptionType`
- `static Entity Factory (String job) throws ExceptionType, ExceptionConstant, ExceptionVariable`

Protected Attributes

- `Properties properties`
- `Relations relations`

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract], [inherited]

Must be override by all classes extending `Entity` if instance is true `Properties` will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<code>instance</code>	if true a runtime entity is create otherwise a modelling entity		
Reimplemented in <code>cmas.ontology.agents.DeploymentAgent</code>		(p.174),	
<code>cmas.ontology.agents.Material</code>	(p.237),	<code>cmas.ontology.agents.Part</code>	(p.255),
<code>cmas.ontology.agents.Resource</code>	(p.298),	<code>cmas.ontology.batch.Batch</code>	(p.151),
<code>cmas.ontology.Executable</code>	(p.187),	<code>cmas.ontology.Hazard</code>	(p.199),
<code>cmas.ontology.interfaces.Goal</code>	(p.195),	<code>cmas.ontology.interfaces.Interface</code>	(p.204),
<code>cmas.ontology.interfaces.InterfaceAssembly</code>			(p.209),
<code>cmas.ontology.interfaces.InterfaceLocation</code>			(p.219),
<code>cmas.ontology.interfaces.InterfaceSpecialisation</code>			(p.224),
<code>cmas.ontology.interfaces.InterfaceTransport</code>			(p.230),
<code>cmas.ontology.processplan.ProcessPlanGoals</code>			(p.266),
<code>cmas.ontology.processplan.ProcessPlanJava</code>			(p.272),
<code>cmas.ontology.processplan.ProcessPlanStructuredText</code>			(p.278),
<code>cmas.ontology.processplan.SkillCall</code>	(p.308),	<code>cmas.ontology.Reference</code>	(p.287),
<code>cmas.ontology.variables.Variable</code>	(p.333),	<code>cmas.ontology.variables.VariableArray</code>	(p.337),
<code>cmas.ontology.variables.VariableBoolean</code>	(p.343),	<code>cmas.ontology.variables.VariableInteger</code>	
<code>(p.349),</code>		<code>cmas.ontology.variables.VariableObject</code>	(p.356),

`cmas.ontology.variables.VariableProperty` (*p.362*), `cmas.ontology.variables.VariableReal` (*p.368*), `cmas.ontology.variables.VariableString` (*p.374*), `cmas.ontology.workflow.Requirement` (*p.293*), and `cmas.ontology.workflow.WorkFlowLog` (*p.386*).

References `Entity()`.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from `cmas.ontology.Container` (*p.161*).

void cmas.ontology.agents.Agent.Deserialize (JsonObject job) throws ExceptionType

Reimplemented from `cmas.ontology.Entity` (*p.178*).

Entity cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static]

Reimplemented from `cmas.ontology.Entity` (*p.178*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in `cmas.ontology.Reference` (*p.288*).

Referenced by `cmas.ontology.processplan.ProcessPlanStructuredText.run()`, and `cmas.ontology.variables.VariableArray.setName()`.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.agents.Agent.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

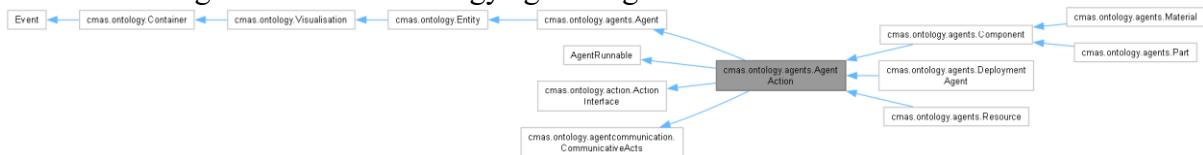
Referenced by **Entity()**.

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

- Agent.java

cmas.ontology.agents.AgentAction Class Reference

Inheritance diagram for cmas.ontology.agents.AgentAction:



Classes

enum AgentStatePublic Member Functions

- **AgentAction (AgentBaseTypes type)**
- **AgentAction (AgentAction entity, boolean instance)**
- synchronized String **getActions ()**
- synchronized void **setActions (String actions)**
- synchronized void **setState (AgentState newState)**
- synchronized **AgentState getState ()**
- void **DeSerialize (JsonObject job) throws ExceptionType**
- JsonObject **Serialize ()**
- synchronized void **setPaused (boolean value)**
- synchronized boolean **getPaused ()**
- synchronized void **postMessage (ACLMensaje message)**
- **Entity getCommunicative ()**
- void **onInit ()**
- void **onDepot ()**
- void **onDeploy ()**
- void **onRun ()**
- void **Running ()**
- void **onUndeploy ()**
- void **onFinished ()**
- void **onFailed ()**
- void **onStop ()**
- void **onMessage (ACLMensaje message)**
- void **run ()**
- **MessageHandler getMessageHandler ()**
- void **gotoLine (Integer line)**
- void **stopEditor ()**
- void **compile () throws ExceptionParameters, Exception**
- String **getPhysicalID ()**
- void **setPhysicalID (String physicalID)**
- String **getAgentSpecificType ()**
- void **setAgentSpecificType (String agentSpecificType)**
- **AgentBaseTypes getAgentBaseType ()**
- void **addAbstractInterface (AbstractInterface abstractInterface, String Name)**
- void **clearAbstractInterfaces ()**
- **AbstractInterface getAbstractInterface (String Name)**
- RuntimeTree **getRuntimeTree ()**
- void **setRuntimeTree (RuntimeTree runtimeTree)**
- Domain **getRuntimeDomain ()**
- void **setRuntimeDomain (Domain runtimeDomain)**
- LinkedBlockingQueue< **WorkFlowLog > getHistory ()**
- void **setHistory (LinkedBlockingQueue< WorkFlowLog > p)**
- void **addToHistory (WorkFlowLog log)**
- Boolean **getAutoDeploy ()**
- void **setAutoDeploy (Boolean autoDeploy)**
- abstract Entity **copy (boolean instance)**

- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType** **getType** ()
- **Properties** **getProperties** ()
- **Property<?>** **getProperty** (String Name)
- **Property<?>** **getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (**MessageHandler** messageHandler, ACLMessage message, **Performative** answerType) throws InterruptedException

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static AgentBaseTypes **getAgentType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- Action **action**
- MessageHandler **messageHandler**
- Properties **properties**
- Relations **relations**

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract], [inherited]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
Reimplemented in cmas.ontology.agents.Material (p.237), cmas.ontology.agents.Resource (p.298), cmas.ontology.Executable (p.187), cmas.ontology.interfaces.Goal (p.195), cmas.ontology.interfaces.InterfaceAssembly cmas.ontology.interfaces.InterfaceLocation cmas.ontology.interfaces.InterfaceSpecialisation cmas.ontology.interfaces.InterfaceTransport cmas.ontology.processplan.ProcessPlanGoals cmas.ontology.processplan.ProcessPlanJava cmas.ontology.processplan.ProcessPlanStructuredText cmas.ontology.processplan.SkillCall (p.308), cmas.ontology.variables.Variable (p.333), cmas.ontology.variables.VariableBoolean (p.343), cmas.ontology.variables.VariableProperty (p.362), cmas.ontology.workflow.Requirement (p.293), and and cmas.ontology.workflow.WorkFlowLog (p.386).	cmas.ontology.agents.DeploymentAgent (p.174), cmas.ontology.agents.Part (p.255), cmas.ontology.batch.Batch (p.151), cmas.ontology.Hazard (p.199), cmas.ontology.interfaces.Interface (p.204), cmas.ontology.interfaces.InterfaceAssembly cmas.ontology.interfaces.InterfaceLocation cmas.ontology.interfaces.InterfaceSpecialisation cmas.ontology.interfaces.InterfaceTransport cmas.ontology.processplan.ProcessPlanGoals cmas.ontology.processplan.ProcessPlanJava cmas.ontology.processplan.ProcessPlanStructuredText cmas.ontology.processplan.SkillCall (p.287), cmas.ontology.variables.VariableArray (p.337), cmas.ontology.variables.VariableInteger (p.349), cmas.ontology.variables.VariableObject (p.356), cmas.ontology.variables.VariableReal (p.368), cmas.ontology.variables.VariableString (p.374), cmas.ontology.workflow.WorkFlowLog (p.386).

References **Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

void cmas.ontology.agents.AgentAction.Deserialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.agents.Agent** (p.138).

Entity cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (p.178).

Entity cmas.ontology.agents.AgentAction.getCommunicative ()

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (p.155).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.agents.AgentAction.getMessageHandler ()

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

void cmas.ontology.agents.AgentAction.gotoLine (Integer line)

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

synchronized void cmas.ontology.agents.AgentAction.postMessage (ACLMessages message)

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.agents.AgentAction.Serialize ()

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

Description	a user defined description
-------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

ID	the new ID value.
----	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

Name	the new name.
------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

Referenced by **Entity()**.

void cmas.ontology.agents.AgentAction.stopEditor ()

Implements **cmas.ontology.action.ActionInterface** (p.133).

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

- AgentAction.java

cmas.ontology.agents.Agent.AgentBaseTypes Enum Reference

Public Attributes

- PART
- MATERIAL
- RESOURCE
- DEPLOYMENTAGENT

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

- Agent.java

cmas.ontology.agents.AgentAction.AgentState Enum Reference

Public Attributes

- **INIT**
- **DEPOT**
- **DEPLOY**
- **RUNNING**
- **UNDEPLOY**
- **FINISHED**
- **FAILED**
- **STOPPED**

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

- AgentAction.java

cmas.ontology.interfaces.Attachment Class Reference

Public Member Functions

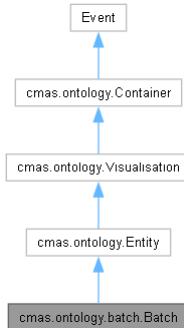
- **InterfaceBase getLocal ()**
- **void setLocal (InterfaceBase local)**
- **InterfaceBase getRemote ()**
- **void setRemote (InterfaceBase remote)**
- **boolean isAttached ()**
- **boolean isSource ()**
- **String getAttachedTo ()**
- **void makeConnection (InterfaceBase localInterface, InterfaceBase remoteInterface, boolean source) throws ExceptionRuntime**
- **String toString ()**

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

- Attachment.java

cmas.ontology.batch.Batch Class Reference

Inheritance diagram for cmas.ontology.batch.Batch:



Public Member Functions

- **Batch (Batch deployBatch, boolean instance)**
- **ArrayList< BatchItem > getBatch ()**
- **void setBatch (ArrayList< BatchItem > value)**
- **void clear ()**
- **int size ()**
- **Integer getNumberOfItems (int i)**
- **BatchItem getDeploymentItem (int i)**
- **Pair< Agent, InterfaceBase > getDeploymentInstances (int i)**
- **JSONArray SerializeBatch ()**
- **ArrayList< BatchItem > DeSerializeBatch (String value)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **void add (BatchItem batchItem)**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRelations ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**
- **ArrayList< Entity > getSpecificRelationType (String Name)**
- **void addSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (Entity ent)**

- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassname** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static Entity **Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
 - Relations **relations**
-

Member Function Documentation

Entity **cmas.ontology.batch.Batch.copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

References **cmas.ontology.Entity.Entity()**.

void **cmas.ontology.Entity.delete** () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void **cmas.ontology.Entity.DeSerialize** (JsonObject job) throws **ExceptionType** [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

String **cmas.ontology.Entity.getDescription** () [inherited]

Returns

the entity description.

String cmas.ontology.batch.Batch.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.batch.Batch.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

Referenced by **Entity()**.

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

- Batch.java

cmas.ontology.batch.BatchItem Class Reference

Public Member Functions

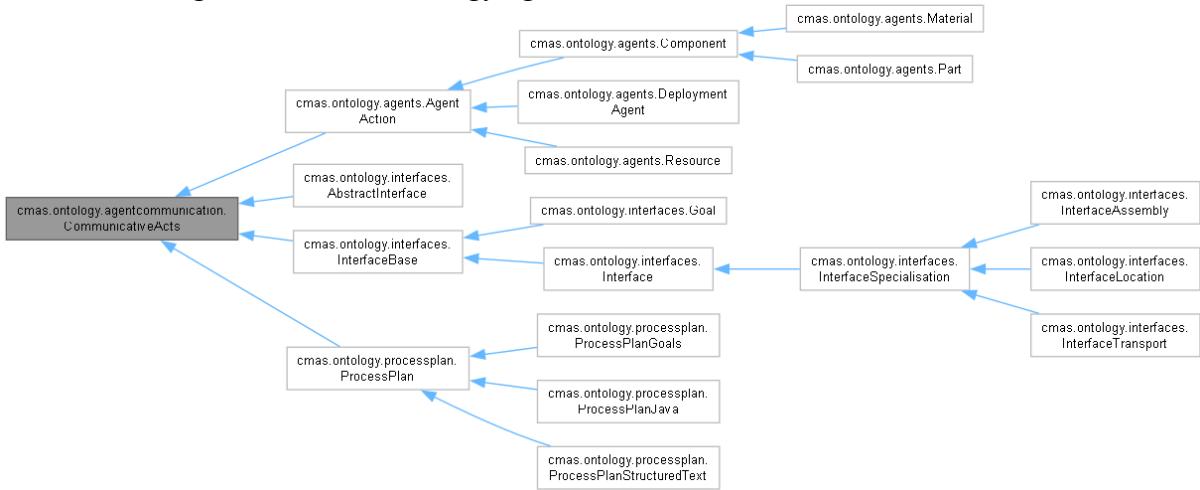
- **BatchItem** (String agentID, String attachToID)
- **BatchItem** (**BatchItem** batchItem)
- **BatchItem** (JsonObject asJsonObject)
- **BatchItem copy ()**
- **String getAgentID ()**
- **void setAgentID (String agentID)**
- **String getAttachToID ()**
- **void setAttachToID (String attachToID)**
- **Integer getNumberOfItems ()**
- **void setNumberOfItems (Integer numberOfItems)**
- **String getDescription ()**
- **void setDescription (String description)**
- **JsonObject serialize ()**
- **void deSerialize (JsonObject obj)**

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

- BatchItem.java

cmas.ontology.agentcommunication.CommunicativeActs Interface Reference

Inheritance diagram for cmas.ontology.agentcommunication.CommunicativeActs:



Public Member Functions

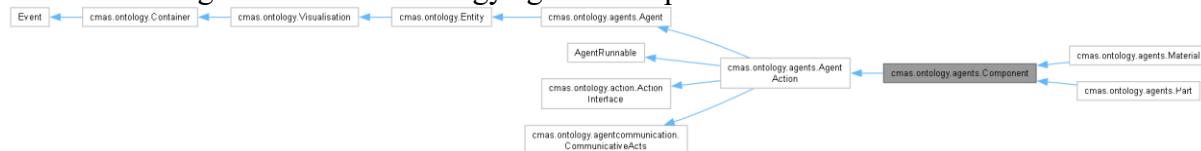
- `Entity getCommunicative ()`
- `MessageHandler getMessageHandler ()`
- `void postMessage (ACLMensaje message)`
- `default ACLMensaje waitForPerformative (MessageHandler messageHandler, ACLMensaje message, Performative answerType) throws InterruptedException`

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

- `CommunicativeActs.java`

cmas.ontology.agents.Component Class Reference

Inheritance diagram for cmas.ontology.agents.Component:



Public Member Functions

- **Component (AgentBaseTypes type)**
- **Component (Component entity, boolean instance)**
- synchronized String **getActions ()**
- synchronized void **setActions (String actions)**
- synchronized void **setState (AgentState newState)**
- synchronized AgentState **getState ()**
- void **DeSerialize (JsonObject job) throws ExceptionType**
- JsonObject **Serialize ()**
- synchronized void **setPaused (boolean value)**
- synchronized boolean **getPaused ()**
- synchronized void **postMessage (ACLMensaje message)**
- **Entity getCommunicative ()**
- void **onInit ()**
- void **onDepot ()**
- void **onDeploy ()**
- void **onRun ()**
- void **Running ()**
- void **onUndeploy ()**
- void **onFinished ()**
- void **onFailed ()**
- void **onStop ()**
- void **onMessage (ACLMensaje message)**
- void **run ()**
- **MessageHandler getMessageHandler ()**
- void **gotoLine (Integer line)**
- void **stopEditor ()**
- void **compile () throws ExceptionParameters, Exception**
- String **getPhysicalID ()**
- void **setPhysicalID (String physicalID)**
- String **getAgentSpecificType ()**
- void **setAgentSpecificType (String agentSpecificType)**
- AgentBaseTypes **getAgentBaseType ()**
- void **addAbstractInterface (AbstractInterface abstractInterface, String Name)**
- void **clearAbstractInterfaces ()**
- AbstractInterface **getAbstractInterface (String Name)**
- RuntimeTree **getRuntimeTree ()**
- void **setRuntimeTree (RuntimeTree runtimeTree)**
- Domain **getRuntimeDomain ()**
- void **setRuntimeDomain (Domain runtimeDomain)**
- LinkedBlockingQueue<WorkFlowLog> **getHistory ()**
- void **setHistory (LinkedBlockingQueue<WorkFlowLog> p)**
- void **addToHistory (WorkFlowLog log)**
- Boolean **getAutoDeploy ()**
- void **setAutoDeploy (Boolean autoDeploy)**
- abstract Entity **copy (boolean instance)**
- void **delete ()**
- String **toString ()**
- String **getID ()**

- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealtions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static AgentBaseTypes **getAgentType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- Action **action**
- MessageHandler **messageHandler**
- Properties **properties**
- Relations **relations**

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract], [inherited]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity	
Reimplemented in cmas.ontology.agents.Material	cmas.ontology.agents.DeploymentAgent (p.174), cmas.ontology.agents.Part (p.255),	
cmas.ontology.agents.Resource	cmas.ontology.batch.Batch (p.151),	
cmas.ontology.Executable	cmas.ontology.Hazard (p.199),	
cmas.ontology.interfaces.Goal	cmas.ontology.interfaces.Interface (p.204),	
cmas.ontology.interfaces.InterfaceAssembly		(p.209),
cmas.ontology.interfaces.InterfaceLocation		(p.219),
cmas.ontology.interfaces.InterfaceSpecialisation		(p.224),
cmas.ontology.interfaces.InterfaceTransport		(p.230),
cmas.ontology.processplan.ProcessPlanGoals		(p.266),
cmas.ontology.processplan.ProcessPlanJava		(p.272),
cmas.ontology.processplan.ProcessPlanStructuredText		(p.278),
cmas.ontology.processplan.SkillCall (p.308),	cmas.ontology.Reference (p.287),	
cmas.ontology.variables.Variable (p.333),	cmas.ontology.variables.VariableArray (p.337),	
cmas.ontology.variables.VariableBoolean (p.343),	cmas.ontology.variables.VariableInteger (p.349),	
	cmas.ontology.variables.VariableObject (p.356),	
cmas.ontology.variables.VariableProperty (p.362),	cmas.ontology.variables.VariableReal (p.368),	
	cmas.ontology.variables.VariableString (p.374),	
cmas.ontology.workflow.Requirement (p.293), and	cmas.ontology.workflow.WorkFlowLog (p.386).	

References **Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

void cmas.ontology.agents.AgentAction.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (p.138).

Entity cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (p.178).

Entity cmas.ontology.agents.AgentAction.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (p.155).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.agents.AgentAction.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

void cmas.ontology.agents.AgentAction.gotoLine (Integer line) [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

synchronized void cmas.ontology.agents.AgentAction.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.agents.AgentAction.Serialize () [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by Entity().

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in cmas.ontology.Reference (p.289), and
cmas.ontology.variables.VariableArray (p.339).

Referenced by Entity().

void cmas.ontology.agents.AgentAction.stopEditor () [inherited]

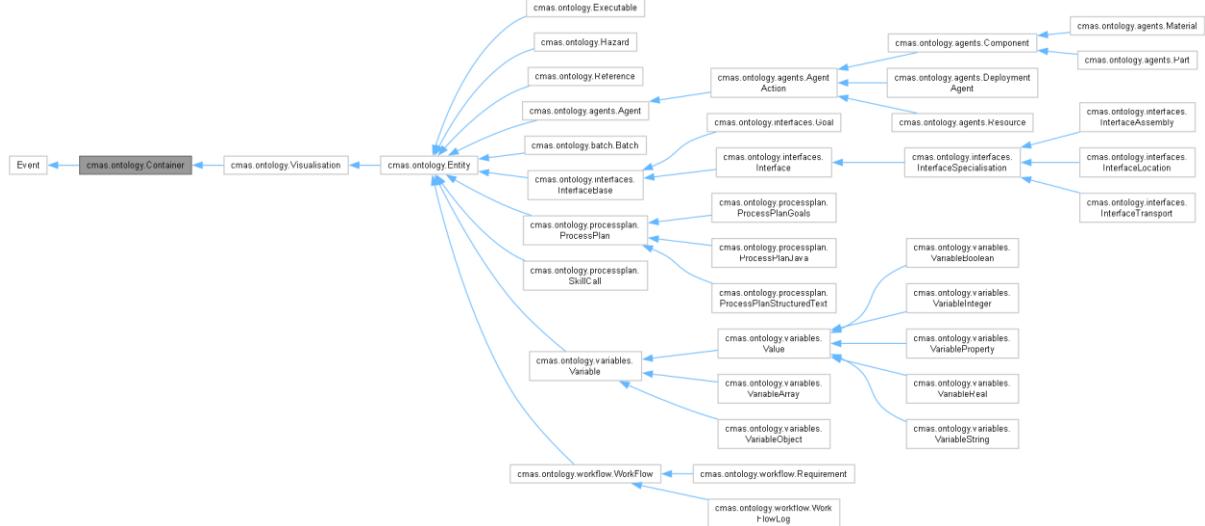
Implements cmas.ontology.action.ActionInterface (p.133).

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

- Component.java

cmas.ontology.Container Class Reference

Inheritance diagram for cmas.ontology.Container:



Public Member Functions

- `Container (EventType TypeOfEvent)`
- `Container (JsonObject job) throws ExceptionType`
- `Container (Container container, boolean instance)`
- `void copyProperties (Container container, boolean instance)`
- `void delete ()`
- `void setOverride (Boolean value)`
- `Boolean getProtected ()`
- `void setProtected (Boolean protectedValue)`
- `Object getUserData ()`
- `void setUserData (Object userData)`
- `String getClass Name ()`
- `void setClassName (String newName)`
- `Relations getRealtionsObject ()`
- `JsonObject Serialize ()`
- `ArrayList< BreakPoint > getBreakPointList ()`
- `void setBreakPointList (ArrayList< BreakPoint > breakPointList)`
- `Boolean isInstance ()`

Protected Attributes

- `Properties properties`
- `Relations relations`

Package Functions

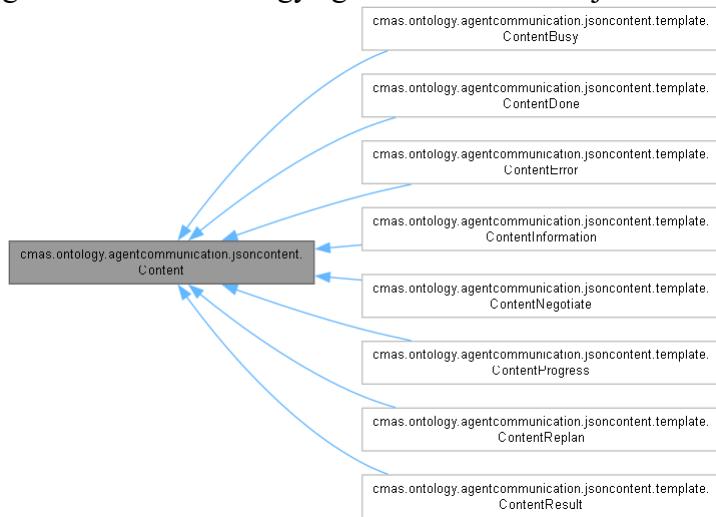
- `void DeSerialize (JsonObject job) throws ExceptionType`

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

- `Container.java`

cmas.ontology.agentcommunication.jsoncontent.Content Class Reference

Inheritance diagram for cmas.ontology.agentcommunication.jsoncontent.Content:



Public Member Functions

- **Content** (String method)
- **Content** (String method, String name, String value)
- String **getParameterValue** (String ParameterName)
- boolean **getParameterValueAsBoolean** (String ParameterName, boolean DefaultValue)
- void **createParameters** (String method, String... values)
- int **getNumberOfParameters** ()
- ContentParameter[] **getAllParameters** ()
- ContentParameter **getParameter** (int index)
- Content **addParameter** (ContentParameter parameter)
- String **toString** ()
- String **serialize** ()
- String **getMethod** ()
- void **setMethod** (String method)

Static Public Member Functions

- static Content **deserialize** (String stringJSON)

Member Function Documentation

void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters (String method, String... values)

Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

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

- Content.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentBusy Class Reference

Inheritance diagram for

cmas.ontology.agentcommunication.jsoncontent.template.ContentBusy:



Public Member Functions

- **ContentBusy** (String Message)
- String **getParameterValue** (String ParameterName)
- boolean **getParameterValueAsBoolean** (String ParameterName, boolean DefaultValue)
- void **createParameters** (String method, String... values)
- int **getNumberOfParameters** ()
- ContentParameter[] **getAllParameters** ()
- ContentParameter **getParameter** (int index)
- Content **addParameter** (ContentParameter parameter)
- String **toString** ()
- String **serialize** ()
- String **getMethod** ()
- void **setMethod** (String method)

Static Public Member Functions

- static String **deserialize** (Content content)
- static Content **deserialize** (String stringJSON)

Static Public Attributes

- static final String **METHOD** ="Busy"

Member Function Documentation

**void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters
(String method, String... values) [inherited]**

Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

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

- ContentBusy.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentDone Class Reference

Inheritance diagram for
cmas.ontology.agentcommunication.jsoncontent.template.ContentDone:



Public Member Functions

- **ContentDone (ProcessPlanReturnValue value)**
- **String getParameterValue (String ParameterName)**
- **boolean getParameterValueAsBoolean (String ParameterName, boolean DefaultValue)**
- **void createParameters (String method, String... values)**
- **int getNumberOfParameters ()**
- **ContentParameter[] getAllParameters ()**
- **ContentParameter getParameter (int index)**
- **Content addParameter (ContentParameter parameter)**
- **String toString ()**
- **String serialize ()**
- **String getMethod ()**
- **void setMethod (String method)**

Static Public Member Functions

- static **ProcessPlanReturnValue deserialize (Content content)**
- static **Content deserialize (String stringJSON)**

Static Public Attributes

- static final String **METHOD** ="Done"

Member Function Documentation

void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters (String method, String... values) [inherited]

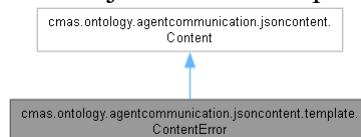
Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

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

- ContentDone.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentError Class Reference

Inheritance diagram for
cmas.ontology.agentcommunication.jsoncontent.template.ContentError:



Public Member Functions

- **ContentError** (String Message, Integer ReturnValue)
- String **getParameterValue** (String ParameterName)
- boolean **getParameterValueAsBoolean** (String ParameterName, boolean DefaultValue)
- void **createParameters** (String method, String... values)
- int **getNumberOfParameters** ()
- ContentParameter[] **getAllParameters** ()
- ContentParameter **getParameter** (int index)
- Content **addParameter** (ContentParameter parameter)
- String **toString** ()
- String **serialize** ()
- String **getMethod** ()
- void **setMethod** (String method)

Static Public Member Functions

- static Content **deserialize** (String stringJSON)

Static Public Attributes

- static final String **METHOD** ="Error"

Member Function Documentation

**void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters
(String method, String... values) [inherited]**

Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

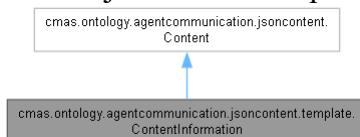
The documentation for this class was generated from the following file:

- ContentError.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentInformation Class Reference

Inheritance diagram for

cmas.ontology.agentcommunication.jsoncontent.template.ContentInformation:



Public Member Functions

- **ContentInformation (String Message)**
- **String getParameterValue (String ParameterName)**
- **boolean getParameterValueAsBoolean (String ParameterName, boolean DefaultValue)**
- **void createParameters (String method, String... values)**
- **int getNumberOfParameters ()**
- **ContentParameter[] getAllParameters ()**
- **ContentParameter getParameter (int index)**
- **Content addParameter (ContentParameter parameter)**
- **String toString ()**
- **String serialize ()**
- **String getMethod ()**
- **void setMethod (String method)**

Static Public Member Functions

- static String **deserialize (Content content)**
- static Content **deserialize (String stringJSON)**

Static Public Attributes

- static final String **METHOD** ="Information"

Member Function Documentation

void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters (String method, String... values) [inherited]

Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

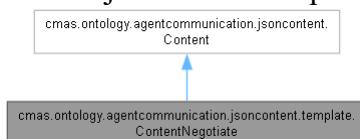
The documentation for this class was generated from the following file:

- ContentInformation.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentNegotiate Class Reference

Inheritance diagram for

cmas.ontology.agentcommunication.jsoncontent.template.ContentNegotiate:



Public Member Functions

- **String getParameterValue (String ParameterName)**
- **boolean getParameterValueAsBoolean (String ParameterName, boolean DefaultValue)**
- **void createParameters (String method, String... values)**
- **int getNumberOfParameters ()**
- **ContentParameter[] getAllParameters ()**
- **ContentParameter getParameter (int index)**
- **Content addParameter (ContentParameter parameter)**
- **String toString ()**
- **String serialize ()**
- **String getMethod ()**
- **void setMethod (String method)**

Static Public Member Functions

- static **Content deserialize (String stringJSON)**

Static Public Attributes

- static final String **METHOD** ="Negotiate"

Member Function Documentation

void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters (String method, String... values) [inherited]

Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

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

- ContentNegotiate.java

cmas.ontology.agentcommunication.jsoncontent.ContentParameter Class Reference

Public Member Functions

- **ContentParameter** (String name, String value)
 - String **getName** ()
 - String **getValue** ()
 - void **setName** (String name)
 - void **setValue** (String value)
 - String **toString** ()
-

Detailed Description

This class define json method parameter.

