

Main differences between V2.4.1 and V2.4.2

The only difference is the correction of an anomaly in the [GPAttitudeSequence](#) class that does not allow to display the available attitude laws.

Main differences between V2.3.3 and V2.4.1

Except the fact that this version uses new [\[GENIUS\]](#) and [\[PATRIUS\]](#) versions, the main evolutions concerns the [GPOrbit](#) widget

New PATRIUS and GENIUS version

This version uses [\[GENIUS\]](#) V1.12.1 and [\[PATRIUS\]](#) V4.10.2. Moreover, the widgets have been validated using [\[PATRIUS_DATASET\]](#) V1.4.3.

New GPOrbit fonctionnalités

- It is now possible to enter an initial orbit using mean parameters. Conversions will also be possible between different mean models and osculating parameters.
- Validity intervals for input parameters are better taken into account.
- Tooltips are available to describe the type of parameters, for example in case of formulae as for the h_x , h_y vector for equinoctial parameters:

The screenshot shows a web form titled "Equinoctial Parameters *". It contains several input fields and a tooltip. The fields are:

- a: ***: Input field with value "7000.0" and unit "km".
- ex: ***: Input field with value "0.0".
- ey: ***: Input field with value "0.0".
- hx: ***: Input field with value "0.0".
- hy: ***: Input field with value "0.0".
- Lo**: Input field with a tooltip that reads "Second component of the inclination vector: $h_y = \tan(i/2) * \sin(\Omega)$ ".
- μ : ***: Input field with value "398600.4415" and unit "km³/s²". A "Select" button is positioned to the left of this field.

Main differences between V2.2.1 and V2.3.3

Summary

- [Dependencies](#)
- [GPOrbit widget improvements and corrections](#)
- [GPFramesConfiguration improvement](#)
- [GPCovariance widget](#)
- [GPForceModels improvement](#)

Dependencies

This version uses [GENIUS](#) V1.11.4 and [PATRIUS](#) V4.8.

GPOrbit widget improvements and corrections

- Previously, when the user modified the type of orbit parameters, if this type owned an anomaly as one of its parameter, this one went back automatically to a true anomaly or the one corresponding to the "**Pivot**". Now, the widget will keep the anomaly originally displayed.
- On angles parameters, warning occurred when values are outside **[-180 deg, +360 deg]**.
- Pattern for date offset entries have been improved.
- Previously, when the user modified "**n**" or "**lon**" values for "**H0-n**" frames definition without selecting the "**Deactivate conversion**" option, the frame conversion was not done.

GPFramesConfiguration improvement

This widget now takes into account the possibility to use an "**EOPs constant**" option to keep the last value of **EOPs** when no ones are available after a given epoch. Moreover, the error messages have been improved, especially when using the **GpOrbitConvertor** tool.

GPCovariance widget

A new widget [GPCorrelation](#) (and some utilities methods) is now available which allows to work with correlation or covariance matrices.

GPForceModels improvement

[GPForceModels](#) has now the possibility to consider complex potential fields for third bodies (by using a new constructor).

Main differences between V2.2 and V2.2.1

The only difference is the correction of the display orbit parameters of the **gporbitconvertor** tool which are now correctly updated when frames configuration are changed.

Main differences between V2.1.1 and V2.2

This version is essentially due to an update of the lower level libraries [GENIUS](#) V1.10.1 and [PATRIUS](#) V4.6.

Summary

- [New reset option in the Copy/Paste menu](#)
- [Adding Copy/Paste/... for GPIntegrator widget](#)
- [GPAttitudeSequence default laws initialization](#)
- [GPOrbit and GPEarthFeatures consistency](#)
- [Anomaly corrections](#)

New reset option in the Copy/Paste menu

Thanks to the new [GENIUS](#) version, there is now a new possibility in the **Copy/Paste/...** menu with the **"Reset"** option that allows to set data to their default values.

Adding Copy/Paste/... for GPIntegrator widget

The menu **Copy/Paste/...** is now available for [GPIntegrator](#) widget.

GPAttitudeSequence default laws initialization

Thanks to the [GENIUS](#) Observer/Observable interfaces, it is now possible to manage consistency between [GPOrbit](#) and [GPFramesConfiguration](#) widgets.

