

# User Manual 4.5 Properties and models: Sensors

De Wiki

Aller à : [navigation](#), [rechercher](#)

[User Manual 4.5 Properties and models: Sensors](#)

## Sommaire

- [1 Introduction](#)
  - [1.1 Scope](#)
  - [1.2 Javadoc](#)
  - [1.3 Links](#)
  - [1.4 Useful Documents](#)
  - [1.5 Package overview](#)
- [2 Features Description](#)
  - [2.1 Properties](#)
    - [2.1.1 Sensor Property](#)
    - [2.1.2 Geometric property](#)
  - [2.2 Sensor model](#)
  - [2.3 Fields of view](#)
- [3 Getting started](#)
- [4 Contents](#)
  - [4.1 Interfaces](#)
  - [4.2 Classes](#)

## Introduction

### Scope

In this section are presented the sensor model for a PATRIUS assembly, the associated part property and the fields of view that are necessary to describe it.

### Javadoc

The [SensorModel](#) is available in the package [fr.cnes.sirius.patrius.assembly.models](#).

The [SensorProperty](#) is available in the package [fr.cnes.sirius.patrius.assembly.properties](#).

The fields of view are available in the package [fr.cnes.sirius.patrius.fieldsofview](#).

### Links

None as of now.

## Useful Documents

None as of now.

## Package overview

The sensor model is associated to one part of an assembly that must have a "SensorProperty" property.

Its main and inhibition targets are several PVCoordinatesProvider (and their radiuses are doubles), its main and inhibition fields of view are under the IFieldOfView interface, in the "fieldsofview" package.



## Features Description

### Properties

#### Sensor Property

To create a sensor model, one part of the assembly must be associated to a SENSOR property.

A sensor property contains :

- a sight axis (not optional)
- (optional) a main spherical target defined by its center and radius (that can be set to zero to create a simple point)
- (optional)if a main target has been defined : a main field of view
- (optional) two arrays (same length) for the inhibition fields and the associated targets (as the main one, they are spherical, with radiuses that can be set to zero to create a simple point)
- (optional) an array of reference axis

All of them are expressed in the sensor part/frame.

This property shall be used through the Sensor Model.

The property type associated is **SENSOR**.

#### Geometric property

The Geometric property associates a part to any shape that implements the [SolidShape](#) interface. (See the [MAT\_GEO\_Home Geometry] chapter of the Mathematics user manual for more details)  
It is used in masking computations.

The property type associated is **GEOMETRY**.

### Sensor model

An instance of the sensor model is associated to one part, that contains a SENSOR property. It realizes the useful computations about this particular sensor : check if the main target is in its field of view, if inhibition or maskings happen, etc...

Here is a complete list of those services. The model provides methods to :

- Check if the main spherical target is in the field of view at a date

Important : for "is in" boolean tests, all spherical targets (main and inhibition) are considered to be in the associated field as soon as a part of them is in.

```
// the model is build from an assembly, giving the name of the part that contains the SENSOR property.
```

```
SensorModel sensor = new SensorModel(assembly, partName);  
boolean targetIsInField = sensor.isMainTargetInField(date);
```

- Check if at least an inhibition spherical target is in its associated inhibition field at a date

```
boolean noInhibition = sensor.noInhibition(date);
```

- Check if the main spherical target is in the field of view AND no inhibition target in its inhibition field at a given date

```
boolean visibilityIsOk = sensor.visibilityOk(date);
```

- Computes the angular distance of the main target CENTER to the border of the main field of view at a date.

```
double angularDistance = sensor.getTargetCenterFOVAngle(date);
```

The result is positive if the target is in the field. NB : in some particular cases of Boolean Field of view, this angular distance is approximated (but the sign is still right).

- Computes the main spherical target direction vector in the part's frame (CX, CY, CZ)

```
Vector3D targetInFrame = sensor.getTargetVectorInSensorFrame(date);
```

- Computes the dihedral angles ( $AX = \text{atan2}(CZ, CY)$ ,  $AY = \text{atan2}(CX, CZ)$ ,  $AZ = \text{atan2}(CY, CX)$ )

```
// the order in this array is : AX, AY, AZ
```

```
double[] angles = sensor.getTargetDihedralAngles(date);
```

- Computes the vector angles (of the main target to the sight axis or one of the reference axis)

```
// to the sight axis
```

```
double angleSightAxis = sensor.getTargetSightAxisAngle(date);
```

```
// to the reference axis number N (for the first N = 1 !!)
```

```
double angleRefAxis = sensor.getTargetRefAxisAngle(date, N);
```