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

- ContentParameter.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentProgress Class Reference

Inheritance diagram for

cmas.ontology.agentcommunication.jsoncontent.template.ContentProgress:



Public Member Functions

- **ContentProgress** (Integer ProgressValue)
- String **getParameterValue** (String ParameterName)
- boolean **getParameterValueAsBoolean** (String ParameterName, boolean DefaultValue)
- void **createParameters** (String method, String... values)
- int **getNumberOfParameters** ()
- ContentParameter[] **getAllParameters** ()
- ContentParameter **getParameter** (int index)
- Content **addParameter** (ContentParameter parameter)
- String **toString** ()
- String **serialize** ()
- String **getMethod** ()
- void **setMethod** (String method)

Static Public Member Functions

- static Content **deserialize** (String stringJSON)

Static Public Attributes

- static final String **METHOD** ="Progress"

Member Function Documentation

**void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters
(String method, String... values) [inherited]**

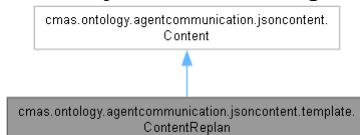
Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

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

- ContentProgress.java

cmas.ontology.agentcommunication.jsoncontent.template.ContentReplan Class Reference

Inheritance diagram for
cmas.ontology.agentcommunication.jsoncontent.template.ContentReplan:



Public Member Functions

- `String getParameterValue (String ParameterName)`
- `boolean getParameterValueAsBoolean (String ParameterName, boolean DefaultValue)`
- `void createParameters (String method, String... values)`
- `int getNumberOfParameters ()`
- `ContentParameter[] getAllParameters ()`
- `ContentParameter getParameter (int index)`
- `Content addParameter (ContentParameter parameter)`
- `String toString ()`
- `String serialize ()`
- `String getMethod ()`
- `void setMethod (String method)`

Static Public Member Functions

- `static Content deserialize (String stringJSON)`

Static Public Attributes

- `static final String METHOD ="Replan"`

Member Function Documentation

void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters (String method, String... values) [inherited]

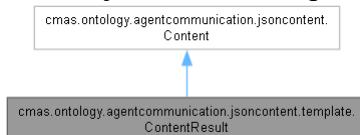
Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example `createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");`

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

- `ContentReplan.java`

cmas.ontology.agentcommunication.jsoncontent.template.ContentResult Class Reference

Inheritance diagram for
cmas.ontology.agentcommunication.jsoncontent.template.ContentResult:



Public Member Functions

- **String getParameterValue (String ParameterName)**
- **boolean getParameterValueAsBoolean (String ParameterName, boolean DefaultValue)**
- **void createParameters (String method, String... values)**
- **int getNumberOfParameters ()**
- **ContentParameter[] getAllParameters ()**
- **ContentParameter getParameter (int index)**
- **Content addParameter (ContentParameter parameter)**
- **String toString ()**
- **String serialize ()**
- **String getMethod ()**
- **void setMethod (String method)**

Static Public Member Functions

- static **Content deserialize (String stringJSON)**

Static Public Attributes

- static final String **METHOD** ="ReturnValues"

Member Function Documentation

void cmas.ontology.agentcommunication.jsoncontent.Content.createParameters (String method, String... values) [inherited]

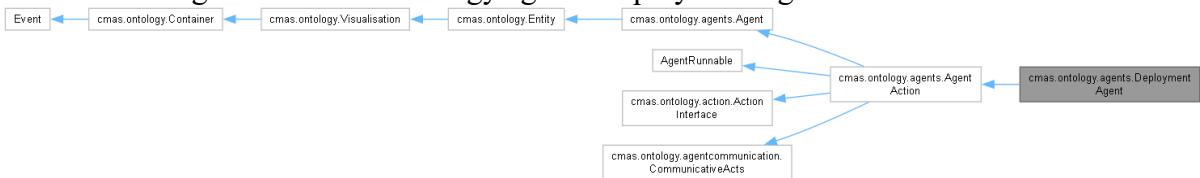
Create and add parameters based on string values. The string values must be in pair, i.e. the name of the parameter and the value. Example createParameters("myMethod", "Parameter1", "Value1", Parameter2", "Value2");

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

- ContentResult.java

cmas.ontology.agents.DeploymentAgent Class Reference

Inheritance diagram for cmas.ontology.agents.DeploymentAgent:



Classes

enum DeploymentCriteriaPublic Member Functions

- **DeploymentAgent** (*DeploymentAgent entity*, boolean *instance*)
- **Entity copy** (boolean *instance*)
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **DeploymentCriteria getDeployAt ()**
- **String getVariableTrigger ()**
- **void setVariableTrigger (String *variableName*)**
- **void setDeployAt (DeploymentCriteria *deployAt*)**
- **long getDeploymentTimeInMS ()**
- **String getDeploymentTime ()**
- **void setDeploymentTime (String *deployArgument*)**
- **Integer getNumberOfTimes ()**
- **void setNumberOfTimes (Integer *numberOfTimes*)**
- **void onRun ()**
- **void onUndeploy ()**
- **void onFailed ()**
- **void onStop ()**
- **void onFinished ()**
- **synchronized String getActions ()**
- **synchronized void setActions (String *actions*)**
- **synchronized void setState (AgentState *newState*)**
- **synchronized AgentState getState ()**
- **void DeSerialize (JsonObject *job*) throws ExceptionType**
- **JsonObject Serialize ()**
- **synchronized void setPaused (boolean *value*)**
- **synchronized boolean getPaused ()**
- **synchronized void postMessage (ACLMessages *message*)**
- **Entity getCommunicative ()**
- **void onInit ()**
- **void onDepot ()**
- **void onDeploy ()**
- **void Running ()**
- **void onMessage (ACLMessages *message*)**
- **void run ()**
- **MessageHandler getMessageHandler ()**
- **void gotoLine (Integer *line*)**
- **void stopEditor ()**
- **void compile () throws ExceptionParameters, Exception**
- **String getPhysicalID ()**
- **void setPhysicalID (String *physicalID*)**
- **String getAgentSpecificType ()**
- **void setAgentSpecificType (String *agentSpecificType*)**
- **AgentBaseTypes getAgentBaseType ()**
- **void addAbstractInterface (AbstractInterface *abstractInterface*, String *Name*)**

- void **clearAbstractInterfaces** ()
- **AbstractInterface** **getAbstractInterface** (String Name)
- **RuntimeTree** **getRuntimeTree** ()
- void **setRuntimeTree** (RuntimeTree runtimeTree)
- Domain **getRuntimeDomain** ()
- void **setRuntimeDomain** (Domain runtimeDomain)
- LinkedBlockingQueue< **WorkFlowLog** > **getHistory** ()
- void **setHistory** (LinkedBlockingQueue< **WorkFlowLog** > p)
- void **addToHistory** (**WorkFlowLog** log)
- Boolean **getAutoDeploy** ()
- void **setAutoDeploy** (Boolean autoDeploy)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType** **getType** ()
- **Properties** **getProperties** ()
- **Property**<?> **getProperty** (String Name)
- **Property**<?> **getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property**<?> Item)
- boolean **isCompatibleRelation** (String Name, **Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- LinkedHashMap< String, ArrayList< **RelationObject** > > **getRealitions** ()
- Collection< ArrayList< **RelationObject** > > **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- **Entity** **findRelationByName** (String RelationName, String EntityName)
- **Entity** **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< **Entity** > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (**Entity** ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- **Relations** **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default **ACLMessages** **waitForPerformative** (**MessageHandler** messageHandler, **ACLMessages** message, **Performative** answerType) throws InterruptedException

Static Public Member Functions

- static **Entity** **Factory** (JsonObject job) throws **ExceptionType**

- static **Entity Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static **AgentBaseTypes getAgentType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- **Action action**
 - **MessageHandler messageHandler**
 - **Properties properties**
 - **Relations relations**
-

Member Function Documentation

Entity cmas.ontology.agents.DeploymentAgent.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.agents.AgentAction.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

Entity cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.agents.AgentAction.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.agents.DeploymentAgent.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.agents.DeploymentAgent.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.agents.AgentAction.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.agents.AgentAction.gotoLine (Integer line) [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

void cmas.ontology.agents.DeploymentAgent.onFailed ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.DeploymentAgent.onFinished ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.DeploymentAgent.onRun ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.DeploymentAgent.onStop ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.DeploymentAgent.onUndeploy ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

synchronized void cmas.ontology.agents.AgentAction.postMessage (ACLMessages message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.agents.AgentAction.Serialize () [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.agents.AgentAction.stopEditor () [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

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

- DeploymentAgent.java

cmas.ontology.agents.DeploymentAgent.DeploymentCriteria Enum Reference

Public Attributes

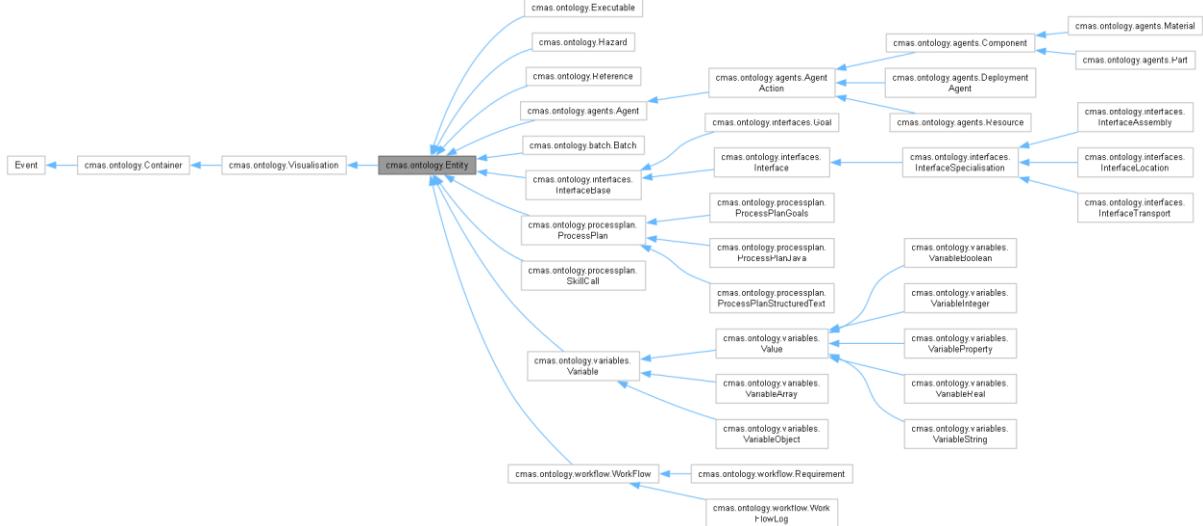
- TIME
- INTERVAL

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

- DeploymentAgent.java

cmas.ontology.Entity Class Reference

Inheritance diagram for cmas.ontology.Entity:



Classes

- enum `EntityTypeclass` `UID`

Public Member Functions

- `Entity (EntityType Type)`
- `Entity (Entity entity, boolean instance)`
- abstract `Entity copy (boolean instance)`
- `void delete ()`
- `String toString ()`
- `String getID ()`
- `void setID (String ID)`
- `void setID ()`
- `String getName ()`
- `void setName (String Name)`
- `String getParentName ()`
- `void setParent (Entity ent)`
- `String getDescription ()`
- `void setDescription (String Description)`
- `String getInstanceID ()`
- `EntityType getType ()`
- `Properties getProperties ()`
- `Property<?> getProperty (String Name)`
- `Property<?> getPropertyIgnoreCase (String Name)`
- `void addProperty (Property<?> Item)`
- `boolean isCompatibleRelation (String Name, Entity ent)`
- `boolean hasRelation (String Name, Entity ent)`
- `LinkedHashMap< String, ArrayList< RelationObject > > getRelations ()`
- `Collection< ArrayList< RelationObject > > getAllRelations ()`
- `ArrayList< RelationObject > getSpecificRelations (String Name)`
- `Entity findRelationByName (String RelationName, String EntityName)`
- `Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)`
- `ArrayList< Entity > getSpecificRelationType (String Name)`
- `void addSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (Entity ent)`

- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass Name** ()
- void **setClassName** (String newName)
- Relations **getRealtonsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static Entity **Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties properties
- Relations relations

Constructor & Destructor Documentation

cmas.ontology.Entity.Entity (EntityType Type)

Entity constructor for all types based on **Entity**. **Entity** is an abstract class. **Entity** will be assigned an unique identifier for the entity i.e., ID that can be read by **getID()**. Default the name will be an empty string to change the name use **setName()**.

Parameters

Type	the base type of entity to create. Note this type can not be changed after the Entity is created.
------	--

References **setID()**, and **setName()**.

Referenced by **cmas.ontology.batch.Batch.copy()**, **copy()**, **cmas.ontology.Executable.copy()**, **cmas.ontology.Hazard.copy()**, **cmas.ontology.processplan.SkillCall.copy()**, **cmas.ontology.Reference.copy()**, **cmas.ontology.variables.Variable.copy()**, and **Entity()**.

cmas.ontology.Entity.Entity (Entity entity, boolean instance)

Copy constructor.

Parameters

entity	the entity to copy
instance	if instance is true a runtime entity will be create otherwise an modelling entity If instance is true all references will be replaces with instances of the referenced objects.

References **Entity()**.

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity	
Reimplemented in cmas.ontology.agents.Material	cmas.ontology.agents.DeploymentAgent	(p.174), (p.237),
cmas.ontology.agents.Resource	cmas.ontology.agents.Part	(p.255), (p.298),
cmas.ontology.Executable	cmas.ontology.Batch	(p.151), (p.187),
cmas.ontology.interfaces.Goal	cmas.ontology.Hazard	(p.199), (p.195),
cmas.ontology.interfaces.InterfaceAssembly	cmas.ontology.interfaces.Interface	(p.204), (p.209),
cmas.ontology.interfaces.InterfaceLocation		(p.219),
cmas.ontology.interfaces.InterfaceSpecialisation		(p.224),
cmas.ontology.interfaces.InterfaceTransport		(p.230),
cmas.ontology.processplan.ProcessPlanGoals		(p.266),
cmas.ontology.processplan.ProcessPlanJava		(p.272),
cmas.ontology.processplan.ProcessPlanStructuredText		(p.278),
cmas.ontology.processplan.SkillCall	cmas.ontology.Reference	(p.287), (p.308),
cmas.ontology.variables.Variable	cmas.ontology.variables.VariableArray	(p.337), (p.333),
cmas.ontology.variables.VariableBoolean	cmas.ontology.variables.VariableInteger	(p.349), (p.343),
	cmas.ontology.variables.VariableObject	(p.356),
cmas.ontology.variables.VariableProperty	cmas.ontology.variables.VariableReal	(p.362), (p.368),
	cmas.ontology.variables.VariableString	(p.374), (p.386).
cmas.ontology.workflow.Requirement	cmas.ontology.workflow.WorkFlowLog	(p.293), and (p.386).

References **Entity()**.

void cmas.ontology.Entity.delete ()

Reimplemented from **cmas.ontology.Container** (p.161).

void cmas.ontology.Entity.DeSerialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.Container** (p.161).

String cmas.ontology.Entity.getDescription ()

Returns

the entity description.

String cmas.ontology.Entity.getID ()

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID ()

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName ()

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType ()**Returns**

entity type.

JsonObject cmas.ontology.Entity.Serialize ()

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description)

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID ()

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID)

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name)

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

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

- Entity.java

cmas.ontology.Entity.EntityType Enum Reference

Public Attributes

- **ENTITY**
 - **VARIABLE**
 - **GOAL**
 - **AGENT**
 - **SKILL**
 - **INTERFACE**
 - **PROCESSPLAN**
 - **REFERENCE**
 - **EXECUTABLE**
 - **WORKFLOW**
 - **HAZARD**
 - **BATCH**
 - **DATA_ROOT**
-

Detailed Description

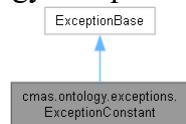
Base type of the entity object.

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

- Entity.java

cmas.ontology.exceptions.ExceptionConstant Class Reference

Inheritance diagram for cmas.ontology.exceptions.ExceptionConstant:



Public Member Functions

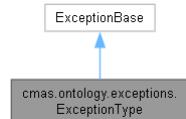
- **ExceptionConstant** (String Message)
- **ExceptionConstant** (String Message, String FileName, int LineNumber)

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

- ExceptionConstant.java

cmas.ontology.exceptions.ExceptionType Class Reference

Inheritance diagram for cmas.ontology.exceptions.ExceptionType:



Public Member Functions

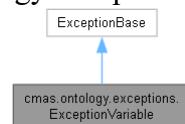
- **ExceptionType** (String Message)

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

- `ExceptionType.java`

cmas.ontology.exceptions.ExceptionVariable Class Reference

Inheritance diagram for cmas.ontology.exceptions.ExceptionVariable:



Public Member Functions

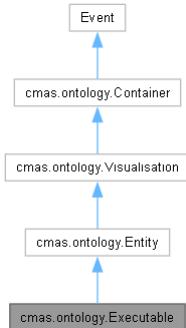
- **ExceptionVariable** (String Message)

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

- ExceptionVariable.java

cmas.ontology.Executable Class Reference

Inheritance diagram for cmas.ontology.Executable:



Classes

enum ExecutableTypesPublic Member Functions

- **Executable** (**Executable** exec, boolean instance)
- **Entity copy** (boolean instance)
- **String getProgram ()**
- **void setProgram (String program)**
- **void compile ()** throws **ExceptionExpressionType**, **ExceptionFailedToExecuteStatement**, **ExceptionUserError**, **ExceptionParser**, **ExceptionBase**, **ExceptionRuntime**, **ExceptionExit**
- **void run ()** throws **ExceptionParameters**, **Exception**
- **void DeSerialize (JsonObject job)** throws **ExceptionType**
- **JsonObject Serialize ()**
- **ExecutableTypes getExecutableType ()**
- **void setExecutableType (ExecutableTypes executableType)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRealtons ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**
- **ArrayList< Entity > getSpecificRelationType (String Name)**
- **void addSpecificRelation (String Name, Entity ent)**

- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassname** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static Entity **Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
 - Relations **relations**
-

Member Function Documentation

Entity **cmas.ontology.Executable.copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

References **cmas.ontology.Entity.Entity()**.

void **cmas.ontology.Entity.delete** () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void **cmas.ontology.Executable.Deserialize** (JsonObject job) throws **ExceptionType**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

String **cmas.ontology.Entity.getDescription** () [inherited]

Returns

the entity description.

String cmas.ontology.Executable.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Executable.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.Executable.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

Referenced by **Entity()**.

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

- Executable.java

cmas.ontology.Executable.ExecutableTypes Enum Reference

Public Attributes

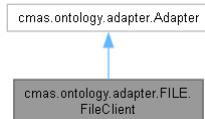
- **ST**
- **SFC**

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

- Executable.java

cmas.ontology.adapter.FILE.FileClient Class Reference

Inheritance diagram for cmas.ontology.adapter.FILE.FileClient:



Public Member Functions

- **FileClient** (String Source)
- void **SetSource** (String Source)
- String **ReadData** (String Address, String Name, Object Value)
- void **WriteData** (String Address, String Name, Boolean bValue)
- void **WriteData** (String Address, String Name, String sValue)
- void **WriteData** (String Address, String Name, Integer iValue)
- void **WriteData** (String Address, String Name, Double dValue)
- boolean **close** ()
- void **setDebug** (boolean bValue)
- String **getDefaultValue** ()
- String **getDescription** ()
- boolean **isValid** (String Source)
- **Adapter** **createNew** (String Source)

Detailed Description

File communications for variables

```
For adapter type of FILE the specific format is: FILE:[PATH]
eg, FILE:OUT:/tmp/outfile.txt
```

Author

Fredrik Danielsson, University West

Member Function Documentation

boolean cmas.ontology.adapter.FILE.FileClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.FILE.FileClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.FILE.FileClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.FILE.FileClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.FILE.FileClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.FILE.FileClient.ReadData (String Address, String Name, Object Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.FILE.FileClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.FILE.FileClient.WriteData (String Address, String Name, Boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.FILE.FileClient.WriteData (String Address, String Name, Double dValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.FILE.FileClient.WriteData (String Address, String Name, Integer iValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.FILE.FileClient.WriteData (String Address, String Name, String sValue)

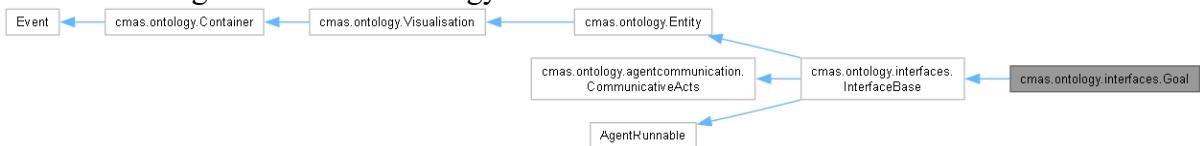
Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- FileClient.java

cmas.ontology.interfaces.Goal Class Reference

Inheritance diagram for cmas.ontology.interfaces.Goal:



Public Member Functions

- **Goal (Goal goal, boolean instance)**
- **Entity copy (boolean instance)**
- **void DeSerialize (JsonObject job) throws ExceptionType**
- **JsonObject Serialize ()**
- **String getPreCondition ()**
- **void setPreCondition (String preCondition)**
- **Integer getPriority ()**
- **void setPriority (Integer priority)**
- **Boolean getAssigned ()**
- **void setAssigned (Boolean assigned)**
- **Boolean getFulfilled ()**
- **void setFulfilled (Boolean fulfilled)**
- **String getProcessPlanName ()**
- **void setProcessPlanName (String processPlan)**
- **boolean evaluatePreCondition (Agent agent) throws ExceptionRuntime, ExceptionBase, ExceptionParser**
- **Boolean getFailed ()**
- **void setFailed (Boolean failed)**
- **Boolean getBreakPoint ()**
- **void setBreakPoint (Boolean breakPoint)**
- **Boolean getDebugArrow ()**
- **void setDebugArrow (Boolean debugArrow)**
- **String getFailedMessage ()**
- **void setFailedMessage (String failedText)**
- **String getInterface ()**
- **void setInterface (String interface1)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **String getBookedByName ()**
- **Boolean isBusy ()**
- **Boolean isBooked ()**
- **void setBooked (CommunicativeActs agent)**
- **CommunicativeActs getBooked ()**
- **Boolean isAttachedTo (InterfaceBase remoteInterface)**
- **Boolean isAttached ()**
- **String getAttachedTo ()**
- **Attachment getAttachment ()**
- **VariableMatchingType getVariableMatching ()**
- **ACLMensaje getOnGoingCommunication ()**
- **void setVariableMatching (VariableMatchingType variableRequest)**
- **void initInterface (Agent agent)**
- **void AttachTo (InterfaceBase remote) throws ExceptionRuntime**
- **void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime**
- **void Detach () throws ExceptionRuntime**
- **void _Detach ()**
- **void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime**
- **Entity getCommunicative ()**
- **void postMessage (ACLMensaje message)**

- void **run** ()
- String **getRequirements** (**Entity** skill, **Agent** callingAgent, **InterfaceBase** callingInterface) throws ExceptionRuntime
- boolean **book** (**InterfaceBase** remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException
- boolean **book** (**CommunicativeActs** delegateAnswer, **InterfaceBase** remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException
- boolean **unbook** (**CommunicativeActs** delegateAnswer, **InterfaceBase** remoteInterface) throws InterruptedException
- **Agent getRuntimeAgent** ()
- synchronized String **getActions** ()
- void **compile** () throws ExceptionParameters, Exception
- synchronized void **setActions** (String actions)
- **Action getActionObject** ()
- **MessageHandler getMessageHandler** ()
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (**Entity** ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, **Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- LinkedHashMap< String, ArrayList< **RelationObject** > > **getRealitions** ()
- Collection< ArrayList< **RelationObject** > > **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- **Entity findRelationByName** (String RelationName, String EntityName)
- **Entity findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< **Entity** > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (**Entity** ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- **Relations getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default **ACLMessages waitForPerformativ** (**MessageHandler** messageHandler, **ACLMessages** message, **Performativ** answerType) throws InterruptedException

Static Public Member Functions

- static **Entity Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static **Entity Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- **Properties properties**
- **Relations relations**

Member Function Documentation

Entity cmas.ontology.interfaces.Goal.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.interfaces.Goal.Deserialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.interfaces.Goal.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.interfaces.Goal.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.interfaces.InterfaceBase.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

void cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.interfaces.Goal.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

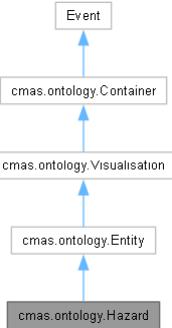
Referenced by **Entity()**.