GPOrbit and GPEarthFeatures consistency

It is now possible to better customize the default attitude laws used inside the [GPAttitudeSequence](#) widget.

Anomaly corrections

- There were some problems with the fact that, since V2.0, **GENOPUS** is able to store in a context file, both **"Pivot"** and **"Display"** data for orbital conditions using the [GPOrbit](#) class. Indeed:
 - if **"Pivot"** and **"Display"** frames were not the same, **GENOPUS** considered orbital parameters coming from the **"Pivot"** data and frame characteristics from the **"Display"** ones which was of course totally wrong.
 - if the user modified the context file not through a **GENIUS/GENOPUS GUI**, it had to modify both **"Pivot"** and **"Display"** data in order to keep consistency between both data.
- Always using the [GPOrbit](#) class, when modifying constants (for example μ) of initial orbital conditions (even only the units), a conversion was done which was useless.
- Wrong management of the [GENIUS](#) Observer/Observable interfaces in some widgets.

Main differences between V2.1 and V2.1.1

Summary

- Improvement of interfaces with [GPIntegrator/GPDop853](#) widgets
- Anomaly corrections

Improvement of interfaces with [GPIntegrator/GPDop853](#) widgets

To better use these widgets, some methods have been Added:

- in [GPIntegrator](#) widget :
 - a getter to know which type of integrator has been selected.
 - a getter to know which kind of tolerances has been selected.
 - two getters to directly get the widgets used for min and max step of the [Dormand Price 853](#) integrator (needed to manage externally a "before/after" action on them).
 - a setter allowing displaying or not tolerances on the mass.
- in [GPIntegrator](#) and [GPDop853](#) widgets:
 - validity intervals and methods to modify them. By default, these intervals are:
 - constant time step > 0
 - $0 < \text{min step} \leq \text{max step}$
 - absolute tolerances > 0
 - relative tolerances ≥ 0 .
 - methods to get reduced tolerances (see corresponding anomalies).
- in [GPDop853](#) widget :
 - method to get the widget giving help for tolerances definition (needed to manage externally a "before/after" action on it).

At last, some constructor and methods, will potentially throw a [GIntervalException](#) (and no more a [GPDateException](#) which was not useful).

Anomaly corrections

- Saving displayed orbit as well as the "pivot" reference is now correctly working. Indeed, in the previous V2.1 version:
 - if the user only modified the type of anomaly then saved, this change was not recovered in the saved file. Thus, the next load displayed the original type of anomaly.
 - if the user modified the type of parameters and afterwards changed again the type of anomaly, for display, only the type of parameters was saved and the next load displayed a true anomaly. Note that anyway, even in this new version, when you change the type of parameters, the anomaly is automatically set to "true".
- In the [GPIntegrator](#) widget, updating tolerances was not correctly done for "Reduced" mode.

Main differences between V2.0.1 and V2.1

Summary

- Compatibility with PATRIUS
- New Time scales
- New maneuvers triggering criteriae
- Impulsive maneuvers with orbital parameters criteriae
- Recurrent date events
- New widget for numerical integrators definition
- Additional validity criteria on PRS coefficients
- New autonomous tool to work on orbit data
- Anomaly corrections

Compatibility with PATRIUS

This version is compatible with the version 4.4 of the [PATRIUS](#) library.

New Time scales

If the default mode is the same than for the previous versions, it is now possible to customize the time scales menu of the [GPAbsoluteDate](#) widget and thus to add these additional scales (available inside the [PATRIUS](#) library) as in the example below:

```
ts[0] = GPDateTimeScale.GMST;
ts[1] = GPDateTimeScale.GST;
ts[2] = GPDateTimeScale.TCG;
ts[3] = GPDateTimeScale.UTC;
gpDate = new GPAbsoluteDate("Date:", date0, ts);
```

New maneuvers triggering criteriae

New criteriae on nodes and apsides have been added to the previous ones (relative or absolute dates and AOL).

