**Track3P Top-Level Commands**

- **TotalTime**
- **ParticlesTrajectories**
- **FieldScales**
- **NormalizedField**
- **Emitter**
- **Domain**
- **Material**
- **OutputImpacts**
- **SingleParticleTrajectory**
- **Postprocess**

Note: Refer to **acdtool** command syntax for postprocessing capabilities.

## TotalTime

TotalTime: 50

**TotalTime** sets the total simulation time in rf cycles. Default is 20 rf cycles.

## ParticlesTrajectories

```
ParticlesTrajectories:
{
   ParticleFile: partpath
  Start: 0
  Stop: 1000000
  Skip: 1
}
```

**ParticlesTrajectories** specifies the particle parameters for particle output iles, which can be used to visualize particles' trajectories. It is only used for **ScanToken** (in **FieldScales**) is 0.

**ParticleFile** defines the particle file names. Default is **partpathXXXX**.

**Start**: Start time step for writing particle files. Default is **0**.

**Stop**: End time step for writing particle files. Default is **1000000**.

**Skip**: Skip time step for writing particle files. Default is **1** for no skip.

## FieldScales

```
FieldScales:
{
  Type: FieldGradient
 ScanToken: 1
 Minimum: 23.0e+6
 Maximum: 25.0e+6
 Interval: 1.0e+6
 Scale: 23.00e+6
}
```

**FieldScales** specifies the parameters for the field levels where track3p simulations are carried out. The field level is represented in different types, and it can be single level or scanned.

**Type** defines the physical quantity that is used to define the field level. There are three types

- **FieldGradient**: The average field gradient along a specified path defined by the container **NormalizedField** [unit in V/m]
- **InputPortPower**: The power at the input port [unit in W]
- **StorEnergy**: The stored energy of the rf field [unit in J]

**ScanToken**: **0** or **1**
**0**: A single field level set by **Scale**; files on particle tracking will be written.
**1**: Scan field level from **Minimum** to **Maximum** in intervals defined by **Interval***;* no particle tracking files will be written.

**Minimum**: The minimum scanned field level [unit defined by **Type**]. For example, if **FieldGradient** is used, the unit is V/m. It is used when **ScanToken** is set to **1**.

**Maximum**: The maximum scanned field level [unit defined by **Type**]. For example, if **FieldGradient** is used, the unit is V/m. It is used when **ScanToken** is set to **1**.

**Interval**: The interval for scanned field levels [unit defined by **Type**]. For example, if **FieldGradient** is used, the unit is V/m. It is used when **ScanToken** is set to **1**.

**Scale:** The single field level [unit defined by **Type**]. For example, if **FieldGradient** is used, the unit is V/m. It is used when **ScanToken** is set to **0**.

## NormalizedField

```
NormalizedField:
{
 Beta: 1
 StartPoint: 0.0 0.0 -0.057
 EndPoint: 0.0 0.0 0.057
 N: 300
}
```

**NormalizedField** specifies the coordinates of the start and end points of the straight path that is used to normalize the average field gradient. It is only used when **Type** in the container **FieldScales** is **FieldGradient**.

**Beta:** Beta parameter for field calculation, default = 1.0

**StartPoint**: x, y, z
The coordinates (x, y, z) of the start point of the path [units in m]

**EndPoint**: x, y, z
The coordinates (x, y, z) of the end point of the path [units in m]

**N:** Number of inner points in the gradient integral line. Default is 300.

## Emitter

```
Emitter:
{
 Type: 2
 t0: 0
 t1: 1
 x0: -0.03
 x1: 0.035
 y0: 0.04
 y1: 0.1
 z0: -0.045
 z1: 0.0
 BoundaryID: 6
 SkipTimeSteps: 1
 N: 1000
 WorkFunction: 4.2
 Beta: 120
}
```

**Emitter** defines the locations and duration for the emission of initial particles (primary particles). Multiple **Emitter** can be defined on a single mesh surface.

**Type**: Defines how the particles are emitted and can be

- **ElectronicSurface** or **2**: Initial particles are emitted from the mesh surface inside a bounding box. This is the default option.
- **SolidElectronicSurface** or **5:** Initial particles are emitted from the mesh surfaces as well as the boundaries of different volumetric regions. This is used for multipacting simulation of rf windows which have different material regions.
- **FieldEmission** or **7:** Primary particles are emitted following Fowler Nordheim Law. This is used for dark current simulation.
- **ParticlesFile or 10:** Load particles from a file for particle tracking.
    - **ParticlesFile:** particles.dat
    - Format of particle data:

num. electrons/ energy(ev) / position_x(m) / y(m) / z(m) / momentum_x / y / z

**t0** and **t1**: Describe the start and end times in rf cycles when particles are emitted. Default it ranges from 0 to 1, but other values can be used.