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

- Goal.java

cmas.ontology.Hazard Class Reference

Inheritance diagram for cmas.ontology.Hazard:



Public Member Functions

- **Hazard** (Hazard hazard, boolean instance)
- **Entity copy** (boolean instance)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- String **getEntityDescription** ()
- String **getEntityIconFile** ()
- Integer **getRisk** ()
- void **setRisk** (Integer risk)
- String **getTargetType** ()
- void **setTargetType** (String target)
- Integer **getFrequency** ()
- void **setFrequency** (Integer frequency)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)

- ImageView **getEntityIcon ()**
- void **copyProperties (Container container, boolean instance)**
- void **setOverride (Boolean value)**
- Boolean **getProtected ()**
- void **setProtected (Boolean protectedValue)**
- Object **getUserData ()**
- void **setUserData (Object userData)**
- String **getClassName ()**
- void **setClassName (String newName)**
- Relations **getRealtionsObject ()**
- ArrayList< BreakPoint > **getBreakPointList ()**
- void **setBreakPointList (ArrayList< BreakPoint > breakPointList)**
- Boolean **isInstance ()**

Static Public Member Functions

- static Entity **Factory (String job)** throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static Entity **Factory (JsonObject job)** throws **ExceptionType, ExceptionConstant, ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity cmas.ontology.Hazard.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

References **cmas.ontology.Entity.Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Hazard.Deserialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.Entity** (*p.178*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Hazard.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Hazard.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.Hazard.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

Referenced by **Entity()**.

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

- Hazard.java

cmas.ontology.action.IndependentClassLoader Class Reference

Inheritance diagram for cmas.ontology.action.IndependentClassLoader:



Static Public Member Functions

- static ClassLoader getInstance ()

Member Function Documentation

ClassLoader cmas.ontology.action.IndependentClassLoader.getInstance () [static]

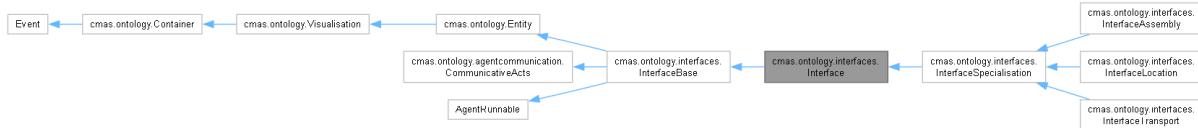
Returns
instance

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

- IndependentClassLoader.java

cmas.ontology.interfaces.Interface Class Reference

Inheritance diagram for cmas.ontology.interfaces.Interface:



Public Member Functions

- **Interface (Interface interf, boolean instance)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **String getBookedByName ()**
- **Boolean isBusy ()**
- **Boolean isBooked ()**
- **void setBooked (CommunicativeActs agent)**
- **CommunicativeActs getBooked ()**
- **Boolean isAttachedTo (InterfaceBase remoteInterface)**
- **Boolean isAttached ()**
- **String getAttachedTo ()**
- **Attachment getAttachment ()**
- **VariableMatchingType getVariableMatching ()**
- **ACLMessage getOnGoingCommunication ()**
- **void setVariableMatching (VariableMatchingType variableRequest)**
- **void initInterface (Agent agent)**
- **void AttachTo (InterfaceBase remote) throws ExceptionRuntime**
- **void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime**
- **void Detach () throws ExceptionRuntime**
- **void _Detach ()**
- **void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessage message)**
- **void run ()**
- **String getRequirements (Entity skill, Agent callingAgent, InterfaceBase callingInterface) throws ExceptionRuntime**
- **boolean book (InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean book (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean unbook (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface) throws InterruptedException**
- **Agent getRuntimeAgent ()**
- **synchronized String getActions ()**
- **void compile () throws ExceptionParameters, Exception**
- **synchronized void setActions (String actions)**
- **Action getActionObject ()**
- **MessageHandler getMessageHandler ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**

- `String getDescription ()`
- `void setDescription (String Description)`
- `String getInstanceID ()`
- `EntityType getType ()`
- `Properties getProperties ()`
- `Property<?> getProperty (String Name)`
- `Property<?> getPropertyIgnoreCase (String Name)`
- `void addProperty (Property<?> Item)`
- `boolean isCompatibleRelation (String Name, Entity ent)`
- `boolean hasRelation (String Name, Entity ent)`
- `LinkedHashMap< String, ArrayList< RelationObject > > getRealitions ()`
- `Collection< ArrayList< RelationObject > > getAllRelations ()`
- `ArrayList< RelationObject > getSpecificRelations (String Name)`
- `Entity findRelationByName (String RelationName, String EntityName)`
- `Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)`
- `ArrayList< Entity > getSpecificRelationType (String Name)`
- `void addSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (Entity ent)`
- `void eventReflection (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)`
- `void DeSerialize (JsonObject job) throws ExceptionType`
- `JsonObject Serialize ()`
- `ImageView getEntityIcon ()`
- `void copyProperties (Container container, boolean instance)`
- `void setOverride (Boolean value)`
- `Boolean getProtected ()`
- `void setProtected (Boolean protectedValue)`
- `Object getUserData ()`
- `void setUserData (Object userData)`
- `String getClassName ()`
- `void setClassName (String newName)`
- `Relations getRealitionsObject ()`
- `ArrayList< BreakPoint > getBreakPointList ()`
- `void setBreakPointList (ArrayList< BreakPoint > breakPointList)`
- `Boolean isInstance ()`
- default `ACLMessage waitForPerformative (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException`

Static Public Member Functions

- static `Interface Factory (JsonObject job) throws ExceptionType`
- static `Entity Factory (String job) throws ExceptionType, ExceptionConstant, ExceptionVariable`

Protected Attributes

- `Properties properties`
- `Relations relations`

Member Function Documentation

Entity cmas.ontology.interfaces.Interface.copy (boolean instance)

Must be override by all classes extending `Entity` if instance is true `Properties` will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<code>instance</code>	if true a runtime entity is create otherwise a modelling entity
-----------------------	---

Reimplemented from `cmas.ontology.Entity` (p.382).

Reimplemented in **cmas.ontology.interfaces.InterfaceAssembly** (*p.209*),
cmas.ontology.interfaces.InterfaceLocation (*p.219*),
cmas.ontology.interfaces.InterfaceSpecialisation (*p.224*),
cmas.ontology.interfaces.InterfaceTransport (*p.230*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Interface cmas.ontology.interfaces.Interface.Factory (JsonObject job) throws ExceptionType [static]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.interfaces.Interface.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.interfaces.Interface.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler `cmas.ontology.interfaces.InterfaceBase.getMessageHandler()` [inherited]

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

String `cmas.ontology.Entity.getName()` [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in `cmas.ontology.Reference` (*p.288*).

Referenced by `cmas.ontology.processplan.ProcessPlanStructuredText.run()`, and `cmas.ontology.variables.VariableArray.setName()`.

EntityType `cmas.ontology.Entity.getType()` [inherited]

Returns

entity type.

void `cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message)` [inherited]

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

JsonObject `cmas.ontology.Entity.Serialize()` [inherited]

Reimplemented from `cmas.ontology.Container` (*p.161*).

void `cmas.ontology.Entity.setDescription (String Description)` [inherited]

Description is a way for the user to document the entity.

Parameters

<code>Description</code>	a user defined description
--------------------------	----------------------------

void `cmas.ontology.Entity.setID()` [inherited]

`setID` will set a new ID by generating a world" unique identifier.

Referenced by `Entity()`.

void `cmas.ontology.Entity.setID (String ID)` [inherited]

Will set a specific ID.

Parameters

<code>ID</code>	the new ID value.
-----------------	-------------------

void `cmas.ontology.Entity.setName (String Name)` [inherited]

set entity name.

Parameters

<code>Name</code>	the new name.
-------------------	---------------

Reimplemented in `cmas.ontology.Reference` (*p.289*), and `cmas.ontology.variables.VariableArray` (*p.339*).

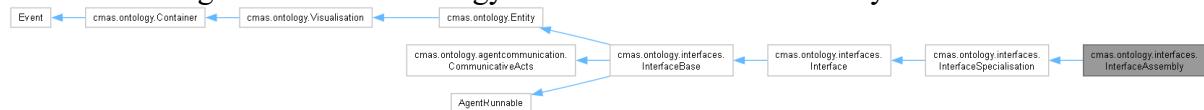
Referenced by [Entity\(\)](#).

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

- [Interface.java](#)

cmas.ontology.interfaces.InterfaceAssembly Class Reference

Inheritance diagram for cmas.ontology.interfaces.InterfaceAssembly:



Public Member Functions

- **InterfaceAssembly (InterfaceSpecialisation interf, boolean instance)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **String getSpecialisationName ()**
- **void setSpecialisationName (String specialisationName)**
- **String getBookedByName ()**
- **Boolean isBusy ()**
- **Boolean isBooked ()**
- **void setBooked (CommunicativeActs agent)**
- **CommunicativeActs getBooked ()**
- **Boolean isAttachedTo (InterfaceBase remoteInterface)**
- **Boolean isAttached ()**
- **String getAttachedTo ()**
- **Attachment getAttachment ()**
- **VariableMatchingType getVariableMatching ()**
- **ACLMessage getOnGoingCommunication ()**
- **void setVariableMatching (VariableMatchingType variableRequest)**
- **void initInterface (Agent agent)**
- **void AttachTo (InterfaceBase remote) throws ExceptionRuntime**
- **void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime**
- **void Detach () throws ExceptionRuntime**
- **void _Detach ()**
- **void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessage message)**
- **void run ()**
- **String getRequirements (Entity skill, Agent callingAgent, InterfaceBase callingInterface) throws ExceptionRuntime**
- **boolean book (InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean book (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean unbook (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface) throws InterruptedException**
- **Agent getRuntimeAgent ()**
- **synchronized String getActions ()**
- **void compile () throws ExceptionParameters, Exception**
- **synchronized void setActions (String actions)**
- **Action getActionObject ()**
- **MessageHandler getMessageHandler ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**

- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- Entity**Type** **getType** ()
- Properties **getProperties** ()
- Property<?> **getProperty** (String Name)
- Property<?> **getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Interface **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity cmas.ontology.interfaces.InterfaceAssembly.**copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.interfaces.InterfaceSpecialisation** (*p.224*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

**void cmas.ontology.Entity.Deserialize (JsonObject job) throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.Container** (*p.161*).

**Interface cmas.ontology.interfaces.Interface.Factory (JsonObject job) throws
ExceptionType [static], [inherited]**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.interfaces.InterfaceAssembly.getEntityDescription ()

Reimplemented from **cmas.ontology.interfaces.InterfaceSpecialisation** (*p.223*).

**String cmas.ontology.interfaces.InterfaceSpecialisation.getEntityIconFile
() [inherited]**

Reimplemented from **cmas.ontology.interfaces.Interface** (*p.203*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

**MessageHandler cmas.ontology.interfaces.InterfaceBase.getMessageHandler
() [inherited]**

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

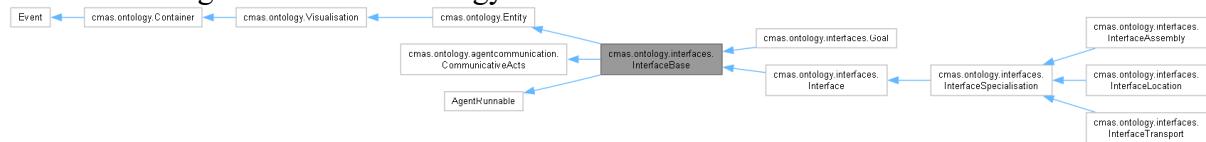
Referenced by **Entity()**.

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

- InterfaceAssembly.java

cmas.ontology.interfaces.InterfaceBase Class Reference

Inheritance diagram for cmas.ontology.interfaces.InterfaceBase:



Classes

- enum `InterfaceStateEnum` `VariableMatchingType`

Public Member Functions

- `InterfaceBase (EntityType type)`
- `InterfaceBase (InterfaceBase interf, boolean instance)`
- `String getBookedByName ()`
- `Boolean isBusy ()`
- `Boolean isBooked ()`
- `void setBooked (CommunicativeActs agent)`
- `CommunicativeActs getBooked ()`
- `Boolean isAttachedTo (InterfaceBase remoteInterface)`
- `Boolean isAttached ()`
- `String getAttachedTo ()`
- `Attachment getAttachment ()`
- `VariableMatchingType getVariableMatching ()`
- `ACLMessage getOnGoingCommunication ()`
- `void setVariableMatching (VariableMatchingType variableRequest)`
- `void initInterface (Agent agent)`
- `void AttachTo (InterfaceBase remote) throws ExceptionRuntime`
- `void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime`
- `void Detach () throws ExceptionRuntime`
- `void _Detach ()`
- `void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime`
- `Entity getCommunicative ()`
- `void postMessage (ACLMessage message)`
- `void run ()`
- `String getRequirements (Entity skill, Agent callingAgent, InterfaceBase callingInterface) throws ExceptionRuntime`
- `boolean book (InterfaceBase remoteInterface, ArrayList<String> desiredSkills, ArrayList<String> desiredVariables) throws InterruptedException`
- `boolean book (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface, ArrayList<String> desiredSkills, ArrayList<String> desiredVariables) throws InterruptedException`
- `boolean unbook (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface) throws InterruptedException`
- `Agent getRuntimeAgent ()`
- synchronized `String getActions ()`
- `void compile () throws ExceptionParameters, Exception`
- synchronized `void setActions (String actions)`
- `Action getActionObject ()`
- `MessageHandler getMessageHandler ()`
- abstract `Entity copy (boolean instance)`
- `void delete ()`
- `String toString ()`
- `String getID ()`
- `void setID (String ID)`
- `void setID ()`
- `String getName ()`

- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static Entity **Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- **Properties properties**
- **Relations relations**

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract], [inherited]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
Reimplemented in cmas.ontology.agents.Material	cmas.ontology.agents.DeploymentAgent (p.174), cmas.ontology.agents.Part (p.255),
cmas.ontology.agents.Resource	cmas.ontology.batch.Batch (p.151),
cmas.ontology.Executable	cmas.ontology.Hazard (p.199),
cmas.ontology.interfaces.Goal	cmas.ontology.interfaces.Interface (p.204),
cmas.ontology.interfaces.InterfaceAssembly	(p.209),
cmas.ontology.interfaces.InterfaceLocation	(p.219),
cmas.ontology.interfaces.InterfaceSpecialisation	(p.224),
cmas.ontology.interfaces.InterfaceTransport	(p.230),
cmas.ontology.processplan.ProcessPlanGoals	(p.266),
cmas.ontology.processplan.ProcessPlanJava	(p.272),
cmas.ontology.processplan.ProcessPlanStructuredText	(p.278),
cmas.ontology.processplan.SkillCall (p.308),	cmas.ontology.Reference (p.287),
cmas.ontology.variables.Variable (p.333),	cmas.ontology.variables.VariableArray (p.337),
cmas.ontology.variables.VariableBoolean (p.343),	cmas.ontology.variables.VariableInteger (p.349),
	cmas.ontology.variables.VariableObject (p.356),
cmas.ontology.variables.VariableProperty (p.362),	cmas.ontology.variables.VariableReal (p.368),
	cmas.ontology.variables.VariableString (p.374),
cmas.ontology.workflow.Requirement (p.293),	and cmas.ontology.workflow.WorkFlowLog (p.386).

References **Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

void cmas.ontology.Entity.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative ()

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (p.155).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.interfaces.InterfaceBase.getMessageHandler ()

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

void cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message)

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

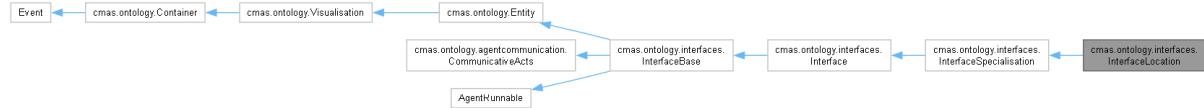
Referenced by **Entity()**.

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

- InterfaceBase.java

cmas.ontology.interfaces.InterfaceLocation Class Reference

Inheritance diagram for cmas.ontology.interfaces.InterfaceLocation:



Public Member Functions

- **InterfaceLocation (InterfaceSpecialisation interf, boolean instance)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **String getSpecialisationName ()**
- **void setSpecialisationName (String specialisationName)**
- **String getBookedByName ()**
- **Boolean isBusy ()**
- **Boolean isBooked ()**
- **void setBooked (CommunicativeActs agent)**
- **CommunicativeActs getBooked ()**
- **Boolean isAttachedTo (InterfaceBase remoteInterface)**
- **Boolean isAttached ()**
- **String getAttachedTo ()**
- **Attachment getAttachment ()**
- **VariableMatchingType getVariableMatching ()**
- **ACLMessage getOnGoingCommunication ()**
- **void setVariableMatching (VariableMatchingType variableRequest)**
- **void initInterface (Agent agent)**
- **void AttachTo (InterfaceBase remote) throws ExceptionRuntime**
- **void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime**
- **void Detach () throws ExceptionRuntime**
- **void _Detach ()**
- **void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessage message)**
- **void run ()**
- **String getRequirements (Entity skill, Agent callingAgent, InterfaceBase callingInterface) throws ExceptionRuntime**
- **boolean book (InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean book (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean unbook (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface) throws InterruptedException**
- **Agent getRuntimeAgent ()**
- **synchronized String getActions ()**
- **void compile () throws ExceptionParameters, Exception**
- **synchronized void setActions (String actions)**
- **Action getActionObject ()**
- **MessageHandler getMessageHandler ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**

- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- Entity**Type** **getType** ()
- Properties **getProperties** ()
- Property<?> **getProperty** (String Name)
- Property<?> **getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassNames** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Interface **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity cmas.ontology.interfaces.InterfaceLocation.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.interfaces.InterfaceSpecialisation** (*p.224*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

**void cmas.ontology.Entity.Deserialize (JsonObject job) throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.Container** (*p.161*).

**Interface cmas.ontology.interfaces.Interface.Factory (JsonObject job) throws
ExceptionType [static], [inherited]**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.interfaces.InterfaceLocation.getEntityDescription ()

Reimplemented from **cmas.ontology.interfaces.InterfaceSpecialisation** (*p.223*).

**String cmas.ontology.interfaces.InterfaceSpecialisation.getEntityIconFile
() [inherited]**

Reimplemented from **cmas.ontology.interfaces.Interface** (*p.203*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

**MessageHandler cmas.ontology.interfaces.InterfaceBase.getMessageHandler
() [inherited]**

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

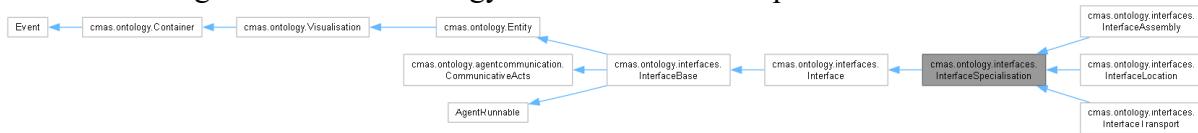
Referenced by **Entity()**.

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

- InterfaceLocation.java

cmas.ontology.interfaces.InterfaceSpecialisation Class Reference

Inheritance diagram for cmas.ontology.interfaces.InterfaceSpecialisation:



Public Member Functions

- **InterfaceSpecialisation (InterfaceSpecialisation interf, boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **String getSpecialisationName ()**
- **void setSpecialisationName (String specialisationName)**
- **Entity copy (boolean instance)**
- **String getBookedByName ()**
- **Boolean isBusy ()**
- **Boolean isBooked ()**
- **void setBooked (CommunicativeActs agent)**
- **CommunicativeActs getBooked ()**
- **Boolean isAttachedTo (InterfaceBase remoteInterface)**
- **Boolean isAttached ()**
- **String getAttachedTo ()**
- **Attachment getAttachment ()**
- **VariableMatchingType getVariableMatching ()**
- **ACLMessage getOnGoingCommunication ()**
- **void setVariableMatching (VariableMatchingType variableRequest)**
- **void initInterface (Agent agent)**
- **void AttachTo (InterfaceBase remote) throws ExceptionRuntime**
- **void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime**
- **void Detach () throws ExceptionRuntime**
- **void _Detach ()**
- **void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessage message)**
- **void run ()**
- **String getRequirements (Entity skill, Agent callingAgent, InterfaceBase callingInterface) throws ExceptionRuntime**
- **boolean book (InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean book (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean unbook (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface) throws InterruptedException**
- **Agent getRuntimeAgent ()**
- **synchronized String getActions ()**
- **void compile () throws ExceptionParameters, Exception**
- **synchronized void setActions (String actions)**
- **Action getActionObject ()**
- **MessageHandler getMessageHandler ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**

- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Interface **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- **Properties properties**
- **Relations relations**

Member Function Documentation

Entity **cmas.ontology.interfaces.InterfaceSpecialisation.copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.interfaces.Interface** (*p.204*).

Reimplemented in **cmas.ontology.interfaces.InterfaceAssembly** (*p.209*),
cmas.ontology.interfaces.InterfaceLocation (*p.219*),
cmas.ontology.interfaces.InterfaceTransport (*p.230*). and

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

**void cmas.ontology.Entity.Deserialize (JsonObject job) throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.Container** (*p.161*).

**Interface cmas.ontology.interfaces.Interface.Factory (JsonObject job) throws
ExceptionType [static], [inherited]**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]**Returns**

the entity description.

String cmas.ontology.interfaces.InterfaceSpecialisation.getEntityDescription ()

Reimplemented from **cmas.ontology.interfaces.Interface** (*p.203*).

String cmas.ontology.interfaces.InterfaceSpecialisation.getEntityIconFile ()

Reimplemented from **cmas.ontology.interfaces.Interface** (*p.203*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler `cmas.ontology.interfaces.InterfaceBase.getMessageHandler()` [inherited]

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

String `cmas.ontology.Entity.getName()` [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in `cmas.ontology.Reference` (*p.288*).

Referenced by `cmas.ontology.processplan.ProcessPlanStructuredText.run()`, and `cmas.ontology.variables.VariableArray.setName()`.

EntityType `cmas.ontology.Entity.getType()` [inherited]

Returns

entity type.

void `cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message)` [inherited]

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (*p.155*).

JsonObject `cmas.ontology.Entity.Serialize()` [inherited]

Reimplemented from `cmas.ontology.Container` (*p.161*).

void `cmas.ontology.Entity.setDescription (String Description)` [inherited]

Description is a way for the user to document the entity.

Parameters

<code>Description</code>	a user defined description
--------------------------	----------------------------

void `cmas.ontology.Entity.setID()` [inherited]

`setID` will set a new ID by generating a world" unique identifier.

Referenced by `Entity()`.

void `cmas.ontology.Entity.setID (String ID)` [inherited]

Will set a specific ID.

Parameters

<code>ID</code>	the new ID value.
-----------------	-------------------

void `cmas.ontology.Entity.setName (String Name)` [inherited]

set entity name.

Parameters

<code>Name</code>	the new name.
-------------------	---------------

Reimplemented in `cmas.ontology.Reference` (*p.289*), and `cmas.ontology.variables.VariableArray` (*p.339*).

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

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

- [InterfaceSpecialisation.java](#)

cmas.ontology.interfaces.InterfaceBase.InterfaceState Enum Reference

Public Attributes

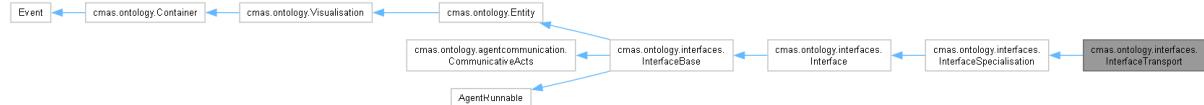
- **IDLE**
- **BUSY**

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

- InterfaceBase.java

cmas.ontology.interfaces.InterfaceTransport Class Reference

Inheritance diagram for cmas.ontology.interfaces.InterfaceTransport:



Public Member Functions

- **InterfaceTransport (InterfaceSpecialisation interf, boolean instance)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **String getSpecialisationName ()**
- **void setSpecialisationName (String specialisationName)**
- **String getBookedByName ()**
- **Boolean isBusy ()**
- **Boolean isBooked ()**
- **void setBooked (CommunicativeActs agent)**
- **CommunicativeActs getBooked ()**
- **Boolean isAttachedTo (InterfaceBase remoteInterface)**
- **Boolean isAttached ()**
- **String getAttachedTo ()**
- **Attachment getAttachment ()**
- **VariableMatchingType getVariableMatching ()**
- **ACLMessage getOnGoingCommunication ()**
- **void setVariableMatching (VariableMatchingType variableRequest)**
- **void initInterface (Agent agent)**
- **void AttachTo (InterfaceBase remote) throws ExceptionRuntime**
- **void _AttachTo (InterfaceBase remote, boolean source) throws ExceptionRuntime**
- **void Detach () throws ExceptionRuntime**
- **void _Detach ()**
- **void Transfer (InterfaceBase remoteInterface) throws ExceptionRuntime**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessage message)**
- **void run ()**
- **String getRequirements (Entity skill, Agent callingAgent, InterfaceBase callingInterface) throws ExceptionRuntime**
- **boolean book (InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean book (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface, ArrayList< String > desiredSkills, ArrayList< String > desiredVariables) throws InterruptedException**
- **boolean unbook (CommunicativeActs delegateAnswer, InterfaceBase remoteInterface) throws InterruptedException**
- **Agent getRuntimeAgent ()**
- **synchronized String getActions ()**
- **void compile () throws ExceptionParameters, Exception**
- **synchronized void setActions (String actions)**
- **Action getActionObject ()**
- **MessageHandler getMessageHandler ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**

- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- Entity**Type** **getType** ()
- Properties **getProperties** ()
- Property<?> **getProperty** (String Name)
- Property<?> **getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassNames** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Interface **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity cmas.ontology.interfaces.InterfaceTransport.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.interfaces.InterfaceSpecialisation** (*p.224*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

**void cmas.ontology.Entity.Deserialize (JsonObject job) throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.Container** (*p.161*).

**Interface cmas.ontology.interfaces.Interface.Factory (JsonObject job) throws
ExceptionType [static], [inherited]**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.interfaces.InterfaceBase.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.interfaces.InterfaceTransport.getEntityDescription ()

Reimplemented from **cmas.ontology.interfaces.InterfaceSpecialisation** (*p.223*).

