

# astroid Dosimetry App User Guide

## Overview

The .decimal astroid Dosimetry App device is used to aid planning and analysis of proton radiation therapy treatments. The astroid Dosimetry App device is not an interactive end user application and is not intended to perform all the functions of a typical radiation treatment planning system. Users of the system will write scripts or use a fully interactive software program that makes calls to the functions provided by the astroid Dosimetry App. In essence, this device serves as a foundational proton dosimetry calculation library that greatly reduces the burden and time required to develop treatment planning and plan analysis software by making readily available much of the core functionality common to these types of applications. This core functionality includes various CT image processing tools, structure and contour modification operators, proton dose calculations, proton aperture and range compensator device design algorithms, and many other radiotherapy specific calculation functions. Since this application does not include a graphical user interface, much of the usability and plan approval tools are not applicable or available herein, and it is expected that derived treatment planning applications follow the guidelines set forth in IEC 62803 and IEC 62366 to make sure all such features are included in final end user applications.

Access to the astroid Dosimetry App is provided by the thinknode™ framework using http json formatted requests. thinknode™ provides the 'backbone' used to send and receive requests, maintain users, realms and organizations, and provide data storage and management.

## Purpose

The purpose of this document is to provide guidance on the setup, access, and usage of the astroid dosimetry app.

## Getting Started

The primary purpose of the astroid Dosimetry App is to provide a library of efficient and easily accessible proton dosimetry functions which can serve as the foundational tools for developing higher level radiation treatment planning and analysis software programs. The Dosimetry App is not intended to be an interactive end-user application; instead it will serve as a source of tested and validated calculation functions which software developers can utilize in other end-user applications. The astroid Dosimetry App provides access to these calculation functions through the thinknode™ framework which is accessible over the internet and performs all calculations in a cloud based environment. thinknode provides robust, redundant data storage, ensuring industry leading data integrity as well as stable software version management and release process.

## Connecting to thinknode™

The Dosimetry App, and all its released versions live on thinknode™ hosted on the Amazon S3 service. Communication with the App is performed over the internet through thinknode https requests which are password protected per user and permission based by thinknode organization administrators. The thinknode startup procedure includes full installation of the appropriate application environment onto the virtual servers, which includes hardware and performance checks to ensure sufficient resources are available for the requested application. A full thinknode™ userguide is found at <https://docs.thinknode.com>.

Basic information needed to communicate with the Dosimetry App through thinknode™:

- thinknode™ user name and password as a base64 encoded blob (e.g.: userName:PassW0rd1 → dXNIck5hbWU6UGFzc1cwcmQx)
- thinknode™ api url (e.g.: <https://api.thinknode.com>)
- thinknode™ version the application is released on (e.g.: v1.0)
- application name (e.g.: Dosimetry)

See [thinknode™ Examples](#) for sample code projects for connecting to the Dosimetry App.

## Instructions for Use

The [Instructions for Use](#) outlines the intended use of the Dosimetry App. Refer to each section below for more detail:

- [Overview](#)
- [Warning](#)
- [User Profile](#)
- [Product Features](#)
  1. [Function Access and Data Storage](#)
  2. [Proton Dose Calculations](#)
  3. [Proton Aperture Designs](#)
  4. [Proton Range Compensator Designs](#)
  5. [Patient Specific Data Model](#)
  6. [Proton Treatment Delivery Machine Model](#)

## Precautions

It is the responsibility of those utilizing this application to ensure all that all usages of this product relating to patient treatments are performed by trained and qualified personnel only and that such personnel is aware that the quality of any generated treatment plans is highly dependent on the quality and correctness of the input data; therefore if any questions or uncertainties exist regarding the quality, units, or identification of input data arise, they must be investigated and resolved before the data are used.

# Proton Delivery System Conventions

(this section could also be named Geometric and Beamline Conventions)

The [Proton Delivery System Conventions](#) page details definitions about the following topics:

- **Proton Delivery System Conventions:**
  - [Coordinate Systems](#)
  - [Beam Representation](#)

## Function Categories

The Dosimetry App exposes three primary classes of functions through its API: dose calculation functions (DCF), design task functions (DTF), and radiotherapy support functions (RSF). Each function has its own type of input argument specifications and output results specifications. It is worthy of pointing out the interdependence of these function classes upon one another. Specifically, the DCFs rely heavily on the use of the RSFs and may take as input, the output devices produced by the DTFs. The DTF are typically higher level functions that will modify data parameters and make repeated use of the various DCFs and RSFs to accomplish some type of “optimal” device or plan design.

- Details of each function, argument parameters, and return values are provided at the [Dosimetry Manifest Guide](#).

## Design Task Functions

Detailed function and data type definitions can be found at [Dosimetry Manifest Guide](#). Below are basic examples of using design task functions for aperture and range compensator designs.

