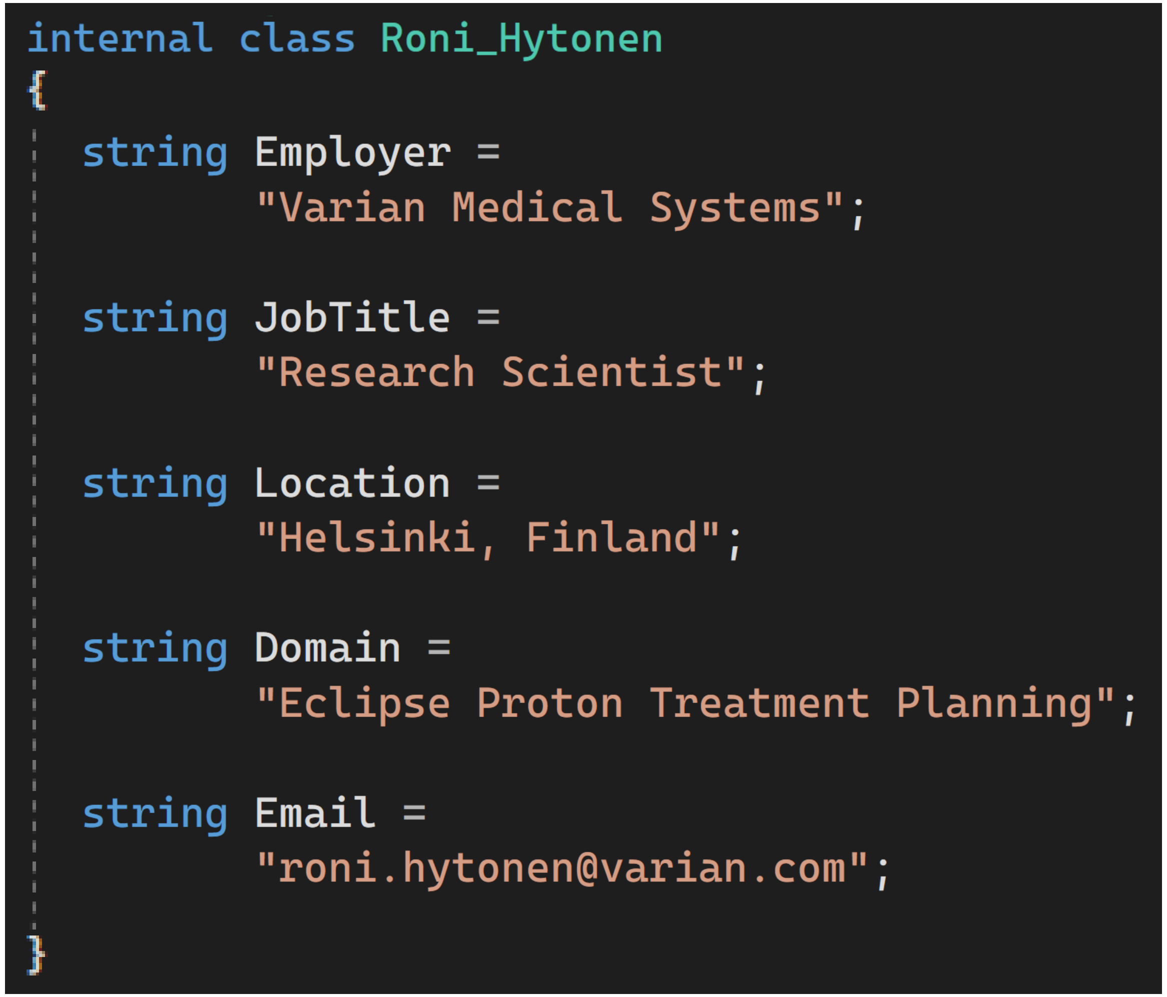
ESAPI for Proton Planning

Features in Eclipse 16–18.0

2023 AAPM Annual Meeting
Varian Developer Symposium

Roni Hytonen
Research Scientist, Varian Proton Planning



```
internal class Roni_Hytonen
{
    string Employer =
           "Varian Medical Systems";

    string JobTitle =
           "Research Scientist";

    string Location =
           "Helsinki, Finland";

    string Domain =
           "Eclipse Proton Treatment Planning";

    string Email =
           "roni.hytonen@varian.com";
}
```