**String cmas.ontology.interfaces.InterfaceSpecialisation.getEntityIconFile
 () [inherited]**

Reimplemented from **cmas.ontology.interfaces.Interface** (*p.203*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

**MessageHandler cmas.ontology.interfaces.InterfaceBase.getMessageHandler
 () [inherited]**

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.interfaces.InterfaceBase.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

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

- InterfaceTransport.java

cmas.ontology.variables.Iterator Class Reference

Public Member Functions

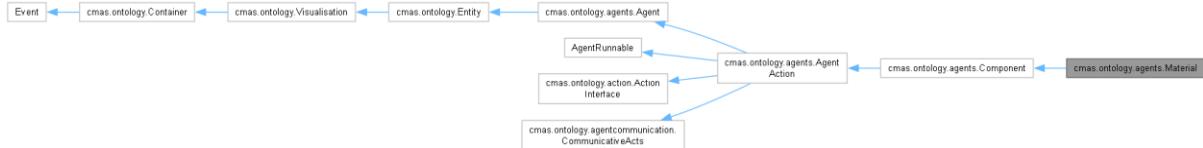
- **Iterator (Variable variable, int index)**
- boolean **hasNext ()**
- Object **getNext ()** throws **ExceptionVariable, ExceptionType, ExceptionConstant**
- int **increaseIndex ()**
- int **getIndex ()**
- void **setIndex (int index)**
- Variable **getVariable ()**
- void **setVariable (Variable variable)**
- Object **getProperty ()**
- void **setProperty (Object property)**
- Variable **getIteratorObject ()**

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

- Iterator.java

cmas.ontology.agents.Material Class Reference

Inheritance diagram for cmas.ontology.agents.Material:



Public Member Functions

- **Material (Material entity, boolean instance)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **synchronized String getActions ()**
- **synchronized void setActions (String actions)**
- **synchronized void setState (AgentState newState)**
- **synchronized AgentState getState ()**
- **void DeSerialize (JsonObject job) throws ExceptionType**
- **JsonObject Serialize ()**
- **synchronized void setPaused (boolean value)**
- **synchronized boolean getPaused ()**
- **synchronized void postMessage (ACLMensaje message)**
- **Entity getCommunicative ()**
- **void onInit ()**
- **void onDepot ()**
- **void onDeploy ()**
- **void onRun ()**
- **void Running ()**
- **void onUndeploy ()**
- **void onFinished ()**
- **void onFailed ()**
- **void onStop ()**
- **void onMessage (ACLMensaje message)**
- **void run ()**
- **MessageHandler getMessageHandler ()**
- **void gotoLine (Integer line)**
- **void stopEditor ()**
- **void compile () throws ExceptionParameters, Exception**
- **String getPhysicalID ()**
- **void setPhysicalID (String physicalID)**
- **String getAgentSpecificType ()**
- **void setAgentSpecificType (String agentSpecificType)**
- **AgentBaseTypes getAgentBaseType ()**
- **void addAbstractInterface (AbstractInterface abstractInterface, String Name)**
- **void clearAbstractInterfaces ()**
- **AbstractInterface getAbstractInterface (String Name)**
- **RuntimeTree getRuntimeTree ()**
- **void setRuntimeTree (RuntimeTree runtimeTree)**
- **Domain getRuntimeDomain ()**
- **void setRuntimeDomain (Domain runtimeDomain)**
- **LinkedBlockingQueue< WorkFlowLog > getHistory ()**
- **void setHistory (LinkedBlockingQueue< WorkFlowLog > p)**
- **void addToHistory (WorkFlowLog log)**
- **Boolean getAutoDeploy ()**
- **void setAutoDeploy (Boolean autoDeploy)**
- **void delete ()**
- **String toString ()**

- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- **Entity findRelationByName** (String RelationName, String EntityName)
- **Entity findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static AgentBaseTypes **getAgentType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- Action **action**
- MessageHandler **messageHandler**
- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity **cmas.ontology.agents.Material.copy (boolean instance)**

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void **cmas.ontology.Entity.delete () [inherited]**

Reimplemented from **cmas.ontology.Container** (*p.161*).

void **cmas.ontology.agents.AgentAction.DeSerialize (JsonObject job) throws ExceptionType [inherited]**

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

Entity **cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static], [inherited]**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity **cmas.ontology.agents.AgentAction.getCommunicative () [inherited]**

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String **cmas.ontology.Entity.getDescription () [inherited]**

Returns

the entity description.

String **cmas.ontology.agents.Material.getEntityDescription ()**

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String **cmas.ontology.agents.Material.getEntityIconFile ()**

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String **cmas.ontology.Entity.getID () [inherited]**

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.agents.AgentAction.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

void cmas.ontology.agents.AgentAction.gotoLine (Integer line) [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

synchronized void cmas.ontology.agents.AgentAction.postMessage (ACLMessages message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.agents.AgentAction.Serialize () [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.agents.AgentAction.stopEditor () [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

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

- Material.java

cmas.ontology.agentcommunication.ACL.MessageHandler Class Reference

Inheritance diagram for cmas.ontology.agentcommunication.ACL.MessageHandler:



Public Member Functions

- void **addMessage** (ACLMessage newMessage)
- ACLMessage **getNextMessage** ()
- ACLMessage **getNextMessage** (long timeout) throws InterruptedException
- boolean **isEmpty** ()

Static Package Attributes

- static final long **INTERRUPT_RESOLUTION** =500L

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

- `MessageHandler.java`

cmas.ontology.adapter.Modbus.ModbusClient.MODBUS_TYPE Enum Reference

Public Attributes

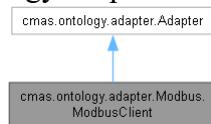
- **TCP**
- **UDP**
- **SERIAL**

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

- ModbusClient.java

cmas.ontology.adapter.Modbus.ModbusClient Class Reference

Inheritance diagram for cmas.ontology.adapter.Modbus.ModbusClient:



Classes

- enum **ADDRESS_TYPE**enum **MODBUS_TYPE**

Public Member Functions

- **ModbusClient** (String Source)
- void **setSource** (String Source)
- void **connectToServer** ()
- SerialParameters **getSerialParameters** ()
- String **getServerHost** ()
- void **setServerHost** (String serverHost)
- int **getServerPort** ()
- void **setServerPort** (int serverPort)
- **MODBUS_TYPE** **getServerType** ()
- void **setServerType** (**MODBUS_TYPE** serverType)
- String **ReadData** (String Address, String Name, Object **Value**) throws Exception
- void **WriteData** (String Address, String Name, Boolean bValue) throws Exception
- void **WriteData** (String Address, String Name, String sValue) throws Exception
- void **WriteData** (String Address, String Name, Integer iValue) throws Exception
- void **WriteData** (String Address, String Name, Double dValue) throws Exception
- boolean **close** ()
- void **setDebug** (boolean bValue)
- String **getDefaultValue** ()
- String **getDescription** ()
- boolean **isValid** (String Source)
- **Adapter** **createNew** (String Source)

Static Package Attributes

- static final int **DEFAULT_ID** =1

Member Function Documentation

boolean cmas.ontology.adapter.Modbus.ModbusClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.Modbus.ModbusClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.Modbus.ModbusClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.Modbus.ModbusClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.Modbus.ModbusClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.Modbus.ModbusClient.ReadData (String Address, String Name, Object Value) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.Modbus.ModbusClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.Modbus.ModbusClient.WriteData (String Address, String Name, Boolean bValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.Modbus.ModbusClient.WriteData (String Address, String Name, Double dValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.Modbus.ModbusClient.WriteData (String Address, String Name, Integer iValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.Modbus.ModbusClient.WriteData (String Address, String Name, String sValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- ModbusClient.java

cmas.ontology.Ontology Class Reference

Classes

- class **OntologyObject** class **OntologyRelation**

Static Public Member Functions

- static void **Read** (String fileName)

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

- Ontology.java

cmas.ontology.Ontology.OntologyObject Class Reference

Package Attributes

- String **Name**
- String **Extends**
- String **Category**
- String **Description**
- List<?> **Property**
- List<?> **OntologyRelation**

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

- Ontology.java

cmas.ontology.Ontology.OntologyRelation Class Reference

Package Attributes

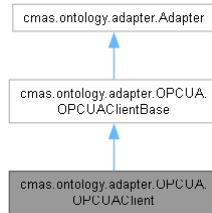
- String **Name**
- List< String > **AcceptedObjects**

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

- Ontology.java

cmas.ontology.adapter.OPCUA.OPCUAClient Class Reference

Inheritance diagram for cmas.ontology.adapter.OPCUA.OPCUAClient:



Public Member Functions

- **OPCUAClient (String Source)**
- **void connectToServer (String Address, String Name)**
- **String ReadData (String Address, String Name, Object Value)**
- **void WriteData (String Address, String Name, String Value)**
- **void WriteData (String Address, String Name, Double Value)**
- **void WriteData (String Address, String Name, Integer Value)**
- **void WriteData (String Address, String Name, Boolean Value)**
- **boolean close ()**
- **String getDefaultValue ()**
- **String getDescription ()**
- **boolean isValid (String Source)**
- **Adapter createNew (String Source)**
- **OpcUaClient createClient () throws InterruptedException, ExecutionException, UaException**
- **String browseNode (OpcUaClient client, NodeId browseRoot, String NodeName, String OutPutId)**
- **boolean gelAllNodes (TreeItem< OPCUANode > rootNode, OpcUaClient client, NodeId browseRoot)**
- **void setSource (String Source)**
- **String getServerEndpointUrl ()**
- **String getServerName ()**
- **void setServerName (String ServerName)**
- **String getServerPort ()**
- **void setServerPort (String serverPort)**
- **String getServerHost ()**
- **void setServerHost (String serverHost)**
- **boolean getDebug ()**
- **void setDebug (boolean bValue)**

Member Function Documentation

boolean cmas.ontology.adapter.OPCUA.OPCUAClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.OPCUA.OPCUAClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.OPCUA.OPCUAClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.OPCUA.OPCUAClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.OPCUA.OPCUAClientBase.getServerHost () [inherited]

Returns

the serverHost

String cmas.ontology.adapter.OPCUA.OPCUAClientBase.getServerName () [inherited]

Returns

the serverName

String cmas.ontology.adapter.OPCUA.OPCUAClientBase.getServerPort () [inherited]

Returns

the serverPort

boolean cmas.ontology.adapter.OPCUA.OPCUAClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.OPCUA.OPCUAClient.ReadData (String Address, String Name, Object Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setDebug (boolean bValue) [inherited]

Parameters

<i>bValue</i>	turn adapter debug on or off
---------------	------------------------------

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setServerHost (String serverHost) [inherited]

Parameters

<i>serverHost</i>	the serverHost to set
-------------------	-----------------------

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setServerName (String ServerName) [inherited]

Parameters

<i>ServerName</i>	the server name to set
-------------------	------------------------

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setServerPort (String serverPort) [inherited]

Parameters

<i>serverPort</i>	the serverPort to set
-------------------	-----------------------

void cmas.ontology.adapter.OPCUA.OPCUAClient.WriteData (String Address, String Name, Boolean Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.OPCUA.OPCUAClient.WriteData (String Address, String Name, Double Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.OPCUA.OPCUAClient.WriteData (String Address, String Name, Integer Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.OPCUA.OPCUAClient.WriteData (String Address, String Name, String Value)

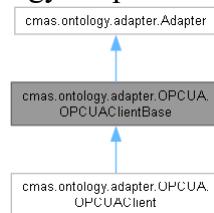
Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- OPCUAClient.java

cmas.ontology.adapter.OPCUA.OPCUAClientBase Class Reference

Inheritance diagram for cmas.ontology.adapter.OPCUA.OPCUAClientBase:



Public Member Functions

- OpcUaClient **createClient** () throws InterruptedException, ExecutionException, UaException
- String **browseNode** (OpcUaClient client, NodeId browseRoot, String nodeName, String OutPutId)
- boolean **gelAllNodes** (TreeItem< OPCUANode > rootNode, OpcUaClient client, NodeId browseRoot)
- void **setSource** (String Source)
- String **getServerEndpointUrl** ()
- String **getServerName** ()
- void **setServerName** (String ServerName)
- String **getServerPort** ()
- void **setServerPort** (String serverPort)
- String **getServerHost** ()
- void **setServerHost** (String serverHost)
- boolean **getDebug** ()
- void **setDebug** (boolean bValue)
- abstract boolean **close** ()
- abstract String **ReadData** (String Address, String Name, Object Value) throws Exception
- abstract void **WriteData** (String Address, String Name, Boolean bValue) throws Exception
- abstract void **WriteData** (String Address, String Name, String sValue) throws Exception
- abstract void **WriteData** (String Address, String Name, Integer iValue) throws Exception
- abstract void **WriteData** (String Address, String Name, Double dValue) throws Exception
- abstract String **getDefaultValue** ()
- abstract String **getDescription** ()
- abstract boolean **isValid** (String Source)
- abstract Adapter **createNew** (String Source)

Member Function Documentation

String cmas.ontology.adapter.OPCUA.OPCUAClientBase.getServerHost ()

Returns

the serverHost

String cmas.ontology.adapter.OPCUA.OPCUAClientBase.getServerName ()

Returns

the serverName

String cmas.ontology.adapter.OPCUA.OPCUAClientBase.getServerPort ()

Returns

the serverPort

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setDebug (boolean bValue)

Parameters

<i>bValue</i>	turn adapter debug on or off
---------------	------------------------------

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setServerHost (String serverHost)

Parameters

<i>serverHost</i>	the serverHost to set
-------------------	-----------------------

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setServerName (String ServerName)

Parameters

<i>ServerName</i>	the server name to set
-------------------	------------------------

void cmas.ontology.adapter.OPCUA.OPCUAClientBase.setServerPort (String serverPort)

Parameters

<i>serverPort</i>	the serverPort to set
-------------------	-----------------------

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

- OPCUAClientBase.java

cmas.ontology.adapter.OPCUA.OPCUANode Class Reference

Public Member Functions

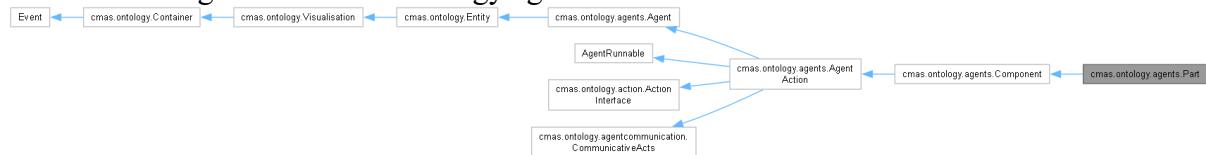
- **OPCUANode (Adapter adapter, ReferenceDescription rd)**
- **String getName ()**
- **String getDescription ()**
- **String getId ()**
- **String getDataType ()**
- **String getValue ()**

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

- OPCUANode.java

cmas.ontology.agents.Part Class Reference

Inheritance diagram for cmas.ontology.agents.Part:



Public Member Functions

- **Part** (Part entity, boolean instance)
- void **onUndeploy** ()
- void **onFailed** ()
- void **onStop** ()
- void **onFinished** ()
- void **onRun** ()
- Entity **copy** (boolean instance)
- BlockSequential **getSFC** ()
- void **setSFC** (BlockSequential sfc)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- Integer **getInstances** ()
- void **setInstances** (Integer counter)
- String **getEntityDescription** ()
- String **getEntityIconFile** ()
- synchronized String **getActions** ()
- synchronized void **setActions** (String actions)
- synchronized void **setState** (AgentState newState)
- synchronized AgentState **getState** ()
- JsonObject **Serialize** ()
- synchronized void **setPaused** (boolean value)
- synchronized boolean **getPaused** ()
- synchronized void **postMessage** (ACLMessages message)
- Entity **getCommunicative** ()
- void **onInit** ()
- void **onDepot** ()
- void **onDeploy** ()
- void **Running** ()
- void **onMessage** (ACLMessages message)
- void **run** ()
- MessageHandler **getMessageHandler** ()
- void **gotoLine** (Integer line)
- void **stopEditor** ()
- void **compile** () throws ExceptionParameters, Exception
- String **getPhysicalID** ()
- void **setPhysicalID** (String physicalID)
- String **getAgentSpecificType** ()
- void **setAgentSpecificType** (String agentSpecificType)
- AgentBaseTypes **getAgentBaseType** ()
- void **addAbstractInterface** (AbstractInterface abstractInterface, String Name)
- void **clearAbstractInterfaces** ()
- AbstractInterface **getAbstractInterface** (String Name)
- RuntimeTree **getRuntimeTree** ()
- void **setRuntimeTree** (RuntimeTree runtimeTree)
- Domain **getRuntimeDomain** ()
- void **setRuntimeDomain** (Domain runtimeDomain)
- LinkedBlockingQueue<**WorkFlowLog**> **getHistory** ()
- void **setHistory** (LinkedBlockingQueue<**WorkFlowLog**> p)
- void **addToHistory** (**WorkFlowLog** log)

- Boolean **getAutoDeploy** ()
- void **setAutoDeploy** (Boolean autoDeploy)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static AgentBaseTypes **getAgentType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- Action **action**
- MessageHandler **messageHandler**
- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity cmas.ontology.agents.Part.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.agents.Part.DeSerialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

Entity cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.agents.AgentAction.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.agents.Part.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.agents.Part.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.agents.AgentAction.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.agents.AgentAction.gotoLine (Integer line) [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

void cmas.ontology.agents.Part.onFailed ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.Part.onFinished ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.Part.onRun ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.Part.onStop ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

void cmas.ontology.agents.Part.onUndeploy ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

synchronized void cmas.ontology.agents.AgentAction.postMessage (ACLMessages message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

JsonObject cmas.ontology.agents.AgentAction.Serialize () [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.agents.AgentAction.stopEditor () [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

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

- Part.java

cmas.ontology.agentcommunication.ACL.ACMessage.Performative Enum Reference

Public Attributes

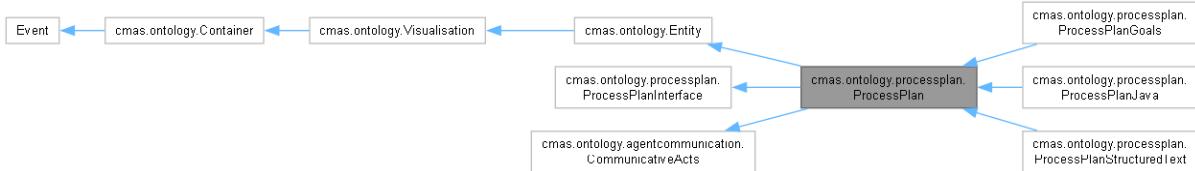
- REQUEST
- QUERY
- INFORM
- CANCEL
- PROGRESS
- REFUSE
- AGREE
- REPLAN
- FAILURE
- DONE
- RESULT

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

- ACMessage.java

cmas.ontology.processplan.ProcessPlan Class Reference

Inheritance diagram for cmas.ontology.processplan.ProcessPlan:



Classes

- enum **ProcessPlanReturnValue** enum **ProcessPlanType**

Public Member Functions

- **ProcessPlan (ProcessPlanType type)**
- **ProcessPlan (ProcessPlan processplan, boolean instance)**
- **void Deserialize (JsonObject job) throws ExceptionType**
- **JsonObject Serialize ()**
- **Integer getCost ()**
- **void setCost (Integer cost)**
- **ProcessPlanType getProcessPlanType ()**
- **WorkFlowTypes getPostAction ()**
- **void setPostAction (WorkFlowTypes action)**
- **void setProcessPlanType (ProcessPlanType processPlanType)**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessages message)**
- **MessageHandler getMessageHandler ()**
- **String getTargetType ()**
- **void setTargetType (String targetType)**
- **abstract Entity copy (boolean instance)**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRelations ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**
- **ArrayList< Entity > getSpecificRelationType (String Name)**
- **void addSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (String Name, Entity ent)**

- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass Name** ()
- void **setClass Name** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- abstract void **resetAllAbstractInterfaces** () throws ExceptionParser
- abstract int **onNegotiate** (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)
- abstract boolean **onOverlap** (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface) throws ExceptionParser
- abstract void **run** (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)
- abstract ProcessPlanReturnValue **execute** (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues) throws ExceptionParameters, ExceptionRuntime, ExceptionBase, ExceptionParser
- abstract void **abort** ()
- default void **loadClasspath** (URLClassLoader classLoader, String classpath)
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static ProcessPlanType **getProcessPlanType** (JsonObject job) throws **ExceptionType**
- static Entity Factory (JsonObject job) throws **ExceptionType**
- static Entity Factory (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract], [inherited]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
Reimplemented in cmas.ontology.agents.Material	cmas.ontology.agents.DeploymentAgent (p.237), cmas.ontology.agents.Part (p.255),
cmas.ontology.agents.Resource	cmas.ontology.batch.Batch (p.151),
cmas.ontology.Executable	cmas.ontology.Hazard (p.199),
cmas.ontology.interfaces.Goal	cmas.ontology.interfaces.Interface (p.204),

<code>cmas.ontology.interfaces.InterfaceAssembly</code>	(p.209),
<code>cmas.ontology.interfaces.InterfaceLocation</code>	(p.219),
<code>cmas.ontology.interfaces.InterfaceSpecialisation</code>	(p.224),
<code>cmas.ontology.interfaces.InterfaceTransport</code>	(p.230),
<code>cmas.ontology.processplan.ProcessPlanGoals</code>	(p.266),
<code>cmas.ontology.processplan.ProcessPlanJava</code>	(p.272),
<code>cmas.ontology.processplan.ProcessPlanStructuredText</code>	(p.278),
<code>cmas.ontology.processplan.SkillCall</code> (p.308), <code>cmas.ontology.Reference</code>	(p.287),
<code>cmas.ontology.variables.Variable</code> (p.333), <code>cmas.ontology.variables.VariableArray</code>	(p.337),
<code>cmas.ontology.variables.VariableBoolean</code> (p.343), <code>cmas.ontology.variables.VariableInteger</code> (p.349), <code>cmas.ontology.variables.VariableObject</code>	(p.356),
<code>cmas.ontology.variables.VariableProperty</code> (p.362), <code>cmas.ontology.variables.VariableReal</code> (p.368), <code>cmas.ontology.variables.VariableString</code>	(p.374),
<code>cmas.ontology.workflow.Requirement</code> (p.293), and <code>cmas.ontology.workflow.WorkFlowLog</code> (p.386).	

References `Entity()`.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from `cmas.ontology.Container` (p.161).

void cmas.ontology.processplan.ProcessPlan.Deserialize (JsonObject job) throws ExceptionType

Reimplemented from `cmas.ontology.Entity` (p.178).

Entity cmas.ontology.processplan.ProcessPlan.Factory (JsonObject job) throws ExceptionType [static]

Reimplemented from `cmas.ontology.Entity` (p.178).

Entity cmas.ontology.processplan.ProcessPlan.getCommunicative ()

Implements `cmas.ontology.agentcommunication.CommunicativeActs` (p.155).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.processplan.ProcessPlan.getMessageHandler ()

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.processplan.ProcessPlan.postMessage (ACLMensaje message)

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

abstract void cmas.ontology.processplan.ProcessPlanInterface.run (Agent agent, Entity target, ACLMensaje message, ContentParameter[] abstractValues) [abstract, inherited]

Implemented in **cmas.ontology.processplan.ProcessPlanStructuredText** (*p.280*).

JSONObject cmas.ontology.processplan.ProcessPlan.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

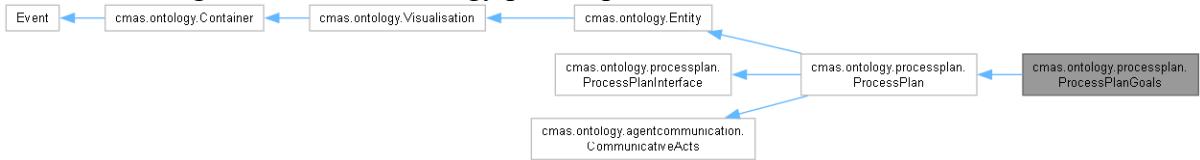
<i>Name</i>	the new name.
Reimplemented in cmas.ontology.Reference (p.289), and cmas.ontology.variables.VariableArray (p.339).	
Referenced by Entity() .	

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

- ProcessPlan.java

cmas.ontology.processplan.ProcessPlanGoals Class Reference

Inheritance diagram for cmas.ontology.processplan.ProcessPlanGoals:



Public Member Functions

- `ProcessPlanGoals (ArrayList< RelationObject > relation)`
- `ProcessPlanGoals (ProcessPlanGoals processplan, ArrayList< RelationObject > relation, boolean instance)`
- `void run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)`
- `ProcessPlanReturnValue execute (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues) throws ExceptionRuntime, ExceptionBase, ExceptionParser`
- `void abort ()`
- `Entity copy (boolean instance)`
- `Entity copy (ArrayList< RelationObject > relation, boolean instance)`
- `DebugResult BreakPointCallBack (Block globalBlock, Block localBlock, Statement statement, BreakPoint breakpoint)`
- `BlockSequential getSFC ()`
- `void setSFC (BlockSequential sfc)`
- `void setOwner (Entity owner)`
- `Entity getOwner ()`
- `void DeSerialize (JsonObject job) throws ExceptionType`
- `JsonObject Serialize ()`
- `int onNegotiate (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)`
- `boolean onOverlap (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface)`
- `void resetAllAbstractInterfaces ()`
- `String getEntityDescription ()`
- `String getEntityIconFile ()`
- `Integer getCost ()`
- `void setCost (Integer cost)`
- `ProcessPlanType getProcessPlanType ()`
- `WorkFlowTypes getPostAction ()`
- `void setPostAction (WorkFlowTypes action)`
- `void setProcessPlanType (ProcessPlanType processPlanType)`
- `Entity getCommunicative ()`
- `void postMessage (ACLMessage message)`
- `MessageHandler getMessageHandler ()`
- `String getTargetType ()`
- `void setTargetType (String targetType)`
- `void delete ()`
- `String toString ()`
- `String getID ()`
- `void setID (String ID)`
- `void setID ()`
- `String getName ()`
- `void setName (String Name)`
- `String getParentName ()`
- `void setParent (Entity ent)`

- `String getDescription ()`
- `void setDescription (String Description)`
- `String getInstanceID ()`
- `EntityType getType ()`
- `Properties getProperties ()`
- `Property<?> getProperty (String Name)`
- `Property<?> getPropertyIgnoreCase (String Name)`
- `void addProperty (Property<?> Item)`
- `boolean isCompatibleRelation (String Name, Entity ent)`
- `boolean hasRelation (String Name, Entity ent)`
- `LinkedHashMap< String, ArrayList< RelationObject > > getRealitions ()`
- `Collection< ArrayList< RelationObject > > getAllRelations ()`
- `ArrayList< RelationObject > getSpecificRelations (String Name)`
- `Entity findRelationByName (String RelationName, String EntityName)`
- `Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)`
- `ArrayList< Entity > getSpecificRelationType (String Name)`
- `void addSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (Entity ent)`
- `void eventReflection (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)`
- `ImageView getEntityIcon ()`
- `void copyProperties (Container container, boolean instance)`
- `void setOverride (Boolean value)`
- `Boolean getProtected ()`
- `void setProtected (Boolean protectedValue)`
- `Object getUserData ()`
- `void setUserData (Object userData)`
- `String getClassName ()`
- `void setClassName (String newName)`
- `Relations getRealitionsObject ()`
- `ArrayList< BreakPoint > getBreakPointList ()`
- `void setBreakPointList (ArrayList< BreakPoint > breakPointList)`
- `Boolean isInstance ()`
- `default void loadClasspath (URLClassLoader classLoader, String classpath)`
- `default ACLMessage waitForPerformative (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException`

Static Public Member Functions

- `static ProcessPlanType getProcessPlanType (JsonObject job) throws ExceptionType`
- `static Entity Factory (JsonObject job) throws ExceptionType`
- `static Entity Factory (String job) throws ExceptionType, ExceptionConstant, ExceptionVariable`

Protected Attributes

- `Properties properties`
- `Relations relations`

Member Function Documentation

void cmas.ontology.processplan.ProcessPlanGoals.abort ()

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

Entity cmas.ontology.processplan.ProcessPlanGoals.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.processplan.ProcessPlanGoals.DeSerialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.processplan.ProcessPlan** (*p.259*).

