

Customize output variables

De Psimu

[Aller à la navigation](#) [Aller à la recherche](#)

La version imprimable n'est plus prise en charge et peut comporter des erreurs de génération.
Veuillez mettre à jour les signets de votre navigateur et utiliser à la place la fonction d'impression par défaut de celui-ci.

Warning: this section is only relevant for PSIMU version 11.4 (or more recent ones).

Create new specific variables in the output list is relatively simple. As an example, we are going to show how to add two additional variables **HPM** and **HAM**, corresponding respectively the mean perigee and apogee altitudes considering the mean semi-major axis and eccentricity already computed by **PSIMU**.



Sommaire

- [1 Creating a class implementing the INewVarsFunction interface](#)
 - [1.1 Adding a constructor](#)
 - [1.2 Implementing the addNewVars\(\) method](#)
 - [1.3 Implementing the computeNewVars\(\) method](#)
- [2 Giving information to PSIMU!](#)

Creating a class implementing the INewVarsFunction interface

As explained in the title of this paragraph, we have to create a new class implementing the [INewVarsFunction](#) interface where we will be able to compute the new variables. On the code below, we see that this interface forces us to have the [addNewVars\(\)](#), [computeNewVars\(\)](#) and [getNames\(\)](#) methods:

```
public class MyNewMeanVars implements INewVarsFunction {  
  
    @Override  
    public void addNewVars(ResultWriter resultWriter) throws SqliteException  
{  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void computeNewVars(ResultWriter resultWriter, String tableName,  
        SpacecraftState currentState,  
        HashMap<String, Object> currentVarsList) throws SqliteException {  
        // TODO Auto-generated method stub  
    }  
}
```

```

@Override
public List<String> getNames() {
    // TODO Auto-generated method stub
    return null;
}

}

```

Adding a constructor

As we will need information on the equatorial radius, we will add a constructor allowing to store a [ExtendedOneAxisEllipsoid](#) object. We will also add a list including the names of the new variables (this list will be returned by the [getNames\(\)](#) method:

```

public class MyNewMeanVars implements INewVarsFunction {
    private final ExtendedOneAxisEllipsoid earth;
    private final ArrayList<String> listOfNames;

    public MyNewMeanVars ( final ExtendedOneAxisEllipsoid earth ) {
        this.earth = earth;
        // List initialization
        listOfNames = new ArrayList<String>();
        listOfNames.add("HPM");
        listOfNames.add("HAM");

    }

    ...

    @Override
    public List<String> getNames() {
        return listOfNames;
    }
}

```

Implementing the [addNewVars\(\)](#) method

To fill the [addNewVars\(\)](#) method, we will have to call for the [addColumn\(\)](#) method giving for each variable and each table (ephemeris and events ones) as inputs:

- its name (String)
- its description (String)
- its unit (String)
- its gap threshold (for plotting; may be null)
- a boolean to know if it will be visible or not

```

public void addNewVars ( final ResultWriter resultWriter ) throws
SqliteException {

    if ( resultWriter != null ) {

```

```

        // Mean perigee altitude creation
        String varName = listOfNames.get(0);
        String varDesc = "Mean perigee altitude";
        String varUnit = "km";
        // Adding a column in the ephemeris table
        resultWriter.addColumn(PsimuUtils.EPH_TABLE, varName, varDesc,
ColumnType.REAL, varUnit, null, true);
        // Adding a column in the event table
        resultWriter.addColumn(PsimuUtils.EVENT_TABLE, varName, varDesc,
ColumnType.REAL, varUnit, null, true);
        // Mean apogee altitude creation
        varName = listOfNames.get(1);
        varDesc = "Mean apogee altitude";
        // Adding a column in the ephemeris table
        resultWriter.addColumn(PsimuUtils.EPH_TABLE, varName, varDesc,
ColumnType.REAL, varUnit, null, true);
        // Adding a column in the event table
        resultWriter.addColumn(PsimuUtils.EVENT_TABLE, varName, varDesc,
ColumnType.REAL, varUnit, null, true);
    }

}