- Computes the elevation angles (of the main target to the plane orthogonal to the sight axis or to one of the reference axis)

```

// to the sight axis
double angleSightAxis = sensor.getTargetSightAxisElevation(date);

// to the reference axis number N (for the first N = 1 !!)
double angleRefAxis = sensor.getTargetRefAxisElevation(date, N);

• Set potentially masking objects and test the masking at a date. Those objects can be parts of the
  same spacecraft (the one carrying the sensor), parts of [MIS_SENSORS_SecondSpc secondary
  spacecrafts] or [MIS_SENSORS_PatriusBodySpheroid celestial bodies].

// two potentially masking celestial bodies
GeometricBodyShape earth = new ExtendedOneAxisEllipsoid(aeEarth, fEarth,
attachedFrameEarth, "earth");
GeometricBodyShape moon= new ExtendedOneAxisEllipsoid(aeMoon, fMoon,
attachedFrameMoon, "moon");

sensor.addMaskingCelestialBody(earth);
sensor.addMaskingCelestialBody(moon);

// a potentially masking spacecraft (the concerned Assembly's parts must have
the right GEOMETRY property)
SecondarySpacecraft issStation = new SecondarySpacecraft (assemblyISS,
propagatorISS, "ISS");

// its potentially masking parts names
String[] maskingPartsISS = {"solar panel 1", "solar panel 2"};

sensor.addSecondaryMaskingSpacecraft(issStation, maskingPartsISS );

// the same spacecraft's potentially masking parts (that must have each a
GEOMERTY property)
String[] maskingPartsSameSpacecraft = {"solar panel", "big antenna"};

sensor.addOwnMaskingParts(maskingPartsSameSpacecraft);

```

## Fields of view

See the dedicated [SPC\_FIELD\_mainPage Fields of view ] page.

## Getting started

[Modèle:SpecialInclusion prefix=\\$theme sub section="GettingStarted"/](#)

## Contents

### Interfaces

None as of now.

## Classes

Class	Summary	Javadoc
<b>SensorProperty</b>	This class is a part property for the PATRIUS assembly. It allows to define a part as a sensor, with associated fields and axis.	...
<b>GeometricProperty</b>	This class is a part property for the PATRIUS assembly. It allows to define the part geometry for masking computations.	...
<b>SensorModel</b>	This class is a model for a sensor integrated in a PATRIUS assembly.	...
<b>SecondarySpacecraft</b>	This class is a potentially sensor-masking secondary spacecraft.	...

See the dedicated [SPC\_FIELD\_mainPage Fields of view ] page for associated contents.

Récupérée de

«

[http://patrius.cnes.fr/index.php?title=User\\_Manual\\_4.5\\_Properties\\_and\\_models:\\_Sensors&oldid=2577](http://patrius.cnes.fr/index.php?title=User_Manual_4.5_Properties_and_models:_Sensors&oldid=2577) »

Catégorie :

- [User Manual 4.5 Spacecraft](#)

## Menu de navigation

### Outils personnels

- [3.145.196.150](#)
- [Discussion avec cette adresse IP](#)
- [Créer un compte](#)
- [Se connecter](#)

### Espaces de noms

- [Page](#)
- [Discussion](#)

### Variantes

### Affichages

- [Lire](#)
- [Voir le texte source](#)
- [Historique](#)
- [Exporter en PDF](#)

### Plus

## Rechercher

## PATRIUS

- [Welcome](#)

## Evolutions

- [Main differences between V4.15 and V4.14](#)
- [Main differences between V4.14 and V4.13](#)
- [Main differences between V4.13 and V4.12](#)
- [Main differences between V4.12 and V4.11](#)
- [Main differences between V4.11 and V4.10](#)
- [Main differences between V4.10 and V4.9](#)
- [Main differences between V4.9 and V4.8](#)
- [Main differences between V4.8 and V4.7](#)
- [Main differences between V4.7 and V4.6.1](#)
- [Main differences between V4.6.1 and V4.5.1](#)
- [Main differences between V4.5.1 and V4.4](#)
- [Main differences between V4.4 and V4.3](#)
- [Main differences between V4.3 and V4.2](#)
- [Main differences between V4.2 and V4.1.1](#)
- [Main differences between V4.1.1 and V4.1](#)
- [Main differences between V4.1 and V4.0](#)
- [Main differences between V4.0 and V3.4.1](#)

## User Manual

- [User Manual 4.15](#)
- [User Manual 4.14](#)
- [User Manual 4.13](#)
- [User Manual 4.12](#)
- [User Manual 4.11](#)
- [User Manual 4.10](#)
- [User Manual 4.9](#)
- [User Manual 4.8](#)
- [User Manual 4.7](#)
- [User Manual 4.6.1](#)
- [User Manual 4.5.1](#)
- [User Manual 4.4](#)
- [User Manual 4.3](#)
- [User Manual 4.2](#)
- [User Manual 4.1](#)
- [User Manual 4.0](#)

- [User Manual 3.4.1](#)
- [User Manual 3.3](#)

## Tutorials

- [Tutorials 4.15](#)
- [Tutorials 4.14](#)
- [Tutorials 4.13.5](#)
- [Tutorials 4.12.1](#)
- [Tutorials 4.8.1](#)
- [Tutorials 4.5.1](#)
- [Tutorials 4.4](#)
- [Tutorials 4.1](#)
- [Tutorials 4.0](#)

## Links

- [CNES freeware server](#)

## Navigation

- [Accueil](#)
- [Modifications récentes](#)
- [Page au hasard](#)
- [Aide](#)

## Outils

- [Pages liées](#)
- [Suivi des pages liées](#)
- [Pages spéciales](#)
- [Adresse de cette version](#)
- [Information sur la page](#)
- [Citer cette page](#)

• Dernière modification de cette page le 2 mai 2020 à 12:54.

- [Politique de confidentialité](#)
- [À propos de Wiki](#)
- [Avertissements](#)

- 