ProcessPlanReturnValue cmas.ontology.processplan.ProcessPlanGoals.execute (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues) throws ExceptionRuntime, ExceptionBase, ExceptionParser

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

Entity cmas.ontology.processplan.ProcessPlan.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.processplan.ProcessPlan.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]**Returns**

the entity description.

String cmas.ontology.processplan.ProcessPlanGoals.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.processplan.ProcessPlanGoals.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.processplan.ProcessPlan.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

int cmas.ontology.processplan.ProcessPlanGoals.onNegotiate (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

boolean cmas.ontology.processplan.ProcessPlanGoals.onOverlap (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

void cmas.ontology.processplan.ProcessPlan.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

void cmas.ontology.processplan.ProcessPlanGoals.resetAllAbstractInterfaces ()

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

void cmas.ontology.processplan.ProcessPlanGoals.run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

JsonObject cmas.ontology.processplan.ProcessPlanGoals.Serialize ()

Reimplemented from **cmas.ontology.processplan.ProcessPlan** (*p.259*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

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

- ProcessPlanGoals.java

cmas.ontology.processplan.ProcessPlanInterface Interface Reference

Inheritance diagram for cmas.ontology.processplan.ProcessPlanInterface:



Public Member Functions

- abstract void **resetAllAbstractInterfaces** () throws ExceptionParser
- abstract int **onNegotiate** (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)
- abstract boolean **onOverlap** (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface) throws ExceptionParser
- abstract void **run** (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)
- abstract **ProcessPlanReturnValue** **execute** (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues) throws ExceptionParameters, ExceptionRuntime, ExceptionBase, ExceptionParser
- abstract void **abort** ()
- default void **loadClasspath** (URLClassLoader classLoader, String classpath)

Member Function Documentation

abstract void cmas.ontology.processplan.ProcessPlanInterface.run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues) [abstract]

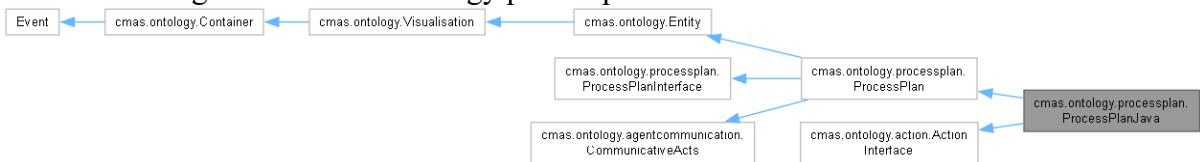
Implemented in **cmas.ontology.processplan.ProcessPlanStructuredText** (*p.280*).

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

- ProcessPlanInterface.java

cmas.ontology.processplan.ProcessPlanJava Class Reference

Inheritance diagram for cmas.ontology.processplan.ProcessPlanJava:



Public Member Functions

- **ProcessPlanJava (ProcessPlanJava processplan, boolean instance)**
- synchronized void **setFileName (String fileName)**
- synchronized String **getFileName ()**
- void **gotoLine (Integer line)**
- void **stopEditor ()**
- void **resetAllAbstractInterfaces ()** throws ExceptionParser
- int **onNegotiate (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)**
- boolean **onOverlap (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface)** throws ExceptionParser
- void **run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)**
- **ProcessPlanReturnValue execute (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues)**
- **ProcessPlanInterface getObject ()** throws ExceptionParser
- void **abort ()**
- **Entity copy (boolean instance)**
- String **getEntityDescription ()**
- String **getEntityIconFile ()**
- String **getDependencies ()**
- void **setDependencies (String dependencies)**
- void **DeSerialize (JsonObject job)** throws **ExceptionType**
- JsonObject **Serialize ()**
- Integer **getCost ()**
- void **setCost (Integer cost)**
- **ProcessPlanType getProcessPlanType ()**
- **WorkFlowTypes getPostAction ()**
- void **setPostAction (WorkFlowTypes action)**
- void **setProcessPlanType (ProcessPlanType processPlanType)**
- **Entity getCommunicative ()**
- void **postMessage (ACLMessage message)**
- **MessageHandler getMessageHandler ()**
- String **getTargetType ()**
- void **setTargetType (String targetType)**
- void **delete ()**
- String **toString ()**
- String **getID ()**
- void **setID (String ID)**
- void **setID ()**
- String **getName ()**
- void **setName (String Name)**
- String **getParentName ()**
- void **setParent (Entity ent)**
- String **getDescription ()**
- void **setDescription (String Description)**
- String **getInstanceID ()**

- **EntityType** **getType** ()
- **Properties** **getProperties** ()
- **Property<?>** **getProperty** (String Name)
- **Property<?>** **getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassname** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default void **loadClasspath** (URLClassLoader classLoader, String classpath)
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static ProcessPlanType **getProcessPlanType** (JsonObject job) throws **ExceptionType**
- static Entity Factory (JsonObject job) throws **ExceptionType**
- static Entity Factory (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Static Public Attributes

- static final boolean ENABLED = false

Protected Attributes

- Properties properties
- Relations relations

Member Function Documentation

void cmas.ontology.processplan.ProcessPlanJava.abort ()

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

Entity cmas.ontology.processplan.ProcessPlanJava.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.processplan.ProcessPlan.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

ProcessPlanReturnValue cmas.ontology.processplan.ProcessPlanJava.execute (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

Entity cmas.ontology.processplan.ProcessPlan.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.processplan.ProcessPlan.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]**Returns**

the entity description.

String cmas.ontology.processplan.ProcessPlanJava.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.processplan.ProcessPlanJava.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.processplan.ProcessPlan.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.processplan.ProcessPlanJava.gotoLine (Integer line)

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

int cmas.ontology.processplan.ProcessPlanJava.onNegotiate (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

boolean cmas.ontology.processplan.ProcessPlanJava.onOverlap (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface) throws ExceptionParser

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

void cmas.ontology.processplan.ProcessPlan.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

void cmas.ontology.processplan.ProcessPlanJava.resetAllAbstractInterfaces () throws ExceptionParser

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

void cmas.ontology.processplan.ProcessPlanJava.run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

JsonObject cmas.ontology.processplan.ProcessPlan.Serialize () [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.processplan.ProcessPlanJava.stopEditor ()

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

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

- ProcessPlanJava.java

cmas.ontology.processplan.ProcessPlan.ProcessPlanReturnValue Enum Reference

Public Member Functions

- int **toInt ()**

Static Public Member Functions

- static **ProcessPlanReturnValue fromInt (int val)**

Public Attributes

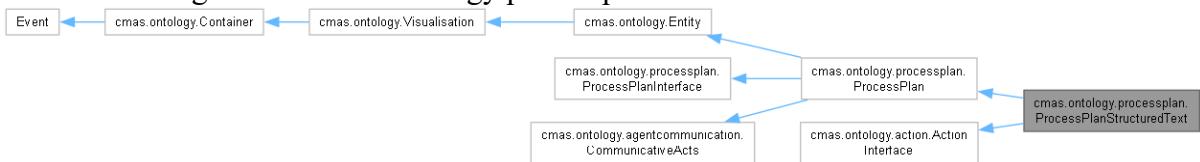
- **RUNNING** =(1)
- **DONE** =(2)
- **FAILED** =(3)
- **INTERRUPTED** =(4)
- **REPLAN** =(5)
- **REDO** =(6)

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

- ProcessPlan.java

cmas.ontology.processplan.ProcessPlanStructuredText Class Reference

Inheritance diagram for cmas.ontology.processplan.ProcessPlanStructuredText:



Public Member Functions

- **ProcessPlanStructuredText (ProcessPlanStructuredText processplan, boolean instance)**
- **Entity copy (boolean instance)**
- **int onNegotiate (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)**
- **boolean onOverlap (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface)**
- **void run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)**
- **ProcessPlanReturnValue execute (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues)**
- **void gotoLine (Integer line)**
- **void stopEditor ()**
- **void abort ()**
- **void DeSerialize (JsonObject job) throws ExceptionType**
- **JsonObject Serialize ()**
- **synchronized String getActions ()**
- **void compile (Agent agent, Entity target) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **void compile (Agent agent, Entity target, ArrayList<Variable> variables, ArrayList<AbstractInterface> abstractInterfaces) throws ExceptionRuntime, ExceptionParser, ExceptionBase, ExceptionExit**
- **void addVariable (VariableObject variableObject)**
- **synchronized void setActions (String actions)**
- **LinkedHashMap< String, AbstractInterface > getAllAbstractInterfaces ()**
- **void resetAllAbstractInterfaces ()**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **Integer getCost ()**
- **void setCost (Integer cost)**
- **ProcessPlanType getProcessPlanType ()**
- **WorkFlowTypes getPostAction ()**
- **void setPostAction (WorkFlowTypes action)**
- **void setProcessPlanType (ProcessPlanType processPlanType)**
- **Entity getCommunicative ()**
- **void postMessage (ACLMessage message)**
- **MessageHandler getMessageHandler ()**
- **String getTargetType ()**
- **void setTargetType (String targetType)**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**

- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- Entity**Type** **getType** ()
- Properties **getProperties** ()
- Property<?> **getProperty** (String Name)
- Property<?> **getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealtions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassname** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()
- default void **loadClasspath** (URLClassLoader classLoader, String classpath)
- default ACLMessage **waitForPerformative** (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException

Static Public Member Functions

- static ProcessPlanType **getProcessPlanType** (JsonObject job) throws **ExceptionType**
- static Entity Factory (JsonObject job) throws **ExceptionType**
- static Entity Factory (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties properties
- Relations relations

Member Function Documentation

void cmas.ontology.processplan.ProcessPlanStructuredText.abort ()

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

Entity cmas.ontology.processplan.ProcessPlanStructuredText.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.processplan.ProcessPlanStructuredText.DeSerialize (JsonObject job) throws ExceptionType

Reimplemented from **cmas.ontology.processplan.ProcessPlan** (*p.259*).

ProcessPlanReturnValue

cmas.ontology.processplan.ProcessPlanStructuredText.execute (Agent agent, Entity target, ACLMessage aclMessage, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

Entity cmas.ontology.processplan.ProcessPlan.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity cmas.ontology.processplan.ProcessPlan.getCommunicative () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getDescription () [inherited]**Returns**

the entity description.

String cmas.ontology.processplan.ProcessPlanStructuredText.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.processplan.ProcessPlanStructuredText.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.processplan.ProcessPlan.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.processplan.ProcessPlanStructuredText.gotoLine (Integer line)

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

int cmas.ontology.processplan.ProcessPlanStructuredText.onNegotiate (Agent agent, Entity target, boolean planning, ContentParameter[] abstractValues)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

boolean cmas.ontology.processplan.ProcessPlanStructuredText.onOverlap (Agent agent, Entity target, boolean planning, InterfaceBase remoteInterface)

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

void cmas.ontology.processplan.ProcessPlan.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

void cmas.ontology.processplan.ProcessPlanStructuredText.resetAllAbstractInterfaces ()

Implements **cmas.ontology.processplan.ProcessPlanInterface** (*p.269*).

void cmas.ontology.processplan.ProcessPlanStructuredText.run (Agent agent, Entity target, ACLMessage message, ContentParameter[] abstractValues)

Will run the process plan in a separate thread and does not wait for it to finish. The result will be sent to the remote caller and a DONE signal is sent back to our own local interface.

Implements **cmas.ontology.ProcessPlanInterface** (*p.269*).

References **cmas.ontology.Entity.getName()**, and **run()**.

Referenced by **run()**.

JsonObject cmas.ontology.processplan.ProcessPlanStructuredText.Serialize ()

Reimplemented from **cmas.ontology.ProcessPlan** (*p.259*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.processplan.ProcessPlanStructuredText.stopEditor ()

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

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

- ProcessPlanStructuredText.java

cmas.ontology.processplan.ProcessPlan.ProcessPlanType Enum Reference

Public Attributes

- STRUCTUREDTEXT
- SFC_GOALS
- JAVA

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

- ProcessPlan.java

cmas.ontology.Properties Class Reference

Public Member Functions

- **Properties** (**Properties** properties, boolean instance)
- **Properties** **copy** (boolean instance)
- Collection<**Property**<?>> **getPropertiesList** ()
- void **delete** ()
- void **clear** ()
- **Property**<?> **getProperty** (String Name)
- **Property**<?> **getPropertyIgnoreCase** (String Name)
- boolean **propertyExist** (**Property**<?> p)
- void **add** (**Property**<?> Item)
- void **remove** (String Name)
- void **setParent** (**Entity** ent)
- void **setOverride** (Boolean value)
- JSONArray **Serialize** ()

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

- Properties.java

cmas.ontology.Property< T > Class Template Reference

Inheritance diagram for cmas.ontology.Property< T >:



Classes

enum Specific.PropertyTypePublic Member Functions

- **Property** (String name)
- **Property** (String name, String classname, String description, Supplier< Object > getter, Consumer< T > setter, Function< String, Object > deserializer, Function< Object, String > serializer, Boolean readonly, Boolean caninherit)
- **Property** (String name, String classname, String description, Supplier< Object > getter, Consumer< T > setter, Function< String, Object > deserializer, Function< Object, String > serializer, Boolean readonly, Boolean caninherit, EnumSet< **Specific.PropertyType** > PropertyType)
- **Property** (**Property<?>** property, boolean instance)
- **Property** (JsonObject job) throws Exception
- **Property< T > copy** (boolean instance)
- void **delete** ()
- Entity **getParent** ()
- void **setParent** (Entity parent)
- String **getParentID** ()
- void **setParentID** (String ID)
- String **getName** ()
- void **setName** (String name)
- String **getDescription** ()
- void **setDescription** (String description)
- Object **getValue** ()
- void **setValue** (Object value)
- Object **getLocalValue** ()
- void **setLocalValue** (Object value)
- Boolean **getCanInherit** ()
- void **setCanInherit** (Boolean canInherit)
- Boolean **getOverride** ()
- void **setOverride** (Boolean overide)
- Supplier< Object > **getGetter** ()
- void **setGetter** (Supplier< Object > getter)
- Consumer< T > **getSetter** ()
- void **setSetter** (Consumer<?> setter)
- String **getClassName** ()
- void **setClassName** (String className)
- Boolean **getReadOnly** ()
- void **setReadOnly** (Boolean readOnly)
- boolean **isEditable** ()
- EnumSet< **Specific.PropertyType** > **getPropertyType** ()
- void **setPropertyType** (EnumSet< **Specific.PropertyType** > PropertyType)
- JsonObject **Serialize** ()
- void **DeSerialize** (JsonObject propObject, boolean loadAll)
- String **getHelpTopic** ()
- String **getHelpWord** ()

Static Public Member Functions

- static Boolean **dezBoolean** (String str)

- static Double **dezReal** (String str)
- static Integer **dezInteger** (String str)

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

- Property.java

cmas.ontology.variables.ValueRange.Range Class Reference

Public Member Functions

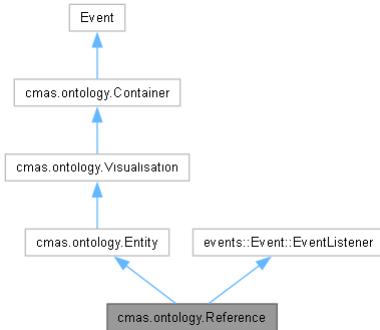
- **Range** (Number value)
- **Range (Range[] ranges)**
- boolean **Check** (Number testValue) throws **ExceptionType**
- boolean **CheckRanges** (Number testValue) throws **ExceptionType**
- boolean **CheckValues** (Number testValue) throws **ExceptionType**
- boolean **isNot** ()
- void **setNot** (boolean not)
- boolean **isGreaterThan** ()
- void **setGreaterThan** (boolean greaterThan)
- boolean **isLessThan** ()
- void **setLessThan** (boolean lessThan)

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

- ValueRange.java

cmas.ontology.Reference Class Reference

Inheritance diagram for cmas.ontology.Reference:



Public Member Functions

- `Reference (Entity owner, Entity referencedObject)`
- `Reference (Reference reference, boolean instance)`
- `Entity copy (boolean instance)`
- `void delete ()`
- `String getReferenceName ()`
- `String getName ()`
- `void setName (String Name)`
- `String getReferenceID ()`
- `Entity getObject ()`
- `void setObject (Entity object)`
- `Entity getOwner ()`
- `void setOwner (Entity owner)`
- `void onEvent (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object Owner)`
- `String getEntityDescription ()`
- `String getEntityIconFile ()`
- `String toString ()`
- `String getID ()`
- `void setID (String ID)`
- `void setID ()`
- `String getParentName ()`
- `void setParent (Entity ent)`
- `String getDescription ()`
- `void setDescription (String Description)`
- `String getInstanceID ()`
- `EntityType getType ()`
- `Properties getProperties ()`
- `Property<?> getProperty (String Name)`
- `Property<?> getPropertyIgnoreCase (String Name)`
- `void addProperty (Property<?> Item)`
- `boolean isCompatibleRelation (String Name, Entity ent)`
- `boolean hasRelation (String Name, Entity ent)`
- `LinkedHashMap< String, ArrayList< RelationObject > > getRealitions ()`
- `Collection< ArrayList< RelationObject > > getAllRelations ()`
- `ArrayList< RelationObject > getSpecificRelations (String Name)`
- `Entity findRelationByName (String RelationName, String EntityName)`
- `Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)`
- `ArrayList< Entity > getSpecificRelationType (String Name)`
- `void addSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (Entity ent)`

- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassname** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static Entity **Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
 - Relations **relations**
-

Member Function Documentation

Entity **cmas.ontology.Reference.copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (p.382).

References **cmas.ontology.Entity.Entity()**.

void **cmas.ontology.Reference.delete** ()

Reimplemented from **cmas.ontology.Entity** (p.178).

void **cmas.ontology.Entity.DeSerialize** (JsonObject job) throws **ExceptionType** [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

String **cmas.ontology.Entity.getDescription** () [inherited]

Returns

the entity description.

String cmas.ontology.Reference.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Reference.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Reference.getName ()

get current name of the entity.

Returns

current name.

Reimplemented from **cmas.ontology.Entity** (*p.387*).

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Reference.setName (String Name)

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

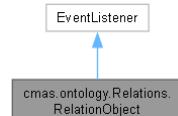
Reimplemented from **cmas.ontology.Entity** (*p.388*).

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

- Reference.java

cmas.ontology.Relations.RelationObject Class Reference

Inheritance diagram for cmas.ontology.Relations.RelationObject:



Public Member Functions

- **RelationObject (Entity relationentity, Entity owner)**
- **RelationObject (RelationObject relationobject)**
- **RelationObject copy ()**
- **void delete ()**
- **void setOwner (Entity entity)**
- **Entity getOwner ()**
- **Entity getObject ()**
- **void onEvent (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)**
- **JsonObject Serialize ()**

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

- Relations.java

cmas.ontology.Relations Class Reference

Classes

class RelationObjectPublic Member Functions

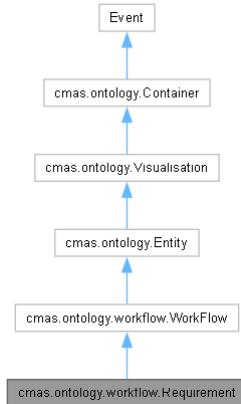
- **Relations** (**Relations** relations, **Container** owner, boolean instance)
- **Relations copy** (**Container** owner, boolean instance)
- void **delete** ()
- LinkedHashMap< String, ArrayList< **RelationObject** >> **getRelations** ()
- Collection< ArrayList< **RelationObject** >> **getRelationList** ()
- ArrayList< **RelationObject** > **getRelation** (String Name)
- ArrayList< **Entity** > **getRelationEntities** (String Name)
- **Entity getSpecificRelationEntity** (String RelationName, String Name)
- void **createRelation** (String Name, ArrayList<? extends **Entity** > AcceptedObjects)
- void **createRelation** (String Name, ArrayList< **RelationObject** > relation, boolean notused)
- void **addRelationType** (String Name, **Entity** e)
- ArrayList< **Entity** > **getRelationTypes** (String Name)
- void **addRelations** (String Name, ArrayList<? extends **Entity** > ve, **Entity** Owner)
- void **addItem** (String Name, **Entity** Item, **Entity** Owner)
- boolean **removeItem** (String Name, **Entity** Item)
- boolean **isCompatible** (String Name, **Entity** ent)
- String **getRealtionName** (**Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- void **setParent** (String Name, **Entity** ent)
- JSONArray **Serialize** ()
- void **DeSerialize** (**Entity** Owner, JSONArray relationVector) throws **ExceptionVariable**

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

- Relations.java

cmas.ontology.workflow.Requirement Class Reference

Inheritance diagram for cmas.ontology.workflow.Requirement:



Public Member Functions

- **Requirement (WorkFlowTypes type, String Name, String Target)**
- **Requirement (Requirement entity, boolean instance)**
- **String getTarget ()**
- **void setTarget (String target)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **SpecificTypes getSpecificType ()**
- **WorkFlowTypes getWorkFlowType ()**
- **void setWorkFlowType (WorkFlowTypes workFlowType)**
- **JsonObject Serialize ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRealitions ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**
- **ArrayList< Entity > getSpecificRelationType (String Name)**
- **void addSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (Entity ent)**

- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassSimpleName** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static SpecificTypes **getSpecificType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- Properties **properties**
 - Relations **relations**
-

Member Function Documentation

Entity cmas.ontology.workflow.Requirement.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (p.382).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

void cmas.ontology.Entity.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

Entity cmas.ontology.workflow.WorkFlow.Factory (JsonObject job) throws ExceptionType [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (p.178).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.workflow.Requirement.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.workflow.Requirement.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

JsonObject cmas.ontology.workflow.WorkFlow.Serialize () [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

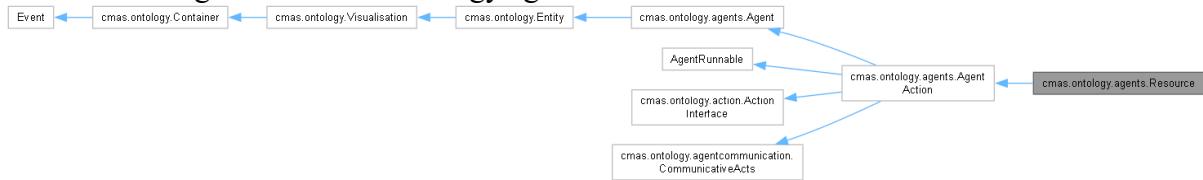
Referenced by **Entity()**.

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

- Requirement.java

cmas.ontology.agents.Resource Class Reference

Inheritance diagram for cmas.ontology.agents.Resource:



Public Member Functions

- **Resource** (Resource entity, boolean instance)
- **Entity copy** (boolean instance)
- void **Running** ()
- void **onMessage** (ACLMessages message)
- String **getEntityDescription** ()
- String **getEntityIconFile** ()
- synchronized String **getActions** ()
- synchronized void **setActions** (String actions)
- synchronized void **setState** (AgentState newState)
- synchronized AgentState **getState** ()
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- JsonObject **Serialize** ()
- synchronized void **setPaused** (boolean value)
- synchronized boolean **getPaused** ()
- synchronized void **postMessage** (ACLMessages message)
- **Entity getCommunicative** ()
- void **onInit** ()
- void **onDepot** ()
- void **onDeploy** ()
- void **onRun** ()
- void **onUndeploy** ()
- void **onFinished** ()
- void **onFailed** ()
- void **onStop** ()
- void **run** ()
- **MessageHandler getMessageHandler** ()
- void **gotoLine** (Integer line)
- void **stopEditor** ()
- void **compile** () throws ExceptionParameters, Exception
- String **getPhysicalID** ()
- void **setPhysicalID** (String physicalID)
- String **getAgentSpecificType** ()
- void **setAgentSpecificType** (String agentSpecificType)
- **AgentBaseTypes getAgentBaseType** ()
- void **addAbstractInterface** (AbstractInterface abstractInterface, String Name)
- void **clearAbstractInterfaces** ()
- **AbstractInterface getAbstractInterface** (String Name)
- RuntimeTree **getRuntimeTree** ()
- void **setRuntimeTree** (RuntimeTree runtimeTree)
- Domain **getRuntimeDomain** ()
- void **setRuntimeDomain** (Domain runtimeDomain)
- LinkedBlockingQueue<**WorkFlowLog**> **getHistory** ()
- void **setHistory** (LinkedBlockingQueue<**WorkFlowLog**> p)
- void **addToHistory** (**WorkFlowLog** log)
- Boolean **getAutoDeploy** ()
- void **setAutoDeploy** (Boolean autoDeploy)
- void **delete** ()

- `String toString ()`
- `String getID ()`
- `void setID (String ID)`
- `void setID ()`
- `String getName ()`
- `void setName (String Name)`
- `String getParentName ()`
- `void setParent (Entity ent)`
- `String getDescription ()`
- `void setDescription (String Description)`
- `String getInstanceID ()`
- `EntityType getType ()`
- `Properties getProperties ()`
- `Property<?> getProperty (String Name)`
- `Property<?> getPropertyIgnoreCase (String Name)`
- `void addProperty (Property<?> Item)`
- `boolean isCompatibleRelation (String Name, Entity ent)`
- `boolean hasRelation (String Name, Entity ent)`
- `LinkedHashMap< String, ArrayList< RelationObject > > getRealtions ()`
- `Collection< ArrayList< RelationObject > > getAllRelations ()`
- `ArrayList< RelationObject > getSpecificRelations (String Name)`
- `Entity findRelationByName (String RelationName, String EntityName)`
- `Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)`
- `ArrayList< Entity > getSpecificRelationType (String Name)`
- `void addSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (String Name, Entity ent)`
- `void removeSpecificRelation (Entity ent)`
- `void eventReflection (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)`
- `ImageView getEntityIcon ()`
- `void copyProperties (Container container, boolean instance)`
- `void setOverride (Boolean value)`
- `Boolean getProtected ()`
- `void setProtected (Boolean protectedValue)`
- `Object getUserData ()`
- `void setUserData (Object userData)`
- `String getClassname ()`
- `void setClassName (String newName)`
- `Relations getRealtionsObject ()`
- `ArrayList< BreakPoint > getBreakPointList ()`
- `void setBreakPointList (ArrayList< BreakPoint > breakPointList)`
- `Boolean isInstance ()`
- `default ACLMessage waitForPerformative (MessageHandler messageHandler, ACLMessage message, Performative answerType) throws InterruptedException`

Static Public Member Functions

- `static Entity Factory (JsonObject job) throws ExceptionType`
- `static Entity Factory (String job) throws ExceptionType, ExceptionConstant, ExceptionVariable`
- `static AgentBaseTypes getAgentType (JsonObject job) throws ExceptionType`

Protected Attributes

- `Action action`
- `MessageHandler messageHandler`
- `Properties properties`
- `Relations relations`

Member Function Documentation

Entity `cmas.ontology.agents.Resource.copy (boolean instance)`

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void `cmas.ontology.Entity.delete () [inherited]`

Reimplemented from **cmas.ontology.Container** (*p.161*).

void `cmas.ontology.agents.AgentAction.DeSerialize (JsonObject job) throws ExceptionType [inherited]`

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

Entity `cmas.ontology.agents.Agent.Factory (JsonObject job) throws ExceptionType [static], [inherited]`

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Entity `cmas.ontology.agents.AgentAction.getCommunicative () [inherited]`

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String `cmas.ontology.Entity.getDescription () [inherited]`

Returns

the entity description.