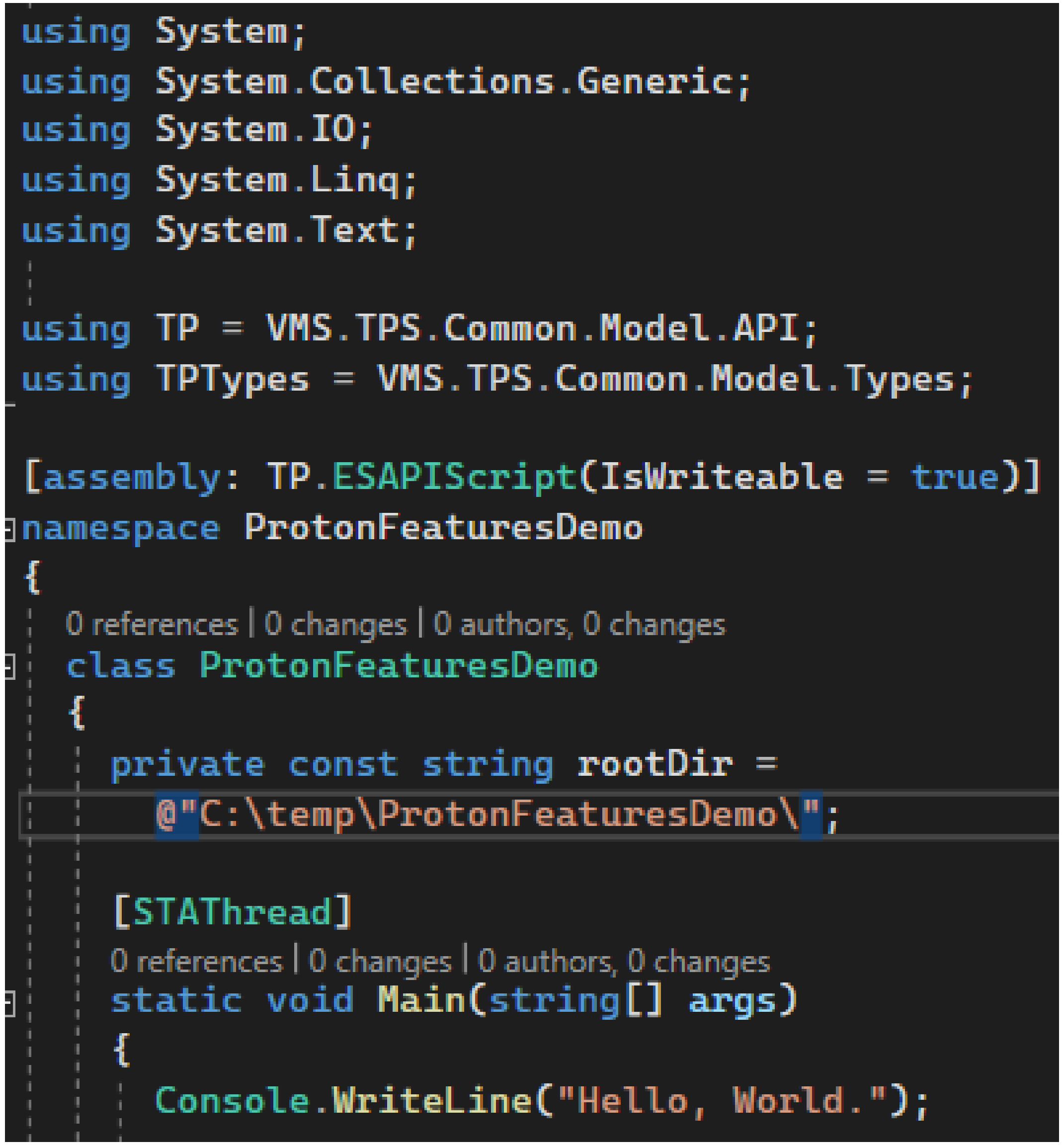
Agenda

ESAPI Features for Protons

With Demos & Examples

Current Limitations

Significant overlap with photon side!

