# Release notes v 3.1.0.

1. **Ion pipe**
   The correction of the ion pipe geometry has been made.

   - The inner area of the ion pipe has its own ID, **SPDPIPE_ISPACE**.
     MC - tracks from the vertex of interaction will receive exactly this label
     (SpdMCtrack::fVModule parameter).
   - Now the vacuum level inside can be specified.

2. **File with parameters**

   - It is possible to fully restore the geometry from the list of parameters, which is usually saved to the output file, **params*.root**. Previously, only the GeoType parameter from the SpdParSet object was used to restore the geometry (now removed from the list of class members, but is still stored in the geometry parameter list).

   - Demo macro: **RestoreInput.C**, function **restore_modules()**.

   - Now using the file with the parameters can be restored:
     - field configuration;
     - list of vertex generators;
     - geometry of setup with all local parameters and settings which were used in the simulation.

3. **Tracker**
   A new auxiliary test module for simulations is added - **Tracker**.

   - It is located in the directory **test/tracker**.

   - In terms of geometry, generally **tracker** is a set of cylindrical layers, "sliced" around the top of the primary interaction. By default the substance in the layers is absent (vacuum), but can be set any.

   - The output file stores information about the particle tracks as an array of objects of the TrackerPoint.

   - The tracker geometry is customizable and can be:

     1) Fully closed type (barrel+endcaps):
        Sets the size of the inner "empty" cylindrical area
        (2 parameters: Rmin -radial size, Zmin - size along the Z axis);
        maximum tracker size
        (2 parameters: Rmax - radial size, Zmax - tracker size along the Z axis);
        N_slices - number of layers (the same for the barrel and endcaps).

In this case, in general, the layer thickness in the barrel and endcaps of the tracker, will be different.

2) endcaps-variant 1:
two cylindrical areas located at equal distance from the center of the setup along the Z axis; layers are "slised" along the Z axis.

3) endcaps-variant 2:
two cylindrical areas located at equal distance from the center of the setup along the Z axis; layers are "slised" radially.

- Geometry of 2 and 3 are also set by 5 parameters:
  - minimum and maximum size along the Z axis (Zmin, Zmax) (position);
  - parameter Rmin - radial bore size inside the cylinder;
  - Rmax - radial maximum area size;
  - N_slices - number of layers.

- Macros for working with the module are in the directory **macro/test/tracker**:
  **ConstructTrackerGeo.C** - tracker geometry;
  **SimuTracker.C** - simulation;
  **DisplayTrackerEvent.C** - view simulated events (particle tracks);
  **CheckTrackerEvents.C** - work with output files (TrackerPoint).

4. **Inner Tracking System**
A new module is added - vertex detector **Inner Tracking System** (SpdIts).
This module is in **its/**.

- The current version of the geometry is a simplified version of the detector compared to the one that is likely to be physically implemented. This variant does not focus on design features. Geometry closer to reality will be created later.

- By default the detector consists of 5 concentric layers. Each layer consists of ladder with silicon chips.
The parameters of the layers are shown in the table below:
  (a) layer number;
  (b) layer distance to Z axis (layer radius), mm;
  (c) size of the ladder along the Z axis, mm;
  (d) number of ladder in layer;
  (e) number of chips on lader.

| a | b | c | d | e |
|---|---|---|---|---|
| 1 | 70.0 | 624.0 | 8 | 9 |
| 2 | 136.0 | 1120.0 | 15 | 17 |
| 3 | 182.0 | 1369.0 | 20 | 21 |
| 4 | 212.0 | 1621.0 | 22 | 25 |
| 5 | 241.0 | 1873.0 | 26 | 29 |

- All ladders are rotated around the axis by 10 degrees, which coincides in direction with the axis Z. Size of active area of one chip is 60.8×60.8 mm. Sensor thickness is 300 $\mu$m. Each chip is divided into 640×320 channels (strips). There is a gap between the chips along the Z axis equal to 2.4 mm (its size can be changed or set to zero).

- Geometry of the vertex Detector is customizable (each layer is configured separately). The number of layers, ladders, chips and channels can be reduced or increased. All the parameters can be changed using the class **SpdItsGeoMapperX** (it is located in the directory spdgeometry/). The position (coordinates) and orientation of the ladder, chips and channels can be obtained using the class **SpdItsGeoBuilder** (it is located in the directory spdgeometry/).

- Geometry can be drawn or customized using a demo macro **geom/ ConstructItsGeo.C**. Also the demo macro can show an example of working with the SpdItsGeoBuilder class.

- Comments to the code in the file **SpdItsGeoMapperX.h** contains the information about how to configure the geometry.