String `cmas.ontology.agents.Resource.getEntityDescription ()`

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String `cmas.ontology.agents.Resource.getEntityIconFile ()`

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String `cmas.ontology.Entity.getID () [inherited]`

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

MessageHandler cmas.ontology.agents.AgentAction.getMessageHandler () [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

void cmas.ontology.agents.AgentAction.gotoLine (Integer line) [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

void cmas.ontology.agents.Resource.onMessage (ACLMensaje message)

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

synchronized void cmas.ontology.agents.AgentAction.postMessage (ACLMensaje message) [inherited]

Implements **cmas.ontology.agentcommunication.CommunicativeActs** (*p.155*).

void cmas.ontology.agents.Resource.Running ()

Reimplemented from **cmas.ontology.agents.AgentAction** (*p.142*).

JsonObject cmas.ontology.agents.AgentAction.Serialize () [inherited]

Reimplemented from **cmas.ontology.agents.Agent** (*p.138*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

Description	a user defined description
-------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

ID	the new ID value.
----	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

Name	the new name.
------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.agents.AgentAction.stopEditor () [inherited]

Implements **cmas.ontology.action.ActionInterface** (*p.133*).

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

- Resource.java

cmas.ontology.adapter.REST.RESTClient.REST_TYPE Enum Reference

Public Attributes

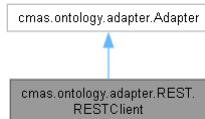
- NON
- ASSOCIATE

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

- RESTClient.java

cmas.ontology.adapter.REST.RESTClient Class Reference

Inheritance diagram for cmas.ontology.adapter.REST.RESTClient:



Classes

enum REST_TYPEPublic Member Functions

- **RESTClient** (String Source)
- void **SetSource** (String Source)
- void **setServerType** (REST_TYPE mode)
- REST_TYPE **getServerType** ()
- String **ReadData** (String Address, String Name, Object Value)
- void **WriteData** (String Address, String Name, Boolean bValue) throws Exception
- void **WriteData** (String Address, String Name, String sValue) throws Exception
- void **WriteData** (String Address, String Name, Integer iValue) throws Exception
- void **WriteData** (String Address, String Name, Double dValue) throws Exception
- boolean **close** ()
- void **setDebug** (boolean bValue)
- String **getDefaultValue** ()
- String **getDescription** ()
- boolean **isValid** (String Source)
- **Adapter** **createNew** (String Source)

Detailed Description

Main class for REST communication using POST and GET

Ideas

from:

<https://stackoverflow.com/questions/32586626/java-equivalent-of-f-networkcredential> Examples in ABB Robot Web Services:
https://developercenter.robotstudio.com/api/rwsApi/_hello_controller_json_8cs-example.html

```
For adapter type of REST the specific format is:  
REST:[HTTP]:[HOST]:[PORT]:[AUTH]:[USER]:[PASW]  
eg,  
REST:http:localhost:80:/rw/rapid/symbol/data/RAPID/T_ROB1/MainModule/:DIGEST:Default  
User:robotics  
    opc.tcp is static for OPC UA  
    HTTP http or https  
    HOST is the host name or TCP/IP address to the server  
    PORT is the port number to the server  
    AUTH is the name of the authentication type to use it can be DIGEST, BASIC or NONE  
    USER User name  
    PASW Password
```

Author

Anders Nilsson, University West

Member Function Documentation

boolean cmas.ontology.adapter.REST.RESTClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.REST.RESTClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.REST.RESTClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.REST.RESTClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.REST.RESTClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.REST.RESTClient.ReadData (String Address, String Name, Object Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.REST.RESTClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.REST.RESTClient.WriteData (String Address, String Name, Boolean bValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.REST.RESTClient.WriteData (String Address, String Name, Double dValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.REST.RESTClient.WriteData (String Address, String Name, Integer iValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.REST.RESTClient.WriteData (String Address, String Name, String sValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- RESTClient.java

cmas.ontology.adapter.socket.SocketTCPClient.SERVER_TYPE Enum Reference

Public Attributes

- **ASCII**
- **BINARY**

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

- `SocketTCPClient.java`

cmas.ontology.adapter.socket.SocketUDPClient.SERVER_TYPE Enum Reference

Public Attributes

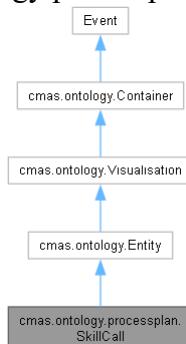
- **ASCII**
- **BINARY**

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

- `SocketUDPClient.java`

cmas.ontology.processplan.SkillCall Class Reference

Inheritance diagram for cmas.ontology.processplan.SkillCall:



Public Member Functions

- **SkillCall** (String Name)
- **SkillCall** (**SkillCall** skillCall, boolean instance)
- **Entity copy** (boolean instance)
- **String getEntityDescription** ()
- **String getEntityIconFile** ()
- **void delete** ()
- **String toString** ()
- **String getID** ()
- **void setID** (String ID)
- **void setID** ()
- **String getName** ()
- **void setName** (String Name)
- **String getParentName** ()
- **void setParent** (Entity ent)
- **String getDescription** ()
- **void setDescription** (String Description)
- **String getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- **void addProperty** (Property<?> Item)
- **boolean isCompatibleRelation** (String Name, Entity ent)
- **boolean hasRelation** (String Name, Entity ent)
- **LinkedHashMap< String, ArrayList< RelationObject > > getRealitions** ()
- **Collection< ArrayList< RelationObject > > getAllRelations** ()
- **ArrayList< RelationObject > getSpecificRelations** (String Name)
- **Entity findRelationByName** (String RelationName, String EntityName)
- **Entity findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- **ArrayList< Entity > getSpecificRelationType** (String Name)
- **void addSpecificRelation** (String Name, Entity ent)
- **void removeSpecificRelation** (String Name, Entity ent)
- **void removeSpecificRelation** (Entity ent)
- **void eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- **void DeSerialize** (JsonObject job) throws **ExceptionType**
- **JsonObject Serialize** ()
- **ImageView getEntityIcon** ()
- **void copyProperties** (Container container, boolean instance)
- **void setOverride** (Boolean value)
- **Boolean getProtected** ()

- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList<BreakPoint> **getBreakPointList** ()
- void **setBreakPointList** (ArrayList<BreakPoint> breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static Entity **Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

Entity **cmas.ontology.processplan.SkillCall.copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (p.382).

References **cmas.ontology.Entity.Entity()**.

void **cmas.ontology.Entity.delete** () [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

void **cmas.ontology.Entity.DeSerialize** (JsonObject job) throws **ExceptionType** [inherited]

Reimplemented from **cmas.ontology.Container** (p.161).

String **cmas.ontology.Entity.getDescription** () [inherited]

Returns

the entity description.

String **cmas.ontology.processplan.SkillCall.getEntityDescription** ()

Reimplemented from **cmas.ontology.Visualisation** (p.378).

String **cmas.ontology.processplan.SkillCall.getEntityIconFile** ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.Entity.Serialize () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

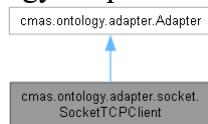
<i>Name</i>	the new name.
Reimplemented in cmas.ontology.Reference (p.289), and cmas.ontology.variables.VariableArray (p.339).	
Referenced by Entity() .	

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

- SkillCall.java

cmas.ontology.adapter.socket.SocketTCPClient Class Reference

Inheritance diagram for cmas.ontology.adapter.socket.SocketTCPClient:



Classes

enum SERVER_TYPEPublic Member Functions

- **SocketTCPClient** (String Source)
- void **setSource** (String Source)
- void **connectToServer** ()
- String **getServerHost** ()
- void **setServerHost** (String serverHost)
- int **getServerPort** ()
- void **setServerPort** (int serverPort)
- SERVER_TYPE **getServerType** ()
- void **setServerType** (SERVER_TYPE serverType)
- int **getTimeout** ()
- void **setTimeout** (int timeout)
- String **ReadData** (String Address, String Name, Object Value) throws Exception
- void **WriteData** (String Address, String Name, Boolean bValue) throws Exception
- void **WriteData** (String Address, String Name, String sValue) throws Exception
- void **WriteData** (String Address, String Name, Integer iValue) throws Exception
- void **WriteData** (String Address, String Name, Double dValue) throws Exception
- boolean **close** ()
- void **setDebug** (boolean bValue)
- String **getDefaultValue** ()
- String **getDescription** ()
- boolean **isValid** (String Source)
- Adapter **createNew** (String Source)

Member Function Documentation

boolean cmas.ontology.adapter.socket.SocketTCPClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.socket.SocketTCPClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.socket.SocketTCPClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.socket.SocketTCPClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.socket.SocketTCPClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.socket.SocketTCPClient.ReadData (String Address, String Name, Object Value) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketTCPClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketTCPClient.WriteData (String Address, String Name, Boolean bValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketTCPClient.WriteData (String Address, String Name, Double dValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketTCPClient.WriteData (String Address, String Name, Integer iValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketTCPClient.WriteData (String Address, String Name, String sValue) throws Exception

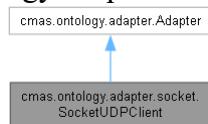
Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- SocketTCPClient.java

cmas.ontology.adapter.socket.SocketUDPClient Class Reference

Inheritance diagram for cmas.ontology.adapter.socket.SocketUDPClient:



Classes

enum SERVER_TYPEPublic Member Functions

- **SocketUDPClient** (String Source)
- void **setSource** (String Source)
- void **connectToServer** ()
- String **getServerHost** ()
- void **setServerHost** (String serverHost)
- int **getServerPort** ()
- void **setServerPort** (int serverPort)
- SERVER_TYPE **getServerType** ()
- void **setServerType** (SERVER_TYPE serverType)
- int **getTimeout** ()
- void **setTimeout** (int timeout)
- String **ReadData** (String Address, String Name, Object Value) throws Exception
- void **WriteData** (String Address, String Name, Boolean bValue) throws Exception
- void **WriteData** (String Address, String Name, String sValue) throws Exception
- void **WriteData** (String Address, String Name, Integer iValue) throws Exception
- void **WriteData** (String Address, String Name, Double dValue) throws Exception
- boolean **close** ()
- void **setDebug** (boolean bValue)
- String **getDefaultValue** ()
- String **getDescription** ()
- boolean **isValid** (String Source)
- **Adapter** **createNew** (String Source)

Member Function Documentation

boolean cmas.ontology.adapter.socket.SocketUDPClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.socket.SocketUDPClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.socket.SocketUDPClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.socket.SocketUDPClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.socket.SocketUDPClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.socket.SocketUDPClient.ReadData (String Address, String Name, Object Value) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketUDPClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketUDPClient.WriteData (String Address, String Name, Boolean bValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketUDPClient.WriteData (String Address, String Name, Double dValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketUDPClient.WriteData (String Address, String Name, Integer iValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.socket.SocketUDPClient.WriteData (String Address, String Name, String sValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- SocketUDPClient.java

cmas.ontology.Property< T >.Specific.PropertyType Enum Reference

Public Attributes

- INHERITANCE
- REFERENCEFILE
- PARENTEDITOR
- IDEEDIT
- TEXTEDITOR
- STRINGEDIT
- VARIABLEEDITOR
- SFCEDITOR
- PROCESSPLANEDITOR
- BATCHEDITOR
- INTERFACEBROWSER
- ADAPTEREDIT
- ADDRESSBROWSE
- HISTORYSHOW
- HIDE
- RUNTIME
- MODELLING
- HIDEBASIC

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

- Property.java

cmas.ontology.workflow.WorkFlow.SpecificTypes Enum Reference

Public Attributes

- REQUIREMENTS
- WORKFLOWLOG

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

- WorkFlow.java

cmas.ontology.variables.Variable.TypeOfVariable Enum Reference

Public Attributes

- **PRIMITIVE**
- **ARRAY**
- **OBJECT**

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

- Variable.java

cmas.ontology.Types Class Reference

Static Public Member Functions

- static Entity **getType** (String Name)
- static Entity **Factory** (String Name) throws InstantiationException, IllegalAccessException, IllegalArgumentException, InvocationTargetException, NoSuchMethodException, SecurityException
- static void **addEntity** (String Category, Entity newObject)

Static Public Attributes

- static final ArrayList<Variable> **ValueTypes**
- static final ArrayList<ProcessPlan> **SkillTypes**
- static final ArrayList<Interface> **InterfaceTypes**
- static final ArrayList<Goal> **GoalTypes**
- static final ArrayList<Executable> **ExecutableTypes**
- static final ArrayList<Hazard> **HazardTypes**
- static final ArrayList<ProcessPlan> **ProcessPlanTypes**
- static final ArrayList<Agent> **AgentTypes**
- static final ArrayList<Batch> **BatchTypes**
- static final ArrayList<DeploymentAgent> **DeploymentTypes**
- static final ArrayList<WorkFlowLog> **WorkFlowTypes**
- static final ArrayList<Requirement> **RequirementTypes**
- static final ArrayList<Entity> **FolderTypes**
- static final ArrayList<Entity> **EntityTypes**

Member Data Documentation

final ArrayList<Agent> cmas.ontology.Types.AgentTypes [static]

Initial value:

```
=new ArrayList<Agent>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Part());
        add(new Material());
        add(new Resource());
    }
}
```

final ArrayList<Batch> cmas.ontology.Types.BatchTypes [static]

Initial value:

```
=new ArrayList<Batch>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Batch());
    }
}
```

final ArrayList<DeploymentAgent> cmas.ontology.Types.DeploymentTypes [static]

Initial value:

```
=new ArrayList<DeploymentAgent>() {
    private static final long serialVersionUID = 1L;
    {
        add(new DeploymentAgent());
    }
}
```

final ArrayList<Entity> cmas.ontology.Types.EntityTypes [static]

Initial value:

```
=new ArrayList<Entity>() {
    private static final long serialVersionUID = 1L;
```

```

    {
        try {
            addAll(AgentTypes);
            addAll(InterfaceTypes);
            addAll(GoalTypes);
            addAll(ValueTypes);

            addAll(WorkFlowTypes);
            addAll(RequirementTypes);
            addAll(ExecutableTypes);
            addAll(ProcessPlanTypes);
            addAll(HazardTypes);

            addAll(DeploymentTypes);
            addAll(BatchTypes);
            add(new Folder());
        } catch (Exception e) {
            if (Main.mainPreferences.getValueAsBoolean("Debug"))
                System.out.println("unable to initialise static Types (3):"
+ e.getMessage());
        }
    }
}

```

final ArrayList<Executable> cmas.ontology.Types.ExecutableTypes [static]

Initial value:

```

=new ArrayList<Executable>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Executable());
    }
}

```

final ArrayList<Entity> cmas.ontology.Types.FolderTypes [static]

Initial value:

```

=new ArrayList<Entity>() {
    private static final long serialVersionUID = 1L;
    {
        try {
            addAll(AgentTypes);
            addAll(InterfaceTypes);
            addAll(GoalTypes);
            addAll(ValueTypes);

            addAll(WorkFlowTypes);
            addAll(RequirementTypes);
            addAll(ExecutableTypes);
            addAll(ProcessPlanTypes);

            addAll(BatchTypes);
            addAll(DeploymentTypes);
        } catch (Exception e) {
            if (Main.mainPreferences.getValueAsBoolean("Debug"))
                System.out.println("unable to initialise static Types
(2) :" + e.getMessage());
        };
    }
}

```

final ArrayList<Goal> cmas.ontology.Types.GoalTypes [static]

Initial value:

```

=new ArrayList<Goal>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Goal());
    }
}

```

final ArrayList<Hazard> cmas.ontology.Types.HazardTypes [static]

Initial value:

```

=new ArrayList<Hazard>() {
    private static final long serialVersionUID = 1L;
    {
}
}

```

```
        add(new Hazard());
    } }
```

final ArrayList<Interface> cmas.ontology.Types.InterfaceTypes [static]

Initial value:

```
=new ArrayList<Interface>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Interface());
        add(new InterfaceSpecialisation());
        add(new InterfaceTransport());
        add(new InterfaceLocation());
        add(new InterfaceAssembly());
    } }
```

final ArrayList<ProcessPlan> cmas.ontology.Types.ProcessPlanTypes [static]

Initial value:

```
=new ArrayList<ProcessPlan>() {
    private static final long serialVersionUID = 1L;
    {
        add(new ProcessPlanStructuredText());
        add(new ProcessPlanGoals());

        if(ProcessPlanJava.ENABLED)
            add(new ProcessPlanJava());
    } }
```

final ArrayList<Requirement> cmas.ontology.Types.RequirementTypes [static]

Initial value:

```
=new ArrayList<Requirement>() {
    private static final long serialVersionUID = 1L;
    {
        add(new Requirement());
    } }
```

final ArrayList<ProcessPlan> cmas.ontology.Types.SkillTypes [static]

Initial value:

```
=new ArrayList<ProcessPlan>() {
    private static final long serialVersionUID = 1L;
    {
        ProcessPlan p=new ProcessPlanStructuredText();
        p.setName("Skill Structured Text");
        add(p);

        p=new ProcessPlanGoals();
        p.setName("Skill SFC Goals");
        add(p);

        if(ProcessPlanJava.ENABLED) {
            p=new ProcessPlanJava();
            p.setName("Skill Java");
            add(p);
        }
    } }
```

final ArrayList<Variable> cmas.ontology.Types.ValueTypes [static]

Initial value:

```
=new ArrayList<Variable>() {
    private static final long serialVersionUID = 1L;
    {
        try {
            add(new VariableInteger());
            add(new VariableReal());
            add(new VariableBoolean());
            add(new VariableString());

            add(new VariableArray("VariableArray", new VariableInteger(),new Integer[] {1}, new Integer[] {2}));
        }
```

```
        add(new VariableObject("VariableObject"));
    }catch(Exception e) {
        if(Main.mainPreferences.getValueAsBoolean("Debug"))
            System.out.println("unable to initialise static Types (1): " +
e.getMessage());
    }
}
```

final ArrayList<WorkFlowLog> cmas.ontology.Types.WorkFlowTypes [static]

Initial value:

```
=new ArrayList<WorkFlowLog>() {
    private static final long serialVersionUID = 1L;
}
```

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

- Types.java

cmas.ontology.Entity.UID Class Reference

Static Public Member Functions

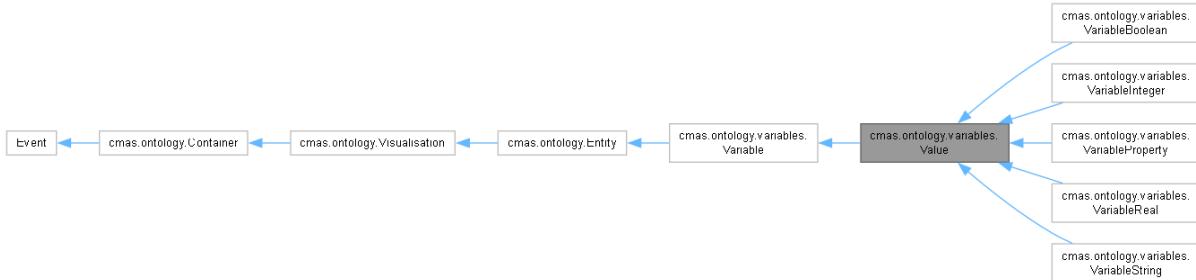
- static synchronized String **Generate ()**

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

- Entity.java

cmas.ontology.variables.Value Class Reference

Inheritance diagram for cmas.ontology.variables.Value:



Classes

enum ValueTypes Public Member Functions

- **Value (ValueTypes type)** throws **ExceptionType**
- **Value (Value variable, boolean instance)** throws **ExceptionType**
- **Object getCopyOfValue ()** throws **ExceptionType**
- **String getStringValue ()** throws **ExceptionType**
- **void setReferenceProperty (Property<?> prop)**
- **Property<?> getReferenceProperty ()**
- **boolean isReferenceProperty ()**
- **ValueTypes getValueType ()**
- **Boolean isBoolean ()**
- **Boolean isInteger ()**
- **Boolean isReal ()**
- **Boolean isString ()**
- **Object getValue ()**
- **void setValue (Object newValue)** throws **ExceptionNullValue, ExceptionType, ExceptionConstant**
- **Object valueOf (String s)**
- **String getUnit ()**
- **void setUnit (String value)**
- **String getUpperBound ()**
- **void setUpperBound (String upperBound)**
- **String getLowerBound ()**
- **void setLowerBound (String lowerBound)**
- **boolean isWithinRange (Object testValue)** throws **ExceptionType**
- **Integer parseInteger (String Value, Integer Default)**
- **boolean parseBoolean (String Value, Boolean Default)**
- **Double parseReal (String Value, Double Default)**
- **JsonObject Serialize ()**
- **abstract Entity copy (boolean instance)**
- **abstract Iterator getIterator ()**
- **Variable getIteratorObject ()**
- **boolean hasNext (Iterator iterator)**
- **Object getNext (Iterator iterator)** throws **ExceptionVariable, ExceptionType, ExceptionConstant**
- **TypeOfVariable getVariableType ()**
- **Boolean isArray ()**
- **Boolean isPrimitive ()**
- **Boolean isObject ()**
- **Boolean getReadOnly ()**
- **Boolean hasAdapter ()**
- **void setReadOnly (Boolean ReadOnly)**
- **Boolean getStoreLongTerm ()**

- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable** **getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType** **getType** ()
- **Properties** **getProperties** ()
- **Property<?>** **getProperty** (String Name)
- **Property<?>** **getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< **RelationObject** >> **getRealtions** ()
- Collection< ArrayList< **RelationObject** >> **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassname** ()
- void **setClassName** (String newName)
- **Relations** **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **ValueTypes** **getValueType** (**JsonObject** job) throws **ExceptionType**
- static **Entity Factory** (**JsonObject** job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static **TypeOfVariable** **getVariableType** (**JsonObject** job) throws **ExceptionType**
- static **TypeOfVariable** **getTypeOfVariable** (**JsonObject** job) throws **ExceptionType**
- static **Entity Factory** (**String** job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Member Functions

- void **setIgnoreReadOnly** (**boolean** value)

Protected Attributes

- **Property<?>** **ReferenceProperty**
- **ValueTypes** **ValueType**
- **String** **UpperBound**
- **String** **LowerBound**
- **String** **Unit**
- **ValueRange** **upperRangeExpression**
- **ValueRange** **lowerRangeExpression**
- **Boolean** **ignoreReadOnly** =true
- **TypeOfVariable** **VariableType**
- **Boolean** **ReadOnly**
- **Boolean** **StoreLongTerm**
- **Boolean** **Error**
- **String** **Address**
- **String** **Source**
- **Adapter** **VariableAdapter**
- **Variable Reference**
- **Properties** **properties**
- **Relations** **relations**

Static Protected Attributes

- static final **Boolean** **ignoreTypeCheck** =false
- static final **Boolean** **ignoreRangeCheck** =false

Member Function Documentation

abstract Entity **cmas.ontology.variables.Variable.copy** (**boolean** **instance**) [**abstract**], [**inherited**]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

instance	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (p.382).

Reimplemented in **cmas.ontology.variables.VariableArray** (p.337), **cmas.ontology.variables.VariableBoolean** (p.343), **cmas.ontology.variables.VariableInteger** (p.349), **cmas.ontology.variables.VariableObject** (p.356), **cmas.ontology.variables.VariableProperty** (p.362), **cmas.ontology.variables.VariableReal** (p.368), and **cmas.ontology.variables.VariableString** (p.374).

References **cmas.ontology.Entity.Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Value.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.Value.getCopyOfValue () throws ExceptionType

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

String cmas.ontology.variables.Value.getStringValue () throws ExceptionType

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

Object cmas.ontology.variables.Value.getValue ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.Value.Serialize ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.variables.Value.setValue (Object newValue) throws ExceptionNullValue, ExceptionType, ExceptionConstant

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.Value.valueOf (String s)

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- Value.java

cmas.ontology.variables.ValueRange Class Reference

Classes

class RangePublic Member Functions

- boolean **Evaluate** (Number value) throws **ExceptionType**
- void **compile** (String expression) throws **ExceptionType**

Static Public Member Functions

- static boolean **isARangeExpression** (String expression)

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

- ValueRange.java

cmas.ontology.variables.Value.ValueTypes Enum Reference

Public Attributes

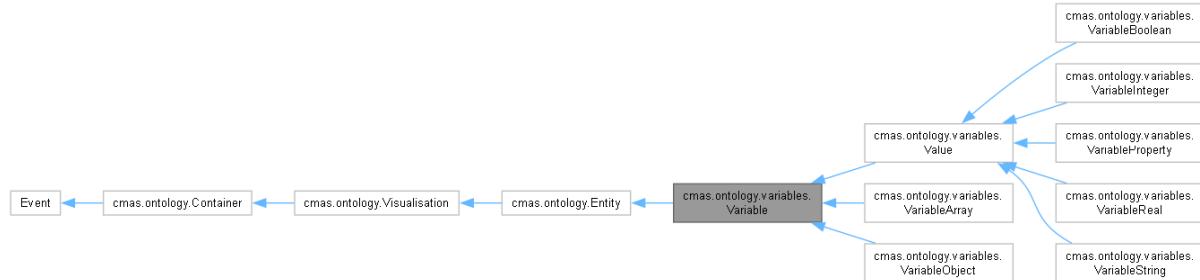
- **NON**
- **INTEGER**
- **REAL**
- **BOOLEAN**
- **STRING**

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

- Value.java

cmas.ontology.variables.Variable Class Reference

Inheritance diagram for cmas.ontology.variables.Variable:



Classes

enum TypeOfVariablePublic Member Functions

- **Variable (TypeOfVariable type)**
- **Variable (Variable variable, boolean instance)**
- abstract **Entity copy** (boolean instance)
- abstract Object **getCopyOfValue ()** throws **ExceptionType**
- abstract String **getStringValue ()** throws **ExceptionType, ExceptionVariable**
- abstract Object **getValue ()**
- abstract void **setValue (Object newValue)** throws **ExceptionType, ExceptionNullValue, ExceptionConstant, ExceptionVariable**
- abstract Object **valueOf (String value)** throws **ExceptionVariable, ExceptionType, ExceptionNullValue, ExceptionConstant**
- abstract **Iterator getIterator ()**
- **Variable getIteratorObject ()**
- boolean **hasNext (Iterator iterator)**
- Object **getNext (Iterator iterator)** throws **ExceptionVariable, ExceptionType, ExceptionConstant**
- **TypeOfVariable getVariableType ()**
- Boolean **isArray ()**
- Boolean **isPrimitive ()**
- Boolean **isObject ()**
- Boolean **getReadOnly ()**
- Boolean **hasAdapter ()**
- void **setReadOnly (Boolean ReadOnly)**
- Boolean **getStoreLongTerm ()**
- void **setStoreLongTerm (Boolean storeLongTerm)**
- Boolean **getError ()**
- void **setError (Boolean error)**
- String **getAddress ()**
- void **setAddress (String address)**
- String **getSource ()**
- void **setSource (String source)**
- void **setReference (Variable ref)**
- **Variable getReference ()**
- boolean **isReference ()**
- JSONObject **Serialize ()**
- boolean **compare (Object o)**
- boolean **isVariableObjectMatching (VariableObject variable1, VariableObject variable2)**
- void **delete ()**
- String **toString ()**
- String **getID ()**
- void **setID (String ID)**
- void **setID ()**

- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static TypeOfVariable **getVariableType** (JsonObject job) throws **ExceptionType**
- static TypeOfVariable **getTypeOfVariable** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (JsonObject job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static Entity **Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**

Protected Attributes

- TypeOfVariable **VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- Adapter **VariableAdapter**
- Variable **Reference**

- **Properties properties**
 - **Relations relations**
-

Member Function Documentation

abstract Entity cmas.ontology.variables.Variable.copy (boolean instance) [abstract]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

Reimplemented in **cmas.ontology.variables.VariableArray** (*p.337*), **cmas.ontology.variables.VariableBoolean** (*p.343*), **cmas.ontology.variables.VariableInteger** (*p.349*), **cmas.ontology.variables.VariableObject** (*p.356*), **cmas.ontology.variables.VariableProperty** (*p.362*), **cmas.ontology.variables.VariableReal** (*p.368*), and **cmas.ontology.variables.VariableString** (*p.374*).