**x0, x1, y0, y1, z0** and **z1**: The bounding box for particle emission from the surface or volumetric boundaries [units in m]

**BoundaryID:** The boundary surface ID for particle emission

**SkipTimeSteps:** Emission skip time step. Default is **1** for no skip.

**N:** Minimum number of electrons in an emitted macro particle. Particles will not be emitted if the number of electrons is less than this value. Only used for field emission

**WorkFunction:** Work function for field emission [unit in eV]. Default is 4.2 eV for copper. Only used for field emission

**Beta:** Beta value for field emission. Only used for field emission

**Domain**

```
Domain:
{
 FieldDir: ./omega3p_results
 ModeID: 0
 FrequencyScanID: 0 // for S3P only
 dt: 1
 MaxImpacts: 50
 LowEnergy: 20
 HighEnergy: 5000
 InitialEnergy: 2.0
 Voltage: 0.0 0.0 0.0
 Bins: 64

// For rf windows
 SolidRegion: 2
 VacuumRegion: 1

// With uniform external magnetic field
 ExternalMagneticField: 0.0 0.0 4.0

// With external magnetic field map
 MagneticFieldMap:
 {
    File: bfield2
    Scaling: 0.0001
    Units: 0.001
    ZOffset:0.27
 }

// With electrostatic field
 ElectroStaticFieldMap: {
  File: ./gun3p_results/OUTPUT/E/test-11.txt
  Scaling: 1.0
 }

 SymmetryFactor: 2
 SymmetryPlane: 5

}
```

**Domain** specifies field parameters for various types of simulation condition. This includes simulations with rf fields, external static field, or with rf windows.

**FieldDir**: The directory storing the omega3p or s3p results. Track3P will read in a mode file from the directory.

**ModeID**: Specifies the mode number calculated using omega3p or port mode ID for s3p imported into track3p. The default is the first eigenmode for Omega3P and the first port mode ID for S3P, starting with the number 0.
**FrequencyScanID**: Eigensystem ID from frequency scan in S3P

**dt**: Parameter used for time step. Default dt = 1.0, timestep = dt/frequency/100, and default timestep = 1.0/frequency/100.

**MaxImpacts**: Maximum impact number, after which particles will die. Default is 50.

**LowEnergy** and **HighEnergy**: Define the impact energy range, beyond which no secondary particles are emitted when a primary particle hit a surface. The energy units are in eV.

**InitialEnergy:** Initial energy of emitted particels [unit in eV]. Default is 2.0 eV.

**Bins:** Parameter used for localizing particles, with the maximum number of bins set in the longest direction among those in x, y, z and the other remaining two directions' bins derived from it. Default is 64.

**Voltage:** External voltage for coax, with input format (a ra rb), where the units in a, ra, rb are [V, m, m] and voltage = a/log(rb/ra). Default in (0 0 0) for no external voltage.

**SolidRegion**: Non-vacuum region, whose block ID is set in Cubit. It is only used for simulation models with rf windows.

**VacuumRegion**: Vacuum region, whose block ID is set in Cubit. It is only used for simulation models with rf windows.

**ExternalMagneticField**: Bx, By, Bz
Uniform external magnetic field (Bx, By, Bz) [units in T]

**MagneticFieldMap**: External magnetic field map to be imported

- **File**: The name of the file containing the external magnetic field map

Track3p read field maps with cylindrical symmetry. Data iare read in from an ASCII "Poisson" file containing (Fr,Fz,Fphi) as a function of z and r positions. The field map is formatted either as follows with for 3 field components:

```
r_0 r_{Nr+1} Nr              // from .. to .. number of intervals in r direction
z_0 z_{Nz+1} Nz              // from .. to .. number of intervals in z direction
Fr Fz Fphi (at z_0, r_0)   // field values at the corresponding points, there need to be (Nr+1)
* (Nz+1) lines starting from here
Fr Fz Fphi (at z_0, r_1)
...
Fr Fz Fphi (at z_0, r_{Nr+1})
Fr Fz Fphi (at z_1, r_0)
...
Fr Fz Fphi (at z_1, r_{Nr+1})
…
Fr Fz Fphi (at z_Nz, r_{Nr+1})
```

or, as follows for 2 field components:

```
r_0 r_{Nr+1} Nr
z_0 z_{Nz+1} Nz
Fr Fz (at z_0, r_0)
Fr Fz (at z_0, r_1)
...
Fr Fz (at z_0, r_{Nr+1})
Fr Fz (at z_1, r_0)
...
Fr Fz (at z_1, r_{Nr+1})
...
Fr Fz (at z_Nz, r_{Nr+1})
```

- **Scaling**: The field scale factor to multiply the fields in the original field map
- **Units**: The distance scale factor to convert original coordinates in the original field map to meters used in track3p
- **ZOffset**: The z-offset of the origin [unit in m] when importing the original field map into track3p simulation model

**ElectroStaticFieldMap:** External electrostatic field map to be imported

- **File**: The name of the file containing the external electrostatic field map
- **Scaling**: The field scale factor to multiply the fields in the original field map

**SymmetryFactor: symmetry plane factor. 2: half; 4: quarter; 1: full**
**SymmetryPlane: symmetry plane boundary surface ID**

## Material

```
Material:{
 Type: Primary
 BoundarySurfaceID:  6
}
Material:{
 Type: Secondary
 BoundarySurfaceID: 3 4 6
}
Material:{
 Type: Absorber
 BoundarySurfaceID:  8 7
}

// For rf windows:
Material:{
 Type: Primary
 SolidBlockID:  2
}
Material:{
 Type: Secondary
 SolidBlockID:  1
}
Material:{
 Type: Absorber
 BoundarySurfaceID:  7 8
}

Material:{
 Type: Secondary
 BoundarySurfaceID: 6
 Model: 2
 SecondaryEmissionYield: sey.dat
 MinimumNumElectrons: 1000
 ElasticThreshold: 1000
}
```

**Material** specifies the characteristics of particles on a surface.

**Type**: Type of surface

- **Primary**: Particles are emitted.
- **Secondary**: Secondary particles are emitted when a particle hit the surface.
- **Absorber:** The particle is absorbed and lost when it hits the surface.

