

Basic Principle

De Opera

[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.

OPERA proposed a **Java** interface. To use this **Java** interface, the developer will have to:

- create the **OPERA** [ReentrySimulation](#) object with ...
 - an [OperaConfigurationProperties](#) object created from an existing file
 - a [List<SolarActivityRow>](#) object including actual and predicted solar activities
- call for a given method of the [ReentrySimulation](#) object with a list of **TLE** as arguments

To obtain the results, some "getter" methods will be available depending on the type of computation.

□

Sommaire

- [1 Do not forget to initialize PATRIUS dataset ...](#)
- [2 How to initialize Opera Properties](#)
- [3 How to initialize solar activity](#)
- [4 Reentry simulation object](#)
- [5 Getting list of TLE](#)

Do not forget to initialize PATRIUS dataset ...

As every computation using [[PATRIUS](#)], it is mandatory to well initialize a data set including main information to propagate trajectories (atmospheric models, third bodies ephemeris, **UTC/TAI** shifts, ...). Thanks to the additional jar available [[here](#)] and directly included in the opera-NN.n-jar-with-dependencies.jar file, it is really easy to do it by calling the [OperaReadUtils.iniPatriusDataset\(\)](#) method.

```
// Patrius Dataset initialization (needed for example to get the UTC time)
OperaReadUtils.iniPatriusDataset(null);
```

Rather than to put **null** as input parameter of the [OperaReadUtils.iniPatriusDataset\(\)](#) method, it is possible to give the name of a directory where additional data could be read.

How to initialize Opera Properties

A properties file is mandatory to initialize all the parameters needed to tune an **OPERA** computation. For more explanation about this kind of file, see [here](#).

Thanks to the static method [getConfigurationProperties\(\)](#) from [OperaReadUtils](#) class, we have just to precise the file path:

```
// Opera properties configuration
final OperaConfigurationProperties conf =
OperaReadUtils.getConfigurationProperties("data/opera-
configuration.properties");
```

How to initialize solar activity

Similarly, we can use another static method from the [OperaSolarActivity](#) class. This method ([getSolarActivityFromFile\(\)](#)) needs as input parameters:

- the path for the file where actual activity is stored
- the path for the file where predicted activity is stored
- the CNES Julian date corresponding to the switch between actual and predicted activity

```
// Solar activity initialization
final String realPath = "data/solar/ACSOL_REAL.act";
final String predPath = "data/solar/ACSOL_PREDICTED.act";
final double switchCJD = 22700.0;
final List<OperaSolarActivityRow> solarActivity =
OperaSolarActivity.getSolarActivityFromFile(realPath, predPath, switchCJD);
```

Reentry simulation object

To create such an object only consists in calling the dedicated constructor with both previous [OperaConfigurationProperties](#) and [List<OperaSolarActivityRow>](#) objects:

```
// Reentry simulation creation
final OperaReentrySimulation simulation = new OperaReentrySimulation(conf,
solarActivity);
```

Getting list of TLE

Getting the list of [TLE](#) is done in two steps:

1. Getting all the available [TLE](#) for a given Norad Id calling the static [OperaTleManager.readTLEs\(\)](#) method
2. Extract from this previous list the needed sublist corresponding to a given duration and an "End Of History" date calling the static [OperaTleManager.selectTLEs\(\)](#) method

```
// TLEs initialization
final int noradId = 10479;
final SortedSet<TLE> tlesSet = OperaTleManager.readTLEs("data/tles", noradId,
true, "txt");
// TLEs selection
final double historyDuration = 80.;
final double endOfHistoryCJD = 22605.0;
final List<OperaTLE> tles = OperaTleManager.selectTLEs(tlesSet,
endOfHistoryCJD, historyDuration);
```

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

OPERA

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

GUI Mode

- [Overall presentation](#)
- [Operaapp configuration file](#)
- [Configuration panel](#)
- [Solar activity panel](#)
- [Parameters panel](#)
- [Console panel](#)
- [Result files](#)

- [Plots panel](#)

Batch mode

- [How to call it](#)

Java interface

- [Basic principle](#)
- [Reentry or S/M Estimation](#)
- [Maneuvers Estimation and Pdf report](#)
- [Searching Norad Ids](#)
- [Tutorials](#)

Evolutions

- [Main differences between V7.2.3 and V7.2.4](#)
- [Main differences between V7.2.2 and V7.2.3](#)
- [Main differences between V7.2.1 and V7.2.2](#)
- [Main differences between V7.2 and V7.2.1](#)

Documentation

- [Conference Papers](#)

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 21 janvier 2020 à 08:38.
- Cette page a été consultée 2 266 fois.

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