References **cmas.ontology.Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

**void cmas.ontology.Entity.Deserialize (JsonObject job) throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.Container** (*p.161*).

**Entity cmas.ontology.variables.Variable.Factory (JsonObject job) throws
ExceptionType, ExceptionConstant, ExceptionVariable [static]**

Reimplemented from **cmas.ontology.Entity** (*p.178*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

JsonObject cmas.ontology.variables.Variable.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

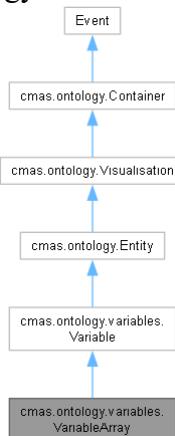
Referenced by **Entity()**.

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

- Variable.java

cmas.ontology.variables.VariableArray Class Reference

Inheritance diagram for cmas.ontology.variables.VariableArray:



Public Member Functions

- **VariableArray** (String Name, Object initialValue, Integer[] LowerRange, Integer[] UpperRange) throws **ExceptionVariable**, **ExceptionType**
- **VariableArray** (**VariableArray** array, boolean instance) throws **ExceptionType**
- void **resize** (Object initialValue, Integer[] LowerRange, Integer[] UpperRange) throws **ExceptionVariable**, **ExceptionType**
- void **setName** (String name)
- Object **getValue** ()
- Object **getValue** (Integer Index) throws **ExceptionVariable**
- void **setValue** (Object newValue) throws **ExceptionType**, **ExceptionVariable**, **ExceptionNullValue**
- void **setValue** (Integer Index, Object newValue) throws **ExceptionVariable**, **ExceptionType**, **ExceptionNullValue**, **ExceptionConstant**
- String **getStringValue** () throws **ExceptionType**, **ExceptionVariable**
- Object **valueOf** (String value) throws **ExceptionType**, **ExceptionVariable**, **ExceptionNullValue**, **ExceptionConstant**
- Object **getCopyOfValue** () throws **ExceptionType**
- Integer **getSize** ()
- Integer **getLowerRange** ()
- Integer **getUpperRange** ()
- void **setUpperRange** (Integer range)
- void **setLowerRange** (Integer range)
- **ValueTypes** **getValueType** ()
- void **setValueType** (**ValueTypes** type)
- JsonObject **Serialize** ()
- **Entity** **copy** (boolean instance)
- String **getEntityDescription** ()
- String **getEntityIconFile** ()
- **Iterator** **getIterator** ()
- boolean **hasNext** (**Iterator** iterator)
- Object **getNext** (**Iterator** iterator) throws **ExceptionVariable**
- **Variable** **getIteratorObject** ()
- boolean **isRelationReference** ()
- void **setRelationReference** (ArrayList<**RelationObject**> relation) throws **ExceptionVariable**, **ExceptionType**
- ArrayList<**RelationObject**> **getRelationReference** ()
- **TypeOfVariable** **getVariableType** ()
- Boolean **isArray** ()
- Boolean **isPrimitive** ()

- Boolean **isObject** ()
- Boolean **getReadOnly** ()
- Boolean **hasAdapter** ()
- void **setReadOnly** (Boolean ReadOnly)
- Boolean **getStoreLongTerm** ()
- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (Variable ref)
- Variable **getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (VariableObject variable1, VariableObject variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- EntityType **getType** ()
- Properties **getProperties** ()
- Property<?> **getProperty** (String Name)
- Property<?> **getPropertyIgnoreCase** (String Name)
- void **addProperty** (Property<?> Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< RelationObject > > **getRealitions** ()
- Collection< ArrayList< RelationObject > > **getAllRelations** ()
- ArrayList< RelationObject > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- Relations **getRealitionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **Entity Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static **TypeOfVariable getVariableType** (JsonObject job) throws **ExceptionType**
- static **TypeOfVariable getTypeOfVariable** (JsonObject job) throws **ExceptionType**
- static **Entity Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- **TypeOfVariable VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- Adapter **VariableAdapter**
- **Variable Reference**
- **Properties properties**
- **Relations relations**

Member Function Documentation

Entity cmas.ontology.variables.VariableArray.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.variables.Variable** (*p.333*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.VariableArray.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.VariableArray.getCopyOfValue () throws ExceptionType

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableArray.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableArray.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableArray.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Variable cmas.ontology.variables.VariableArray.getIteratorObject ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

Object cmas.ontology.variables.VariableArray.getNext (Iterator iterator) throws ExceptionVariable

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.variables.VariableArray.getStringValue () throws ExceptionType, ExceptionVariable

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

Object cmas.ontology.variables.VariableArray.getValue ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

boolean cmas.ontology.variables.VariableArray.hasNext (Iterator iterator)

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.VariableArray.Serialize ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setId () [inherited]

setId will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setId (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.variables.VariableArray.setName (String Name)

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented from **cmas.ontology.Entity** (*p.388*).

References **cmas.ontology.Entity.getName()**, and **setName()**.

Referenced by **setName()**.

**void cmas.ontology.variables.VariableArray.setValue (Object newValue) throws
ExceptionType, ExceptionVariable, ExceptionNullValue**

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

**Object cmas.ontology.variables.VariableArray.valueOf (String value) throws
ExceptionType, ExceptionVariable, ExceptionNullValue, ExceptionConstant**

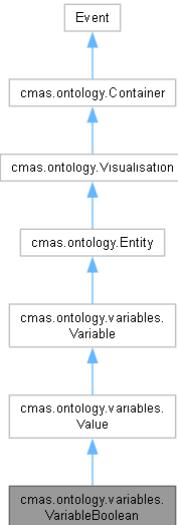
Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- VariableArray.java

cmas.ontology.variables.VariableBoolean Class Reference

Inheritance diagram for cmas.ontology.variables.VariableBoolean:



Public Member Functions

- **VariableBoolean (String Name)** throws **ExceptionType, ExceptionConstant**
- **VariableBoolean (String Name, Boolean DefaultValue)** throws **ExceptionType, ExceptionConstant**
- **VariableBoolean (VariableBoolean variable, boolean instance)** throws **ExceptionType**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **Iterator getIterator ()**
- **Object getCopyOfValue ()** throws **ExceptionType**
- **String getStringValue ()** throws **ExceptionType**
- **ValueTypes getValueType ()**
- **void setReferenceProperty (Property<?> prop)**
- **Property<?> getReferenceProperty ()**
- **boolean isReferenceProperty ()**
- **Boolean isBoolean ()**
- **Boolean isInteger ()**
- **Boolean isReal ()**
- **Boolean isString ()**
- **Object getValue ()**
- **void setValue (Object newValue)** throws **ExceptionNullValue, ExceptionType, ExceptionConstant**
- **Object valueOf (String s)**
- **String getUnit ()**
- **void setUnit (String value)**
- **String getUpperBound ()**
- **void setUpperBound (String upperBound)**
- **String getLowerBound ()**
- **void setLowerBound (String lowerBound)**
- **boolean isWithinRange (Object testValue)** throws **ExceptionType**
- **Integer parseInteger (String Value, Integer Default)**
- **boolean parseBoolean (String Value, Boolean Default)**
- **Double parseReal (String Value, Double Default)**
- **JsonObject Serialize ()**
- **Variable getIteratorObject ()**
- **boolean hasNext (Iterator iterator)**

- Object **getNext** (Iterator iterator) throws **ExceptionVariable**, **ExceptionType**, **ExceptionConstant**
- **TypeOfVariable getVariableType** ()
- Boolean **isArray** ()
- Boolean **isPrimitive** ()
- Boolean **isObject** ()
- Boolean **getReadOnly** ()
- Boolean **hasAdapter** ()
- void **setReadOnly** (Boolean ReadOnly)
- Boolean **getStoreLongTerm** ()
- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, **Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- LinkedHashMap< String, ArrayList< **RelationObject** >> **getRealitions** ()
- Collection< ArrayList< **RelationObject** >> **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- **Entity findRelationByName** (String RelationName, String EntityName)
- **Entity findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (**Entity** ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)

- String **getClassName** ()
- void **setClassName** (String newName)
- **Relations getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **ValueTypes getValueType** (JsonObject job) throws **ExceptionType**
- static **Entity Factory** (JsonObject job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **Entity Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **TypeOfVariable getVariableType** (JsonObject job) throws **ExceptionType**
- static **TypeOfVariable getTypeOfVariable** (JsonObject job) throws **ExceptionType**

Protected Member Functions

- void **setIgnoreReadOnly** (boolean value)

Protected Attributes

- **Property<?> ReferenceProperty**
- **ValueTypes ValueType**
- String **UpperBound**
- String **LowerBound**
- String **Unit**
- **ValueRange upperRangeExpression**
- **ValueRange lowerRangeExpression**
- Boolean **ignoreReadOnly** =true
- **TypeOfVariable VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- **Adapter VariableAdapter**
- **Variable Reference**
- **Properties properties**
- **Relations relations**

Static Protected Attributes

- static final Boolean **ignoreTypeCheck** =false
- static final Boolean **ignoreRangeCheck** =false

Member Function Documentation

Entity cmas.ontology.variables.VariableBoolean.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.variables.Variable** (*p.333*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Value.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static], [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.Value.getCopyOfValue () throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableBoolean.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableBoolean.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableBoolean.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

**String cmas.ontology.variables.Value.getStringValue () throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType ()[inherited]**Returns**

entity type.

Object cmas.ontology.variables.Value.getValue ()[inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.Value.Serialize ()[inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID ()[inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID)[inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

```
void cmas.ontology.variables.Value.setValue (Object newValue) throws  
ExceptionNullValue, ExceptionType, ExceptionConstant [inherited]
```

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

```
Object cmas.ontology.variables.Value.valueOf (String s) [inherited]
```

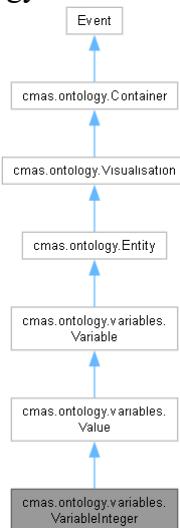
Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- VariableBoolean.java

cmas.ontology.variables.VariableInteger Class Reference

Inheritance diagram for cmas.ontology.variables.VariableInteger:



Public Member Functions

- **VariableInteger (String Name)** throws **ExceptionType, ExceptionConstant**
- **VariableInteger (String Name, Integer DefaultValue)** throws **ExceptionType, ExceptionConstant**
- **VariableInteger (VariableInteger variable, boolean instance)** throws **ExceptionType**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **Iterator getIterator ()**
- **Object getCopyOfValue ()** throws **ExceptionType**
- **String getStringValue ()** throws **ExceptionType**
- **ValueTypes getValueType ()**
- **void setReferenceProperty (Property<?> prop)**
- **Property<?> getReferenceProperty ()**
- **boolean isReferenceProperty ()**
- **Boolean isBoolean ()**
- **Boolean isInteger ()**
- **Boolean isReal ()**
- **Boolean isString ()**
- **Object getValue ()**
- **void setValue (Object newValue)** throws **ExceptionNullValue, ExceptionType, ExceptionConstant**
- **Object valueOf (String s)**
- **String getUnit ()**
- **void setUnit (String value)**
- **String getUpperBound ()**
- **void setUpperBound (String upperBound)**
- **String getLowerBound ()**
- **void setLowerBound (String lowerBound)**
- **boolean isWithinRange (Object testValue)** throws **ExceptionType**
- **Integer parseInteger (String Value, Integer Default)**
- **boolean parseBoolean (String Value, Boolean Default)**
- **Double parseReal (String Value, Double Default)**
- **JsonObject Serialize ()**
- **Variable getIteratorObject ()**
- **boolean hasNext (Iterator iterator)**

- Object **getNext** (Iterator iterator) throws **ExceptionVariable**, **ExceptionType**, **ExceptionConstant**
- **TypeOfVariable getVariableType** ()
- Boolean **isArray** ()
- Boolean **isPrimitive** ()
- Boolean **isObject** ()
- Boolean **getReadOnly** ()
- Boolean **hasAdapter** ()
- void **setReadOnly** (Boolean ReadOnly)
- Boolean **getStoreLongTerm** ()
- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< **RelationObject** >> **getRealitions** ()
- Collection< ArrayList< **RelationObject** >> **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)

- String **getClassName** ()
- void **setClassName** (String newName)
- **Relations getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **ValueTypes getValueType** (JsonObject job) throws **ExceptionType**
- static **Entity Factory** (JsonObject job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **Entity Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **TypeOfVariable getVariableType** (JsonObject job) throws **ExceptionType**
- static **TypeOfVariable getTypeOfVariable** (JsonObject job) throws **ExceptionType**

Protected Member Functions

- void **setIgnoreReadOnly** (boolean value)

Protected Attributes

- **Property<?> ReferenceProperty**
- **ValueTypes ValueType**
- String **UpperBound**
- String **LowerBound**
- String **Unit**
- **ValueRange upperRangeExpression**
- **ValueRange lowerRangeExpression**
- Boolean **ignoreReadOnly** =true
- **TypeOfVariable VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- **Adapter VariableAdapter**
- **Variable Reference**
- **Properties properties**
- **Relations relations**

Static Protected Attributes

- static final Boolean **ignoreTypeCheck** =false
- static final Boolean **ignoreRangeCheck** =false

Member Function Documentation

Entity cmas.ontology.variables.VariableInteger.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.variables.Variable** (*p.333*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Value.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static], [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.Value.getCopyOfValue () throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableInteger.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableInteger.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableInteger.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

**String cmas.ontology.variables.Value.getStringValue () throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

Object cmas.ontology.variables.Value.getValue () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.Value.Serialize () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

```
void cmas.ontology.variables.Value.setValue (Object newValue) throws  
ExceptionNullValue, ExceptionType, ExceptionConstant [inherited]
```

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

```
Object cmas.ontology.variables.Value.valueOf (String s) [inherited]
```

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- VariableInteger.java

cmas.ontology.interfaces.InterfaceBase.VariableMatchingType Enum Reference

Public Attributes

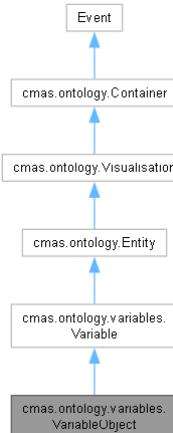
- REQUIRED
- OPTIONAL
- IGNORE

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

- InterfaceBase.java

cmas.ontology.variables.VariableObject Class Reference

Inheritance diagram for cmas.ontology.variables.VariableObject:



Public Member Functions

- **VariableObject (String Name)** throws **ExceptionVariable, ExceptionType**
- **VariableObject (String Name, VariableObject DefaultValue)** throws **ExceptionType, ExceptionVariable, ExceptionNullValue, ExceptionConstant**
- **VariableObject (VariableObject variableobject, boolean instance)** throws **ExceptionType**
- Object **getValue ()**
- Integer **getSize ()**
- void **clear ()**
- void **setValue (Object newValue)** throws **ExceptionType, ExceptionVariable, ExceptionNullValue, ExceptionConstant**
- void **replaceValue (Object newValue)** throws **ExceptionType, ExceptionVariable, ExceptionNullValue, ExceptionConstant**
- String **getStringValue ()** throws **ExceptionType, ExceptionVariable**
- String **getJSONObjectValue ()** throws **ExceptionType, ExceptionVariable**
- Object **valueOf (String value)** throws **ExceptionVariable, ExceptionType, ExceptionNullValue, ExceptionConstant**
- Object **getCopyOfValue ()** throws **ExceptionType**
- Entity **copy (boolean instance)**
- String **getEntityDescription ()**
- String **getEntityIconFile ()**
- Iterator **getIterator ()**
- boolean **hasNext (Iterator iterator)**
- Object **getNext (Iterator iterator)** throws **ExceptionVariable**
- Variable **getIteratorObject ()**
- Entity **getOntologyObjectReference ()**
- void **setOntologyObjectReference (Entity ontologyObjectReference)** throws **ExceptionParameters**
- TypeOfVariable **getVariableType ()**
- Boolean **isArray ()**
- Boolean **isPrimitive ()**
- Boolean **isObject ()**
- Boolean **getReadOnly ()**
- Boolean **hasAdapter ()**
- void **setReadOnly (Boolean ReadOnly)**
- Boolean **getStoreLongTerm ()**
- void **setStoreLongTerm (Boolean storeLongTerm)**
- Boolean **getError ()**
- void **setError (Boolean error)**
- String **getAddress ()**

- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable** **getReference** ()
- boolean **isReference** ()
- JsonObject **Serialize** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType** **getType** ()
- **Properties** **getProperties** ()
- **Property<?>** **getProperty** (String Name)
- **Property<?>** **getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, Entity ent)
- boolean **hasRelation** (String Name, Entity ent)
- LinkedHashMap< String, ArrayList< **RelationObject** > > **getRealtions** ()
- Collection< ArrayList< **RelationObject** > > **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- Entity **findRelationByName** (String RelationName, String EntityName)
- Entity **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- **Relations** **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **Variable** **fromJson** (String Name, JsonElement json) throws **ExceptionVariable**, **ExceptionType**, **ExceptionConstant**, **ExceptionNullValue**
- static **TypeOfVariable** **getVariableType** (JsonObject job) throws **ExceptionType**
- static **TypeOfVariable** **getTypeOfVariable** (JsonObject job) throws **ExceptionType**

- static **Entity Factory** (JsonObject job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static **Entity Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- **TypeOfVariable** **VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- **Adapter VariableAdapter**
- **Variable Reference**
- **Properties properties**
- **Relations relations**

Member Function Documentation

Entity cmas.ontology.variables.VariableObject.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.variables.Variable** (*p.333*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Variable.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

Object cmas.ontology.variables.VariableObject.getCopyOfValue () throws ExceptionType

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableObject.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableObject.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableObject.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Variable cmas.ontology.variables.VariableObject.getIteratorObject ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

Object cmas.ontology.variables.VariableObject.getNext (Iterator iterator) throws ExceptionVariable

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.variables.VariableObject.getStringValue () throws ExceptionType, ExceptionVariable

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

Object cmas.ontology.variables.VariableObject.getValue ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

boolean cmas.ontology.variables.VariableObject.hasNext (Iterator iterator)

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.Variable.Serialize () [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setId () [inherited]

setId will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setId (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

**void cmas.ontology.variables.VariableObject.setValue (Object newValue) throws
ExceptionType, ExceptionVariable, ExceptionNullValue, ExceptionConstant**

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

**Object cmas.ontology.variables.VariableObject.valueOf (String value) throws
ExceptionVariable, ExceptionType, ExceptionNullValue, ExceptionConstant**

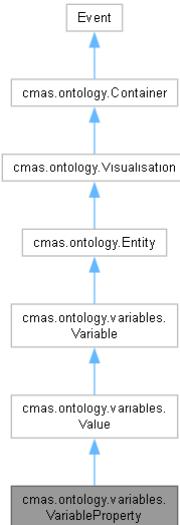
Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- VariableObject.java

cmas.ontology.variables.VariableProperty Class Reference

Inheritance diagram for cmas.ontology.variables.VariableProperty:



Public Member Functions

- **VariableProperty (Property<?> Prop)** throws **ExceptionType, ExceptionConstant**
- **String getStringValue ()** throws **ExceptionType**
- **Object getCopyOfValue ()** throws **ExceptionType**
- **Object getValue ()**
- **void setValue (Object newValue)** throws **ExceptionNullValue, ExceptionType, ExceptionConstant**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **Iterator getIterator ()**
- **ValueTypes getValueType ()**
- **void setReferenceProperty (Property<?> prop)**
- **Property<?> getReferenceProperty ()**
- **boolean isReferenceProperty ()**
- **Boolean isBoolean ()**
- **Boolean isInteger ()**
- **Boolean isReal ()**
- **Boolean isString ()**
- **Object valueOf (String s)**
- **String getUnit ()**
- **void setUnit (String value)**
- **String getUpperBound ()**
- **void setUpperBound (String upperBound)**
- **String getLowerBound ()**
- **void setLowerBound (String lowerBound)**
- **boolean isWithinRange (Object testValue)** throws **ExceptionType**
- **Integer parseInteger (String Value, Integer Default)**
- **boolean parseBoolean (String Value, Boolean Default)**
- **Double parseReal (String Value, Double Default)**
- **JsonObject Serialize ()**
- **Variable getIteratorObject ()**
- **boolean hasNext (Iterator iterator)**
- **Object getNext (Iterator iterator)** throws **ExceptionVariable, ExceptionType, ExceptionConstant**
- **TypeOfVariable getVariableType ()**

- Boolean **isArray** ()
- Boolean **isPrimitive** ()
- Boolean **isObject** ()
- Boolean **getReadOnly** ()
- Boolean **hasAdapter** ()
- void **setReadOnly** (Boolean ReadOnly)
- Boolean **getStoreLongTerm** ()
- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable** **getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType** **getType** ()
- **Properties** **getProperties** ()
- **Property<?>** **getProperty** (String Name)
- **Property<?>** **GetPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, **Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- LinkedHashMap< String, ArrayList< **RelationObject** > > **getRealtions** ()
- Collection< ArrayList< **RelationObject** > > **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- **Entity** **findRelationByName** (String RelationName, String EntityName)
- **Entity** **findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (**Entity** ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- **Relations** **getRealtionsObject** ()

- `ArrayList< BreakPoint > getBreakPointList ()`
- `void setBreakPointList (ArrayList< BreakPoint > breakPointList)`
- `Boolean isInstance ()`

Static Public Member Functions

- `static ValueTypes getValueType (JsonObject job) throws ExceptionType`
- `static Entity Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable`
- `static Entity Factory (String job) throws ExceptionType, ExceptionConstant, ExceptionVariable`
- `static TypeOfVariable getVariableType (JsonObject job) throws ExceptionType`
- `static TypeOfVariable getTypeOfVariable (JsonObject job) throws ExceptionType`

Protected Member Functions

- `void setIgnoreReadOnly (boolean value)`

Protected Attributes

- `Property<?> property`
- `Property<?> ReferenceProperty`
- `ValueTypes ValueType`
- `String UpperBound`
- `String LowerBound`
- `String Unit`
- `ValueRange upperRangeExpression`
- `ValueRange lowerRangeExpression`
- `Boolean ignoreReadOnly =true`
- `TypeOfVariable VariableType`
- `Boolean ReadOnly`
- `Boolean StoreLongTerm`
- `Boolean Error`
- `String Address`
- `String Source`
- `Adapter VariableAdapter`
- `Variable Reference`
- `Properties properties`
- `Relations relations`

Static Protected Attributes

- `static final Boolean ignoreTypeCheck =false`
- `static final Boolean ignoreRangeCheck =false`

Member Function Documentation

Entity cmas.ontology.variables.VariableProperty.copy (boolean instance)

Must be override by all classes extending `Entity` if instance is true `Properties` will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<code>instance</code>	if true a runtime entity is create otherwise a modelling entity
-----------------------	---

Reimplemented from `cmas.ontology.variables.Variable` (*p.333*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from `cmas.ontology.Container` (*p.161*).

void cmas.ontology.Entity.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Value.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static], [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.VariableProperty.getCopyOfValue () throws ExceptionType

Reimplemented from **cmas.ontology.variables.Value** (*p.323*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableProperty.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableProperty.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableProperty.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

String cmas.ontology.variables.VariableProperty.getStringValue () throws ExceptionType

Reimplemented from **cmas.ontology.variables.Value** (*p.323*).

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

Object cmas.ontology.variables.VariableProperty.getValue ()

Reimplemented from **cmas.ontology.variables.Value** (*p.323*).

JsonObject cmas.ontology.variables.Value.Serialize () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

void cmas.ontology.variables.VariableProperty.setValue (Object newValue) throws ExceptionNullValue, ExceptionType, ExceptionConstant

Reimplemented from **cmas.ontology.variables.Value** (*p.323*).