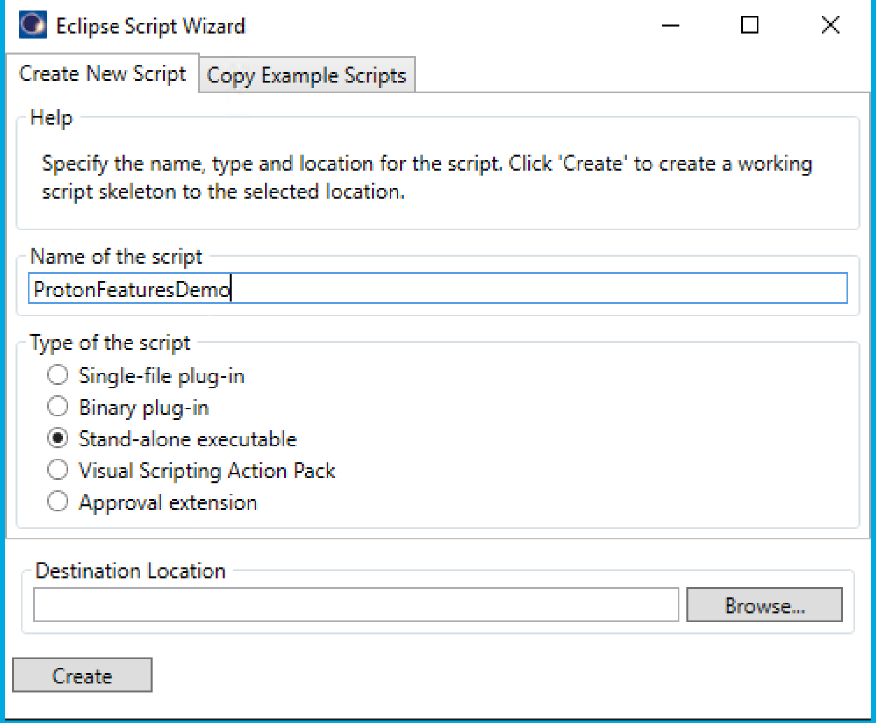


```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;

using TP = VMS.TPS.Common.Model.API;
using TPTypes = VMS.TPS.Common.Model.Types;

[assembly: TP.ESAPIScript(IsWriteable = true)]
namespace ProtonFeaturesDemo
{
  0 references | 0 changes | 0 authors, 0 changes
  class ProtonFeaturesDemo
  {
    private const string rootDir =
      @"C:\temp\ProtonFeaturesDemo\";

    [STAThread]
    0 references | 0 changes | 0 authors, 0 changes
    static void Main(string[] args)
    {
      Console.WriteLine("Hello, World.");
```

```
…\Debug > .\ProtonFeaturesDemo.exe
Hello, World.
```

Getting Started

- Create a new .exe ESAPI project with the Eclipse Script Wizard.
- With no ESAPI calls, build and run.
- If all good, we can proceed.



Eclipse Script Wizard

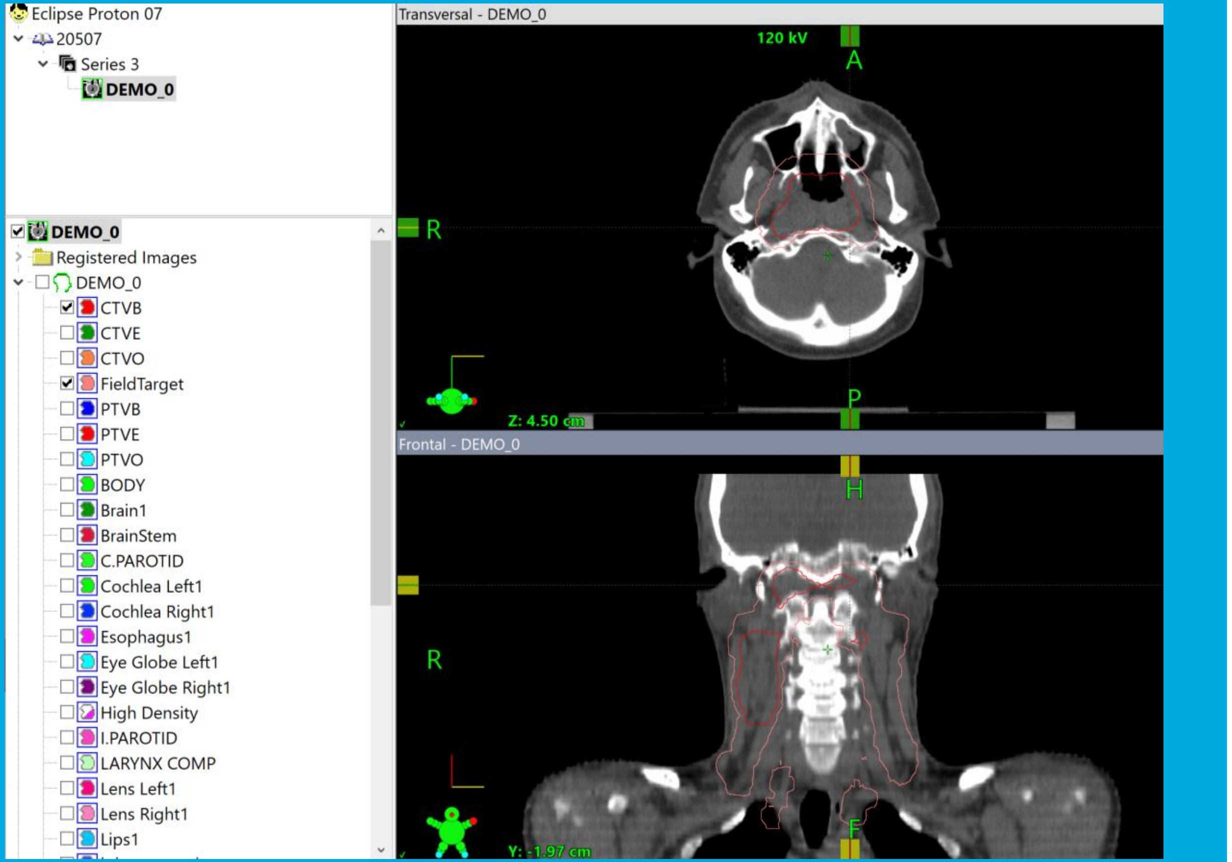
Help

Specify the name, type and location for the script. Click 'Create' to create a working script skeleton to the selected location.