```

Implementing the `computeNewVars()` method

Using the data previously computed as included in the **currentState** and **listOfCurrentVars** objects, we have just to compute the new variables then to store them using the `addValue()` method of the `ResultWriter` object.

```

public void computeNewVars(final ResultWriter resultWriter, final String
tableName,
    final SpacecraftState currentState,
    final HashMap<String, Object> listOfCurrentVars) throws
SQLException {
    // Getting previously computed mean semi major axis and eccentricity
    final double am = (Double)listOfCurrentVars.get("AM");
    final double em = (Double)listOfCurrentVars.get("EM");
    // Computation of the mean perigee and apogee altitude
    final double hpm = am*(1. - em) - earth.getEquatorialRadius();
    final double ham = am*(1. + em) - earth.getEquatorialRadius();
    // Storing in the table
    if ( resultWriter != null ) {
        resultWriter.addValue(tableName, listOfNames.get(0), hpm);
        resultWriter.addValue(tableName, listOfNames.get(1), ham);
    }
    // Storing in the current list
    if ( listOfCurrentVars != null ) {
        listOfCurrentVars.put("HPM", hpm);
        listOfCurrentVars.put("HAM", ham);
    }
}

```

}

Giving information to PSIMU!

All we need now is to call for the `addOutputVarsFunction()` method of the `OutputConfig` class with the object issued from our `MyNewMeanVars` class !

```
final OutputConfig output = new OutputConfig(...);  
output.addOutputVarsFunction(new MyNewMeanVars(EARTH));
```

Récupérée de « http://psimu.cnes.fr/index.php?title=Customize_output_variables&oldid=762 »

Menu de navigation

Outils personnels

- [Se connecter](#)

Espaces de noms

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

français

Affichages

- [Lire](#)
- [Voir le texte source](#)
- [Voir l'historique](#)
- [Exporter en PDF](#)

Plus

Rechercher

PSIMU

- [Welcome](#)
- [Quick start](#)
- [News](#)

GUI Mode

- [Overall presentation](#)
- [Initial Orbit](#)
- [Earth features](#)
- [Vehicle](#)
- [Forces](#)
- [Maneuvers](#)
- [Attitude](#)
- [Integrator](#)
- [Events](#)
- [Output](#)
- [Console](#)

Batch mode

- [How to call it](#)

Java interface

- [Basic principle](#)
- [Data initialization](#)
- [Propagation](#)
- [Printing results](#)
- [Customize output variables](#)

Evolutions

- [Main differences between V11.7.4 and V11.8](#)
- [Main differences between V11.7.3 and V11.7.4](#)
- [Main differences between V11.7.2 and V11.7.3](#)
- [Main differences between V11.7.1 and V11.7.2](#)
- [Main differences between V11.6.2 and V11.7.1](#)
- [Main differences between V11.5 and V11.6.2](#)
- [Main differences between V11.4.1 and V11.5](#)
- [Main differences between V11.4 and V11.4.1](#)
- [Main differences between V11.3 and V11.4](#)
- [Main differences between V11.2 and V11.3](#)
- [Main differences between V11.1 and V11.2](#)
- [Main differences between V11.0 and V11.1](#)

Training

- [Tutorials package for V11.7.x](#)
- [Tutorials package for V11.6](#)
- [Tutorials package for V11.5](#)
- [Tutorials package for V11.4](#)

- [Tutorials package for V11.3](#)
- [Tutorials package for V11.2](#)
- [Tutorials package for V11.0](#)

Links

- [CNES freeware server](#)

Outils

- [Pages liées](#)
- [Suivi des pages liées](#)
- [Pages spéciales](#)
- [Lien permanent](#)
- [Informations sur la page](#)
- [Parcourir les propriétés](#)
- [Citer cette page](#)

- La dernière modification de cette page a été faite le 8 novembre 2019 à 09:38.
- Cette page a été consultée 4 116 fois.

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