Impulsive maneuvers with orbital parameters criteriae

It is now possible to define an impulsive maneuver giving orbital parameters increments:

- on semi-major axis
- on eccentricity (and semi-major axis eventually)
- on inclination (and semi-major axis eventually)

Recurrent date events

It is now possible to define date (relative or absolute ones) events with a recurrence (for example each day) using specific widgets as [GPRecurrentDateDetector](#) and [GPRecurrentRelativeDateDetector](#).

New widget for numerical integrators definition

New specific widgets have been created to deal with numerical integrators as some of ones defined in [PATRIUS](#): [Runge Kutta 4](#), [Runge Kutta 6](#) and [Dormand Price 853](#)

Additional validity criteria on PRS coefficients

Besides validity interval of $[0,1]$ for each coefficient, there is an additional test on the sum of the three coefficients (absorption, specular and diffuse) that must be equal to 1.

New autonomous tool to work on orbit data

For people who want to work on orbit characteristics (date, frame, parameters) without coding, it is now possible to use a dedicated autonomous tool with its own **GUI** (gporbitconvertor-x.x-jar-with-dependencies.jar).

Anomaly corrections

- Impossibility to initialize an orbit using a H0-n frame.
- Bad [PATRIUS](#) Orbit building when using the `getPatriusObject()` method: if the displayed frame is different of the "pivot" one, displayed frame is taken into account but orbital elements are coming from the "pivot" one (this bug was also present in previous versions).
- Calling [GPVisibilityFromStationDetector](#) or [GPAttitudeSequence](#) widgets make the tool incompatible with a headless mode (no display).

Main differences between V2.0 and V2.0.1

Summary

- [Event management](#)
- [GPVisibilityFromStationDetector data read](#)
- [GPManeuverScenario and GPManeuverSequence I/F](#)
- [Bad Local time correction](#)

Due to the upgrade of [PSIMU](#) tool (V11.3), some slight modifications have been made necessary. These modifications are mainly internal to **GENOPUS** and deal with:

Event management

“n, m, p” event management, especially with maneuvers and attitude sequences have been modified.

GPVisibilityFromStationDetector data read

Problem linked to the [read\(\)](#) method of the [GPVisibilityFromStationDetector](#) widget: some data was not correctly read.

GPManeuverScenario and GPManeuverSequence I/F

Some interface problems on [GPManeuverSequence](#) and [GPAttitudeSequence](#) classes led to internal modifications.

Bad Local time correction

Bad correction from [radian](#) to [hours](#) of the logger (12 hours offset) in the [CustomLocalTimeDetector](#) class.

Main differences between V1.3.1 and V2.0

Summary

- [PATRIUS](#) and [GENIUS](#) upgrade
- Saving the Orbit data displayed
- TLE entry
- By default values
- Ground Station widget

PATRIUS and GENIUS upgrade

Now, [GENOPUS](#) is linked with a more recent [PATRIUS](#) version (V4.1.1). This update permits to get less "Custom" classes (see [here](#)) as these ones have been deployed directly in [PATRIUS](#). [GENOPUS](#) uses also the last [GENIUS](#) version (V1.8)

Saving the Orbit data displayed

It is now possible to save the way the orbit data have been displayed via the [GPOrbit](#) widget independently of the "Pivot" (i.e. the way the orbit data have been saved in memory).

TLE entry

Always for the [GPOrbit](#) widget, it is now possible to enter directly [TLE](#) data. These data will be automatically converted in osculating parameters.

By default values

In the [GPSunPointingAttitudeLaw](#) widget the by default axis are now [+I](#) and [+J](#) (rather than [+I](#) and [+I](#) in the previous version).

Ground Station widget

A specific [GPGroundStation](#) widget for defining ground station data has been created. This widget is also used in the [GPVisibilityFromStationDetector](#) widget

Main differences between V1.3 and V1.3.1

The only difference with the previous V1.3 version is the fact that it is originally linked to PatriusDataset V1.0.3:

- new **UTC/TAI** table
- new **JPL DE406** ephemeris (by default !)
- new **EOP** data

Main differences between V1.2.1 and V1.3

Some corrections and new features mainly due to the update of the [PATRIUS](#) version (3.3) and use of **GENOPUS** by [PSIMU](#).