Object `cmas.ontology.variables.Value.valueOf (String s)` [inherited]

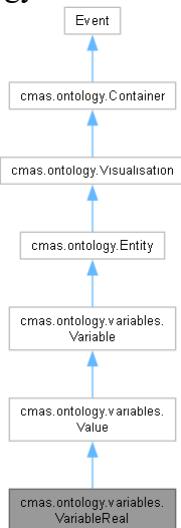
Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- VariableProperty.java

cmas.ontology.variables.VariableReal Class Reference

Inheritance diagram for cmas.ontology.variables.VariableReal:



Public Member Functions

- **VariableReal (String Name)** throws **ExceptionType, ExceptionConstant**
- **VariableReal (String Name, Double DefaultValue)** throws **ExceptionType, ExceptionConstant**
- **VariableReal (VariableReal variable, boolean instance)** throws **ExceptionType**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **Iterator getIterator ()**
- **Object getCopyOfValue ()** throws **ExceptionType**
- **String getStringValue ()** throws **ExceptionType**
- **ValueTypes getValueType ()**
- **void setReferenceProperty (Property<?> prop)**
- **Property<?> getReferenceProperty ()**
- **boolean isReferenceProperty ()**
- **Boolean isBoolean ()**
- **Boolean isInteger ()**
- **Boolean isReal ()**
- **Boolean isString ()**
- **Object getValue ()**
- **void setValue (Object newValue)** throws **ExceptionNullValue, ExceptionType, ExceptionConstant**
- **Object valueOf (String s)**
- **String getUnit ()**
- **void setUnit (String value)**
- **String getUpperBound ()**
- **void setUpperBound (String upperBound)**
- **String getLowerBound ()**
- **void setLowerBound (String lowerBound)**
- **boolean isWithinRange (Object testValue)** throws **ExceptionType**
- **Integer parseInteger (String Value, Integer Default)**
- **boolean parseBoolean (String Value, Boolean Default)**
- **Double parseReal (String Value, Double Default)**
- **JsonObject Serialize ()**
- **Variable getIteratorObject ()**
- **boolean hasNext (Iterator iterator)**

- Object **getNext** (Iterator iterator) throws **ExceptionVariable**, **ExceptionType**, **ExceptionConstant**
- **TypeOfVariable getVariableType** ()
- Boolean **isArray** ()
- Boolean **isPrimitive** ()
- Boolean **isObject** ()
- Boolean **getReadOnly** ()
- Boolean **hasAdapter** ()
- void **setReadOnly** (Boolean ReadOnly)
- Boolean **getStoreLongTerm** ()
- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, **Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- LinkedHashMap< String, ArrayList< **RelationObject** >> **getRealitions** ()
- Collection< ArrayList< **RelationObject** >> **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- **Entity findRelationByName** (String RelationName, String EntityName)
- **Entity findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (**Entity** ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)

- String **getClassName** ()
- void **setClassName** (String newName)
- **Relations getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **ValueTypes getValueType** (JsonObject job) throws **ExceptionType**
- static **Entity Factory** (JsonObject job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **Entity Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **TypeOfVariable getVariableType** (JsonObject job) throws **ExceptionType**
- static **TypeOfVariable getTypeOfVariable** (JsonObject job) throws **ExceptionType**

Protected Member Functions

- void **setIgnoreReadOnly** (boolean value)

Protected Attributes

- **Property<?> ReferenceProperty**
- **ValueTypes ValueType**
- String **UpperBound**
- String **LowerBound**
- String **Unit**
- **ValueRange upperRangeExpression**
- **ValueRange lowerRangeExpression**
- Boolean **ignoreReadOnly** =true
- **TypeOfVariable VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- **Adapter VariableAdapter**
- **Variable Reference**
- **Properties properties**
- **Relations relations**

Static Protected Attributes

- static final Boolean **ignoreTypeCheck** =false
- static final Boolean **ignoreRangeCheck** =false

Member Function Documentation

Entity cmas.ontology.variables.VariableReal.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.variables.Variable** (*p.333*).

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Value.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static], [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.Value.getCopyOfValue () throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableReal.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableReal.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableReal.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

**String cmas.ontology.variables.Value.getStringValue () throws
ExceptionType [inherited]**

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType () [inherited]**Returns**

entity type.

Object cmas.ontology.variables.Value.getValue () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.Value.Serialize () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

Referenced by **Entity()**.

```
void cmas.ontology.variables.Value.setValue (Object newValue) throws  
ExceptionNullValue, ExceptionType, ExceptionConstant [inherited]
```

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

```
Object cmas.ontology.variables.Value.valueOf (String s) [inherited]
```

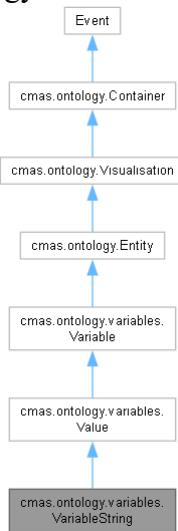
Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

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

- VariableReal.java

cmas.ontology.variables.VariableString Class Reference

Inheritance diagram for cmas.ontology.variables.VariableString:



Public Member Functions

- `VariableString (String Name)` throws `ExceptionType, ExceptionConstant`
- `VariableString (String Name, String DefaultValue)` throws `ExceptionType, ExceptionConstant`
- `VariableString (VariableString variable, boolean instance)` throws `ExceptionType`
- `Entity copy (boolean instance)`
- `String getEntityDescription ()`
- `String getEntityIconFile ()`
- `Iterator getIterator ()`
- `boolean hasNext (Iterator iterator)`
- `Object getNext (Iterator iterator)` throws `ExceptionVariable, ExceptionType, ExceptionConstant`
- `Variable getIteratorObject ()`
- `Object getCopyOfValue ()` throws `ExceptionType`
- `String getStringValue ()` throws `ExceptionType`
- `ValueTypes getValueType ()`
- `void setReferenceProperty (Property<?> prop)`
- `Property<?> getReferenceProperty ()`
- `boolean isReferenceProperty ()`
- `Boolean isBoolean ()`
- `Boolean isInteger ()`
- `Boolean isReal ()`
- `Boolean isString ()`
- `Object getValue ()`
- `void setValue (Object newValue)` throws `ExceptionNullValue, ExceptionType, ExceptionConstant`
- `Object valueOf (String s)`
- `String getUnit ()`
- `void setUnit (String value)`
- `String getUpperBound ()`
- `void setUpperBound (String upperBound)`
- `String getLowerBound ()`
- `void setLowerBound (String lowerBound)`
- `boolean isWithinRange (Object testValue)` throws `ExceptionType`
- `Integer parseInteger (String Value, Integer Default)`
- `boolean parseBoolean (String Value, Boolean Default)`

- Double **parseReal** (String Value, Double Default)
- JsonObject **Serialize** ()
- **TypeOfVariable getVariableType** ()
- Boolean **isArray** ()
- Boolean **isPrimitive** ()
- Boolean **isObject** ()
- Boolean **getReadOnly** ()
- Boolean **hasAdapter** ()
- void **setReadOnly** (Boolean ReadOnly)
- Boolean **getStoreLongTerm** ()
- void **setStoreLongTerm** (Boolean storeLongTerm)
- Boolean **getError** ()
- void **setError** (Boolean error)
- String **getAddress** ()
- void **setAddress** (String address)
- String **getSource** ()
- void **setSource** (String source)
- void **setReference** (**Variable** ref)
- **Variable getReference** ()
- boolean **isReference** ()
- boolean **compare** (Object o)
- boolean **isVariableObjectMatching** (**VariableObject** variable1, **VariableObject** variable2)
- void **delete** ()
- String **toString** ()
- String **getID** ()
- void **setID** (String ID)
- void **setID** ()
- String **getName** ()
- void **setName** (String Name)
- String **getParentName** ()
- void **setParent** (Entity ent)
- String **getDescription** ()
- void **setDescription** (String Description)
- String **getInstanceID** ()
- **EntityType getType** ()
- **Properties getProperties** ()
- **Property<?> getProperty** (String Name)
- **Property<?> getPropertyIgnoreCase** (String Name)
- void **addProperty** (**Property<?>** Item)
- boolean **isCompatibleRelation** (String Name, **Entity** ent)
- boolean **hasRelation** (String Name, **Entity** ent)
- LinkedHashMap< String, ArrayList< **RelationObject** >> **getRealitions** ()
- Collection< ArrayList< **RelationObject** >> **getAllRelations** ()
- ArrayList< **RelationObject** > **getSpecificRelations** (String Name)
- **Entity findRelationByName** (String RelationName, String EntityName)
- **Entity findRelationByNameIgnoreCase** (String RelationName, String EntityName)
- ArrayList< Entity > **getSpecificRelationType** (String Name)
- void **addSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (String Name, **Entity** ent)
- void **removeSpecificRelation** (**Entity** ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)

- String **getClassName** ()
- void **setClassName** (String newName)
- **Relations getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static **ValueTypes getValueType** (JsonObject job) throws **ExceptionType**
- static **Entity Factory** (JsonObject job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **Entity Factory** (String job) throws **ExceptionType, ExceptionConstant, ExceptionVariable**
- static **TypeOfVariable getVariableType** (JsonObject job) throws **ExceptionType**
- static **TypeOfVariable getTypeOfVariable** (JsonObject job) throws **ExceptionType**

Protected Member Functions

- void **setIgnoreReadOnly** (boolean value)

Protected Attributes

- **Property<?> ReferenceProperty**
- **ValueTypes ValueType**
- String **UpperBound**
- String **LowerBound**
- String **Unit**
- **ValueRange upperRangeExpression**
- **ValueRange lowerRangeExpression**
- Boolean **ignoreReadOnly** =true
- **TypeOfVariable VariableType**
- Boolean **ReadOnly**
- Boolean **StoreLongTerm**
- Boolean **Error**
- String **Address**
- String **Source**
- **Adapter VariableAdapter**
- **Variable Reference**
- **Properties properties**
- **Relations relations**

Static Protected Attributes

- static final Boolean **ignoreTypeCheck** =false
- static final Boolean **ignoreRangeCheck** =false

Member Function Documentation

Entity cmas.ontology.variables.VariableString.copy (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.variables.Variable** (*p.333*).

void cmas.ontology.Entity.delete ()[inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.Deserialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.variables.Value.Factory (JsonObject job) throws ExceptionType, ExceptionConstant, ExceptionVariable [static], [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Object cmas.ontology.variables.Value.getCopyOfValue () throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.variables.VariableString.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.variables.VariableString.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

Iterator cmas.ontology.variables.VariableString.getIterator ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

Variable cmas.ontology.variables.VariableString.getIteratorObject ()

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

Object cmas.ontology.variables.VariableString.getNext (Iterator iterator) throws ExceptionVariable, ExceptionType, ExceptionConstant

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

String cmas.ontology.variables.Value.getStringValue () throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

Object cmas.ontology.variables.Value.getValue () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

boolean cmas.ontology.variables.VariableString.hasNext (Iterator iterator)

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

JsonObject cmas.ontology.variables.Value.Serialize () [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (*p.331*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

Referenced by **Entity()**.

void cmas.ontology.variables.Value.setValue (Object newValue) throws ExceptionNullValue, ExceptionType, ExceptionConstant [inherited]

Reimplemented from **cmas.ontology.variables.Variable** (p.331).

Object cmas.ontology.variables.Value.valueOf (String s) [inherited]

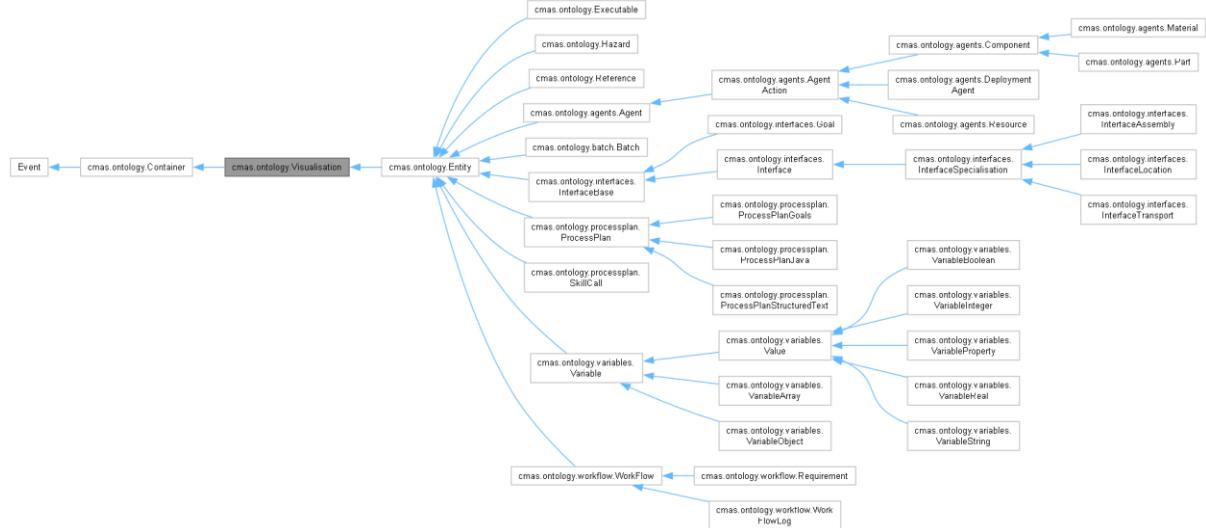
Reimplemented from **cmas.ontology.variables.Variable** (p.331).

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

- VariableString.java

cmas.ontology.Visualisation Class Reference

Inheritance diagram for cmas.ontology.Visualisation:



Public Member Functions

- **Visualisation** (EventType TypeOfEvent)
- **Visualisation** (JsonObject job) throws **ExceptionType**
- **Visualisation** (**Visualisation** visualisation, boolean instance)
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (**Container** container, boolean instance)
- void **delete** ()
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClassName** ()
- void **setClassName** (String newName)
- **Relations** **getRealtionsObject** ()
- JsonObject **Serialize** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Protected Attributes

- Properties **properties**
- Relations **relations**

Package Functions

- void **DeSerialize** (JsonObject job) throws **ExceptionType**

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

- Visualisation.java

cmas.ontology.adapter.websocket.WebSocketClient Class Reference

Inheritance diagram for cmas.ontology.adapter.websocket.WebSocketClient:



Public Member Functions

- **WebSocketClient (String Source)**
- **void setSource (String Source)**
- **void connectToServer ()**
- **void onMessage (String message)**
- **void onOpen (Session session)**
- **void onClose ()**
- **String ReadData (String Address, String Name, Object Value) throws Exception**
- **void WriteData (String Address, String Name, Boolean bValue) throws Exception**
- **void WriteData (String Address, String Name, String sValue) throws Exception**
- **void WriteData (String Address, String Name, Integer iValue) throws Exception**
- **void WriteData (String Address, String Name, Double dValue) throws Exception**
- **boolean close ()**
- **void setDebug (boolean bValue)**
- **String getDefaultValue ()**
- **String getDescription ()**
- **boolean isValid (String Source)**
- **Adapter createNew (String Source)**

Member Function Documentation

boolean cmas.ontology.adapter.websocket.WebSocketClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.websocket.WebSocketClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.websocket.WebSocketClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.websocket.WebSocketClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.websocket.WebSocketClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.websocket.WebSocketClient.ReadData (String Address, String Name, Object Value) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.websocket.WebSocketClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.websocket.WebSocketClient.WriteData (String Address, String Name, Boolean bValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.websocket.WebSocketClient.WriteData (String Address, String Name, Double dValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.websocket.WebSocketClient.WriteData (String Address, String Name, Integer iValue) throws Exception

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.websocket.WebSocketClient.WriteData (String Address, String Name, String sValue) throws Exception

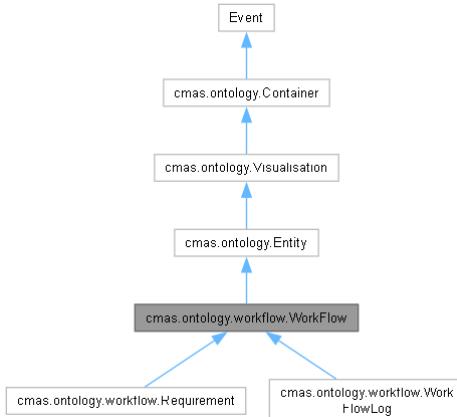
Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- `WebSocketClient.java`

cmas.ontology.workflow.WorkFlow Class Reference

Inheritance diagram for cmas.ontology.workflow.WorkFlow:



Classes

- enum **SpecificTypes** enum **WorkFlowTypes**

Public Member Functions

- **WorkFlow (SpecificTypes type)**
- **WorkFlow (WorkFlow entity, boolean instance)**
- **WorkFlowTypes getWorkFlowType ()**
- **void setWorkFlowType (WorkFlowTypes workFlowType)**
- **SpecificTypes getSpecificType ()**
- **JsonObject Serialize ()**
- **abstract Entity copy (boolean instance)**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRealitions ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**
- **ArrayList< Entity > getSpecificRelationType (String Name)**
- **void addSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (String Name, Entity ent)**
- **void removeSpecificRelation (Entity ent)**

- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- abstract String **getEntityDescription** ()
- abstract String **getEntityIconFile** ()
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass Name** ()
- void **setClass Name** (String newName)
- Relations **getRealtionsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static SpecificTypes **getSpecificType** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**

Protected Attributes

- Properties **properties**
- Relations **relations**

Member Function Documentation

abstract Entity cmas.ontology.Entity.copy (boolean instance) [abstract], [inherited]

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
Reimplemented in cmas.ontology.agents.Material	cmas.ontology.agents.DeploymentAgent (p.174), (p.237), cmas.ontology.agents.Part (p.255),
cmas.ontology.agents.Resource	(p.298), cmas.ontology.batch.Batch (p.151),
cmas.ontology.Executable	(p.187), cmas.ontology.Hazard (p.199),
cmas.ontology.interfaces.Goal	(p.195), cmas.ontology.interfaces.Interface (p.204),
cmas.ontology.interfaces.InterfaceAssembly	(p.209),
cmas.ontology.interfaces.InterfaceLocation	(p.219),
cmas.ontology.interfaces.InterfaceSpecialisation	(p.224),
cmas.ontology.interfaces.InterfaceTransport	(p.230),
cmas.ontology.processplan.ProcessPlanGoals	(p.266),
cmas.ontology.processplan.ProcessPlanJava	(p.272),
cmas.ontology.processplan.ProcessPlanStructuredText	(p.278),
cmas.ontology.processplan.SkillCall	(p.308), cmas.ontology.Reference (p.287),
cmas.ontology.variables.Variable (p.333), cmas.ontology.variables.VariableArray	(p.337),
cmas.ontology.variables.VariableBoolean (p.343), cmas.ontology.variables.VariableInteger (p.349), cmas.ontology.variables.VariableObject	(p.356),
cmas.ontology.variables.VariableProperty (p.362), cmas.ontology.variables.VariableReal (p.368), cmas.ontology.variables.VariableString	(p.374),

cmas.ontology.workflow.Requirement (*p.293*), and **cmas.ontology.workflow.WorkFlowLog** (*p.386*).

References **Entity()**.

void cmas.ontology.Entity.delete () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void cmas.ontology.Entity.DeSerialize (JsonObject job) throws ExceptionType [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity cmas.ontology.workflow.WorkFlow.Factory (JsonObject job) throws ExceptionType [static]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.workflow.WorkFlow.Serialize ()

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by **Entity()**.

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (*p.289*), and **cmas.ontology.variables.VariableArray** (*p.339*).

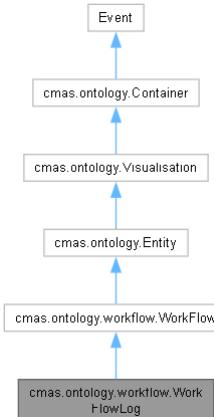
Referenced by **Entity()**.

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

- [WorkFlow.java](#)

cmas.ontology.workflow.WorkFlowLog Class Reference

Inheritance diagram for cmas.ontology.workflow.WorkFlowLog:



Public Member Functions

- **WorkFlowLog (WorkFlowTypes type, String Name, String Target)**
- **WorkFlowLog (WorkFlowLog entity, boolean instance)**
- **String getTime ()**
- **void setTime (String time)**
- **void setTime ()**
- **String getTarget ()**
- **void setTarget (String target)**
- **Entity copy (boolean instance)**
- **String getEntityDescription ()**
- **String getEntityIconFile ()**
- **SpecificTypes getSpecificType ()**
- **WorkFlowTypes getWorkFlowType ()**
- **void setWorkFlowType (WorkFlowTypes workFlowType)**
- **JsonObject Serialize ()**
- **void delete ()**
- **String toString ()**
- **String getID ()**
- **void setID (String ID)**
- **void setID ()**
- **String getName ()**
- **void setName (String Name)**
- **String getParentName ()**
- **void setParent (Entity ent)**
- **String getDescription ()**
- **void setDescription (String Description)**
- **String getInstanceID ()**
- **EntityType getType ()**
- **Properties getProperties ()**
- **Property<?> getProperty (String Name)**
- **Property<?> getPropertyIgnoreCase (String Name)**
- **void addProperty (Property<?> Item)**
- **boolean isCompatibleRelation (String Name, Entity ent)**
- **boolean hasRelation (String Name, Entity ent)**
- **LinkedHashMap< String, ArrayList< RelationObject > > getRealitions ()**
- **Collection< ArrayList< RelationObject > > getAllRelations ()**
- **ArrayList< RelationObject > getSpecificRelations (String Name)**
- **Entity findRelationByName (String RelationName, String EntityName)**
- **Entity findRelationByNameIgnoreCase (String RelationName, String EntityName)**
- **ArrayList< Entity > getSpecificRelationType (String Name)**

- void **addSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (String Name, Entity ent)
- void **removeSpecificRelation** (Entity ent)
- void **eventReflection** (EventType TypeOfEvent, EventCause CauseOfEvent, Object Argument, Object anOwner)
- void **DeSerialize** (JsonObject job) throws **ExceptionType**
- ImageView **getEntityIcon** ()
- void **copyProperties** (Container container, boolean instance)
- void **setOverride** (Boolean value)
- Boolean **getProtected** ()
- void **setProtected** (Boolean protectedValue)
- Object **getUserData** ()
- void **setUserData** (Object userData)
- String **getClass_Name** ()
- void **setClassName** (String newName)
- Relations **getRealtonsObject** ()
- ArrayList< BreakPoint > **getBreakPointList** ()
- void **setBreakPointList** (ArrayList< BreakPoint > breakPointList)
- Boolean **isInstance** ()

Static Public Member Functions

- static Entity **Factory** (JsonObject job) throws **ExceptionType**
- static Entity **Factory** (String job) throws **ExceptionType**, **ExceptionConstant**, **ExceptionVariable**
- static SpecificTypes **getSpecificType** (JsonObject job) throws **ExceptionType**

Protected Attributes

- Properties properties
- Relations relations

Member Function Documentation

Entity **cmas.ontology.workflow.WorkFlowLog.copy** (boolean instance)

Must be override by all classes extending **Entity** if instance is true **Properties** will be created without a link to parent (parent value will be used) and relations without a link.

Parameters

<i>instance</i>	if true a runtime entity is create otherwise a modelling entity
-----------------	---

Reimplemented from **cmas.ontology.Entity** (*p.382*).

void **cmas.ontology.Entity.delete** () [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

void **cmas.ontology.Entity.DeSerialize** (JsonObject job) throws **ExceptionType** [inherited]

Reimplemented from **cmas.ontology.Container** (*p.161*).

Entity **cmas.ontology.workflow.WorkFlow.Factory** (JsonObject job) throws **ExceptionType** [static], [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

String cmas.ontology.Entity.getDescription () [inherited]

Returns

the entity description.

String cmas.ontology.workflow.WorkFlowLog.getEntityDescription ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.workflow.WorkFlowLog.getEntityIconFile ()

Reimplemented from **cmas.ontology.Visualisation** (*p.378*).

String cmas.ontology.Entity.getID () [inherited]

Get the ID of the entity.

Returns

a string with current ID.

String cmas.ontology.Entity.getInstanceID () [inherited]

A unique runtime ID. The InstanceID is generated automatically by the system when an entity is instantiated.

Returns

runtime instance id

String cmas.ontology.Entity.getName () [inherited]

get current name of the entity.

Returns

current name.

Reimplemented in **cmas.ontology.Reference** (*p.288*).

Referenced by **cmas.ontology.processplan.ProcessPlanStructuredText.run()**, and **cmas.ontology.variables.VariableArray.setName()**.

EntityType cmas.ontology.Entity.getType () [inherited]

Returns

entity type.

JsonObject cmas.ontology.workflow.WorkFlow.Serialize () [inherited]

Reimplemented from **cmas.ontology.Entity** (*p.178*).

void cmas.ontology.Entity.setDescription (String Description) [inherited]

Description is a way for the user to document the entity.

Parameters

<i>Description</i>	a user defined description
--------------------	----------------------------

void cmas.ontology.Entity.setID () [inherited]

setID will set a new ID by generating a world" unique identifier.

Referenced by Entity().

void cmas.ontology.Entity.setID (String ID) [inherited]

Will set a specific ID.

Parameters

<i>ID</i>	the new ID value.
-----------	-------------------

void cmas.ontology.Entity.setName (String Name) [inherited]

set entity name.

Parameters

<i>Name</i>	the new name.
-------------	---------------

Reimplemented in **cmas.ontology.Reference** (p.289), and **cmas.ontology.variables.VariableArray** (p.339).

Referenced by Entity().

void cmas.ontology.workflow.WorkFlowLog.setTime ()

Will use the local time to set the time with the following format yyyy-MM-dd HH:mm:ss:SSS

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

- WorkFlowLog.java

cmas.ontology.workflow.WorkFlow.WorkFlowTypes Enum Reference

Public Attributes

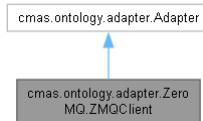
- **NONE**
- **SKILL**
- **CALL**
- **GOAL**
- **ATTACHED**
- **DETACHED**
- **ASSEMBLED**
- **TRANSFERED**
- **CREATED**
- **DEPLOYED**
- **FINISHED**
- **FAILED**
- **UNDEPLOYED**
- **INDEPOT**

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

- WorkFlow.java

cmas.ontology.adapter.ZeroMQ.ZMQClient Class Reference

Inheritance diagram for cmas.ontology.adapter.ZeroMQ.ZMQClient:



Public Member Functions

- **ZMQClient** (String Source)
- void **SetSource** (String Source)
- String **ReadData** (String Address, String Name, Object Value)
- void **WriteData** (String Address, String Name, Boolean bValue)
- void **WriteData** (String Address, String Name, String sValue)
- void **WriteData** (String Address, String Name, Integer iValue)
- void **WriteData** (String Address, String Name, Double dValue)
- void **Write** (String string)
- boolean **close** ()
- void **setDebug** (boolean bValue)
- String **getDefaultValue** ()
- String **getDescription** ()
- boolean **isValid** (String Source)
- **Adapter** **createNew** (String Source)

Detailed Description

ZeroMQ communications for variables

```
For adapter type of ZeroMQ the specific format is: ZMQ.type:[TCP]:[//host]:[port]
eg, ZMQ:tcp://*:5555
```

Type can one of: DEALER, PAIR, PUB, PULL, PUSH, REP, REQ, ROUTER, STREAM, SUB, XPUB, XSUB

Author

Fredrik Danielsson, University West

Member Function Documentation

boolean cmas.ontology.adapter.ZeroMQ.ZMQClient.close ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

Adapter cmas.ontology.adapter.ZeroMQ.ZMQClient.createNew (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.ZeroMQ.ZMQClient.getDefaultValue ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.ZeroMQ.ZMQClient.getDescription ()

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

boolean cmas.ontology.adapter.ZeroMQ.ZMQClient.isValid (String Source)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

String cmas.ontology.adapter.ZeroMQ.ZMQClient.ReadData (String Address, String Name, Object Value)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.ZeroMQ.ZMQClient.setDebug (boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.ZeroMQ.ZMQClient.WriteData (String Address, String Name, Boolean bValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.ZeroMQ.ZMQClient.WriteData (String Address, String Name, Double dValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.ZeroMQ.ZMQClient.WriteData (String Address, String Name, Integer iValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

void cmas.ontology.adapter.ZeroMQ.ZMQClient.WriteData (String Address, String Name, String sValue)

Implements **cmas.ontology.adapter.Adapter** (*p.135*).

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

- ZMQClient.java

Index

INDEX