- The data is saved to the output file as arrays of objects of **SpdItsPoint**. Each SpdItsPoint object contains the following information about the particle track:
  - ID of the chip, through which the particle passed;
  - the parameters of the particle itself (momentum and coordinate) at the input and output points of the chip;
  - the moment (time) of entry of the particle into the chip and the length of the track at the moment of entry;
  - total energy release in the chip.

- **SpdItsPoint** contains two arrays of numbers - a list of channels through which the particle passed and the array of energy release corresponding to these channels.

- Actually SpdItsPoint object is a description of a segment of a particle track inside the chip. In terms of space in the output file, this data format is economical, as well as convenient for processing.

- An example of a macro for working with Its data:
  **macro/analysis/its/CheckItsData.C**
  How the macro works:
  a) opens a root files with parameters and data;
  b) completely restores the configuration of the vertex detector that was used in the simulation from the file with parameters;
  c) sequentially reads events from the file;
  d) for each event displays the contents of the SpdItsPoint objects, chip parameters (name, geopath, number, coordinates), a list of all channels with their parameters (strip coordinates and numbers), energy release in individual channels and total in the chip.

5. **Change the type of geometry**
   Change the default geometry type (some parameters of modules or set), which is used for simulation (macro **SimuHyb.C**).

   - The type of geometry is determined by a method SpdCommonGeoMapper::DefineHybGeometrySet().

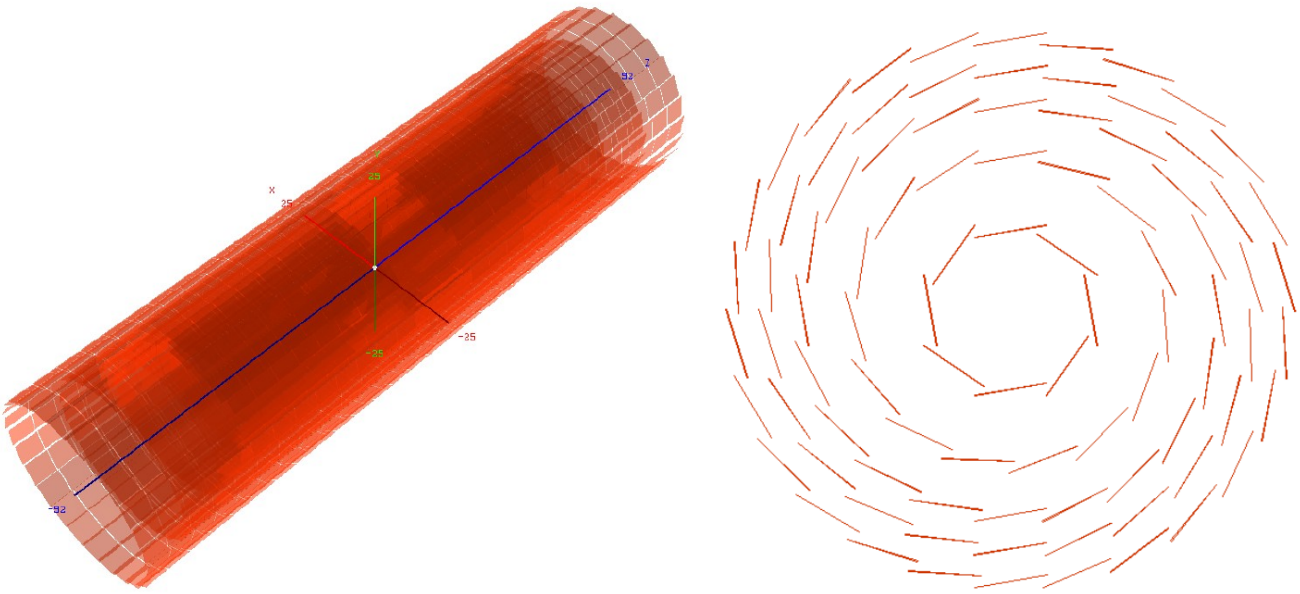

Figure 1: Inner Tracking System

- In the current version of the setup configuration in the **TsTB** module (barrel of the tracking system) left only the central part, and two laterals will be the part of **TsTEC** (tracker endcaps).

- At the moment, geomery of **TsTEC** is not yet written. So, temporarily used endcaps-option 1 of the test module Tracker: two cylinders (with a bore for the ion pipe) are located on laterals of the central part of the tracker, inside the hybrid magnet. Each cylinder consists of 200 layers 1 cm thick. By default, the substance of the layers is not set (vacuum). Particle passage data through the layers of endcaps are also stored in the output file as an array of TrackerPoint objects.

- The current geometry of the setup can be viewed by running the macro **geom/ConstructHybGeo.C**.

6. **Field map**

- By default field map is now called **map_hyb_1T5cm.txt**.

- Also, another one **map_hyb_2T5cm.txt** has been added to the input/ directory. Using it, the current in the coils was increased by 2 times.