Summary

- [Compatibility with PATRIUS V3.3](#)
- [Copy&Paste](#)
- [GPFramesConfiguration](#)
- [GPEventDetectors](#)
- [GPVehicle](#)
- [GPForceModels](#)
- [GPManeuvers](#)
- [GPAttitudeSequence](#)
- [GPGeodeticPoint](#)
- [CustomUS76, CustomMSISE2000](#)
- [Variable names](#)
- [Anomaly corrections](#)

Compatibility with PATRIUS V3.3

This **GENOPUS** version is compatible with **PATRIUS V3.3** (especially with its new management of exceptions due to divisions by zero).

Copy&Paste

Some **Copy&Paste** (and **Import/Export**) functionalities have been added to **GENOPUS** widgets as [GPOrbit](#), [GPVehicle](#), [GPManeuverSequence](#), ...).

GPFramesConfiguration

It is now possible to initialize the [GPFramesConfiguration](#) widget with a specific configuration not necessarily issued from a Factory thanks to [setFramesConfiguration\(\)](#), [setDefaultFramesConfiguration\(\)](#) and [setSavedFramesConfiguration\(\)](#) as well as adapted constructors with a [FramesConfiguration](#) object as input.

GPEventDetectors

- It is now possible to specify the list of available actions associated to the event and, when only one action is indicated, it does not appear on the **GUI**.
- for events which manage multiple occurrences, changes occurred to be more obvious for the user with now three entries:
 - From n^{th} occurrence
 - Every n occurrence
 - Until n^{th} occurrence
- There is no more CodingEvents in the events definition

=> incompatibility in the code due to the possibility to specify the list of available actions and to the occurrence management

GPVehicle

The way to manage mass properties has been improved in order to allow to make mandatory the fact to display dry mass value. Moreover, when this value is equal to zero, it raises a warning and no more an error.

GPForceModels

- It is possible now to hide the choice of the "attraction models" (i.e. **Balmino**, **Cunnigham**, ...)
- as well as to define which models will be displayed.
- Maximum degrees are now displayed when variable potential is selected.
- By default values are proposed for **SRP**.
- Name of the model must be set as a new parameter of the constructor.

=> incompatibility in the code due to additional arguments for the full constructor

GPManeuvers

- If engines and/or tanks needed to define a maneuver is set to **NONE**, the maneuver (and the sequence) raised an error.
- by default, maneuvers components are set to "**Angular**" as event is a "**Relative date**".

GPAttitudeSequence

- It is now possible to define an attitude sequence more simply that using switches, using a single attitude law.
- the constructor has changed as it is also possible to define how the attitude laws list will be displayed (one by one or all in a vertical lists allowed by the [GENIUS GComponentList](#) widget).
- If an attitude law is set to **NONE**, the sequence raised an error.
- only **RESET_STATE** or **RESET_DERIVATIVE** may be selected as actions for switch events.
- by default, attitude laws angular bias definition is set to "**ZYX Cardan Angles**".

=> incompatibility in the code due to the new boolean flag

GPGeodeticPoint

A new argument has been added to the constructor with a boolean that indicates if the name of the geodetic point will be displayed or not.

=> incompatibility in the code due to the new boolean flag

CustomUS76, CustomMSISE2000

The ellipsoid used as an entry is now necessarily a [[PATRIUS](#)] [ExtendedOneAxisellipsoid](#) object and no more a [BodyShape](#) which is one of its interface.

=> may be some incompatibility in the code if the [BodyShape](#) interface was used

Variable names

Some names changed in the XML files in order to be more consistent (for example, capitals vs lowercase letters).

=> some incompatibility with old XML files

Anomaly corrections

- bad management of the clear function of the [GPVehicle](#) when it is not yet initialized
- problem in the [getLaw\(\)](#) method of the [GPSunPointingAttitudeLaw](#) class as the first argument when creating the [CustomSunPointing](#) object was the Sun. It implied that the second direction was normal to the satellite orbit rather than the poles axis as described.
- bad update of the units for the [GPDateOffset](#) widget.
- bug when the initial list of maneuvers was null.
- bug when a rotation was initialized