- **Aperture Design:** Explanation and sample process for aperture geometric design functions
  - [Compute Aperture](#)
  - [Point vs Dual Source](#)
  - [Aperture Mill Radius and Smoothing](#)
  - [Aperture Shape Manipulation](#)
- **Range Compensator Design :** Explanation and sample process for range compensator degrader functions
  - [Compute an Optimized Range Compensator](#)
  - [Patch Field Range Compensator](#)

## Dose Calculation Functions

Detailed function and data type definitions can be found at [Dosimetry Manifest Guide](#). Below are basic descriptions and details about the Dose Calculation Functions, dose modeling, and dose construction.

- **SOBP Dose Functions**
  - [Dose Calculation Modeling](#)
  - [Construction and Modeling](#)

## Radiotherapy Support Functions

Detailed function and data type definitions can be found at [Dosimetry Manifest Guide](#). Below are basic descriptions and details about common Radiotherapy Support Functions.

- **Radiotherapy Support Functions**
  - [Image Processing](#)
  - [Contour and Structure Modification](#)
  - [Degrader Manipulation](#)

## Data Types

Details of each datatype used by the Dosimetry App can be found at the [Dosimetry Manifest Guide](#).

Some more complex data types have detailed descriptions and example usages shown below:

- [Dij Matrix](#)

## Examples

### thinknode™

Official thinknode™ user guide documentation can be found at <https://docs.thinknode.com>. Below are basic examples of communicating with thinknode™ calculation providers.

- **C++:** Sample C++ project (note: this example requires third party libraries) that details the following thinknode™ api functions:
  - [C++: Post to Immutable Storage](#)
  - [C++: Post a Calculation Request](#)
- **Python:** Sample Python project (note: this example requires third party libraries) that details the following thinknode™ api functions:
  - [Python: Post to Immutable Storage](#)
  - [Python: Post a Calculation Request](#)
- **Node.js:** Sample node.js scripts
  - [Node: Decrypt Base 64 Blob Data](#)

## DTFs, DCFs, & RSFs

Refer to the [Function Categories](#) section for explanation and usages for each function type.

## System Usage

Normal system usage requires following the specifications for functions and data types as detailed by the [Dosimetry App Manifest Guide](#). Making correct calls to the calculation provider requires following the specifications for api requests as detailed by the [thinknode™ API guide](#). Additionally, as most large data structures are returned in base64 encoded format, users can and should perform error checking during decoding to be certain data was not lost or corrupted during transmission.

Details about the intended use of the function categories can be found at [Function Categories](#) section.

## Improper System Usage

When using the astroid Dosimetry App, as with any complex program, there is the potential for misuse. The thinknode™ framework and Dosimetry App limit most of the potential misuse of exposed functions and data types, but there's still a potential for misinterpretation of the results provided by the app. Below are a few examples of topics that are easily confused or overlooked when performing even basic tasks.

### Grid and Image Data Interpretation

The difference between the grid and image origin definitions can easily result in image or structure data being misinterpreted. Grid origins are defined at the top left most corner of the grid, and image origins (where the pixel data is) is defined at the center as shown below.



When looping through image data on a regular grid, it's important to remember that the location of the pixel value of the image doesn't match the corresponding grid location exactly. Forgetting this can result in image data being interpreted 1/2 a slice off.

### Interpreting Point Data

When reading point data from an array, the data is formatted as [ X, Y, Z, X, Y, Z, X, Y, ... Z ].

### Coordinate Systems

It's important to remember there are multiple coordinate systems being used simultaneously for

dosimetry planning that are all very dependent on each other.

For a more detailed look at the coordinate systems refer to [Proton Delivery System Coordinate Systems](#)

## Known Application Limitations

Below are listed the known application limitations, defects, or inconsistencies.

### 1. Dose Algorithms (Dose Calculation Functions [DCF])

- The DCFs follow the approach of Hong et al (1996), therefore the DCFs suffer from similar known limitations as described by therein “The algorithms do not model the ‘slit scattering’ from the aperture and collimator. This effect could be substantial at the surface of the patient in some cases (Mayo et al 1993, Gottschalk et al 1994). The pencil beam algorithm also produces errors in the shadow of thick inhomogeneities whose edge is parallel to the beam’s central axis, just as do pencil beam algorithms for electron beams (Hogstrom et al 1981).” For further details and examples of these limitations, please refer to the work of Hong et al (<http://iopscience.iop.org/0031-9155/41/8/005>).

### 2. `compute_sobp_pd_dose` is the preferred sobp dose calculation function.

1. Using Dij sparse matrix causes application memory allocation issues for medium and large case scenarios.

### 3. Large array\_types of number\_types create issues with uploading to the ISS provider if the points are too dense.

1. Consider keeping arrays of numbers from being overly large.

Should you discover or suspect any additional errors or limitations not noted above, please contact the .decimal astroid Dosimetry App development lead, Kevin, Erhart, at [kerhart@dotdecimal.com](mailto:kerhart@dotdecimal.com) with details of your concerns. All concerns will be addressed as quickly as possible and follow up with the submitter will occur throughout the inquire and correction process.

From:

<http://apps.dotdecimal.com/> - **decimal App Documentation**

Permanent link:

<http://apps.dotdecimal.com/doku.php?id=userguide:userguide&rev=1424272230>

Last update: **2021/07/29 18:19**