**BoundarySurfaceID:** Reference number of boundary surface, which is set in Cubit.

**SolidBlockID:** Reference number of the block ID for solid material, which is set in Cubit, It is only used for rf windows.

**Model:** defines different types of secondary emission. Default is **1.**

- **1:** For multipacting simulation. When a particle hits a surface, one secondary particle will be emitted from the same location with angle normal to the surface.
- **2:** For dark current simulation. When a particle hits a surface, true secondary and reflected particles will be emitted with random angles.
- **3:** Furman-Pivi secondary emission model. When a particle hits a surface, multiple secondary particles will be emitted following Furman-Pivi model.

**MinimumNumElectrons:** Minimum number of electrons in each secondary macro particle. If the number of electrons is less than this number, no secondary particle is emitted. This parameter only applies to **Model 2.**

**ElasticThreshold:** Minimum number of electrons in each secondary reflected-type macro particle. If the number of electrons is less than this number, no secondary reflected-type particle is emitted. This parameter only applies to **Model 2.**

**SecondaryEmissionYield:** Secondary emission yield data file formatted as:

Column 1: impact energy specified in **eV**.
Column 2:corresponding secondary emission yield coefficient

## OutputImpacts

OutputImpacts: off

**OutputImpacts** specifies if particle impact information will be written.
**on**: Particle impact information will be dumped into files for detailed studies.
**off**: No particle impact information will be saved.

## SingleParticleTrajectory

SingleParticleTrajectory: off

**SingleParticleTrajectory** specifies if all single particle trajectory will be written in separated files. Default is **off**.
**on**: Each single particle trajectory will be written in a file named “particleID”.
**off**: No single particle trajectory will be dumped.

## Postprocess

```
Postprocess: {
 Toggle: on

 ResonantParticles: {
        Token: on
        InitialImpacts      :   4
        EnergyRange         : 25 10000
        PhaseTolerance      :   0.01
    }

 EnhancementCounter: {
        Token: on
        SEYFileName1: copper.txt
        BoundarySurfaceID1: 6
        SEYFileName2: niobium.txt
        BoundarySurfaceID2: 3
         SolidVolID: 0
         MinimumEC: 0.01

     }
}
```

**Postprocess** specifies the option for postprocessing of data from multipacting simulations.

**Toggle**: **on** or **off**
**on**: Turn on the postprocessing procedure.
**off**: Turn off the postprocessing procedure.

**ResonantParticles**: Define the parameters for tracking resonant particles.

- **Token**: **on** or **off**
  - **on**: Turn on the tracking of resonant particles.
  - **off**: Turn off the tracking of resonant particles.
- **InitialImpacts:** Particles with impact number greater than the number of initial impacts are considered. Default is 4.
- **EnergyRange**: E1, E2
  The energy range between E1 and E2 to consider if a particle exhibits a resonant trajectory [units in eV]. When the impact energy of the particle is out of this range, it will be discarded from the list of resonant particles.
- **PhaseTolerance:** Phase tolerance to determine resonant particles

**EnhancementCounter:** Specifies the parameters used to calculate the particle enhancement counter. The enhancement counter is defined as the total number of secondary electrons originated form a single primary electron during the number of rf cycles in a simulation. It is written as

$$ec = \delta_1 * \delta_2 * \ldots * \delta_m$$

where $\delta_i$ is the secondary emission yield (SEY) at the *i*th impact and its value is determined by the impact energy and the SEY curve. The parameters for setting the enhancement counter are

- **Token**: **on** or **off**
  **on**: Turn on the enhancement counter.
  **off**: Turn off the enhancement counter.
- **BoundarySurfaceID1**: The boundary surface ID, which is set in Cubit, used for calculating the enhancement counter
- **SEYFileName1:** The name of the file containing the SEY for the corresponding material on **BoundarySurfaceID1**. The first and second columns in the file are the impact energy and the number of emitted particles due to a single particle impact, respectively.
- **BoundarySurfaceID2**: The boundary surface ID, which is set in Cubit, used for calculating the enhancement counter
- **SEYFileName2:** The name of the file containing the SEY for the corresponding material on **BoundarySurfaceID2**. The first and second columns in the file are the impact energy and the number of emitted particles due to a single particle impact, respectively.
- **SolidVolID:** Window volume ID, which is set in Cubit
- **MinimumEC:** Minimum enhancement counter, less than which no data will not be written to file. Default is 10.