Name of the script: ProtonFeaturesDemo

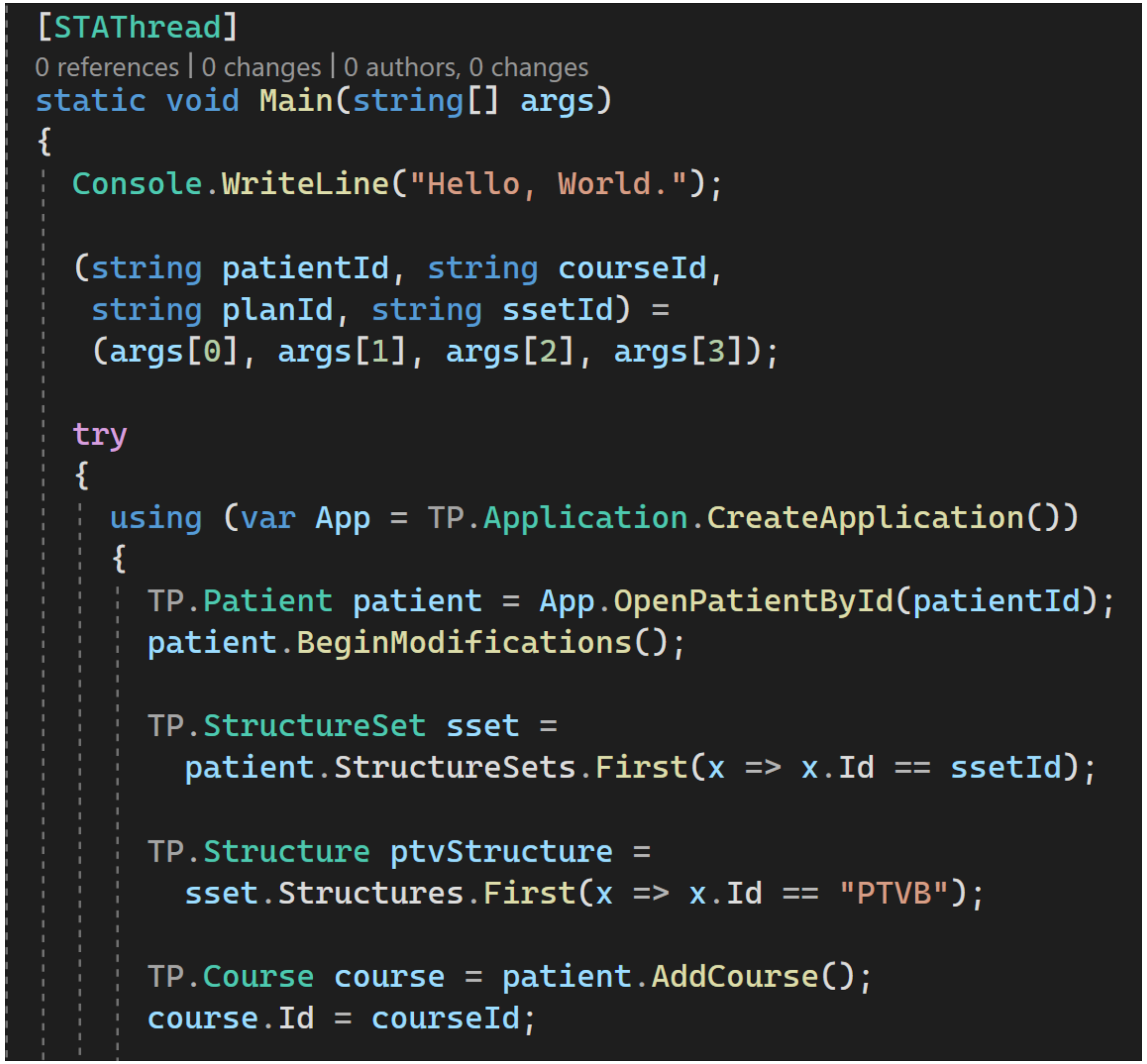
Type of the script
○ Single-file plug-in
○ Binary plug-in
● Stand-alone executable
○ Visual Scripting Action Pack
○ Approval extension

Destination Location



Sample Data

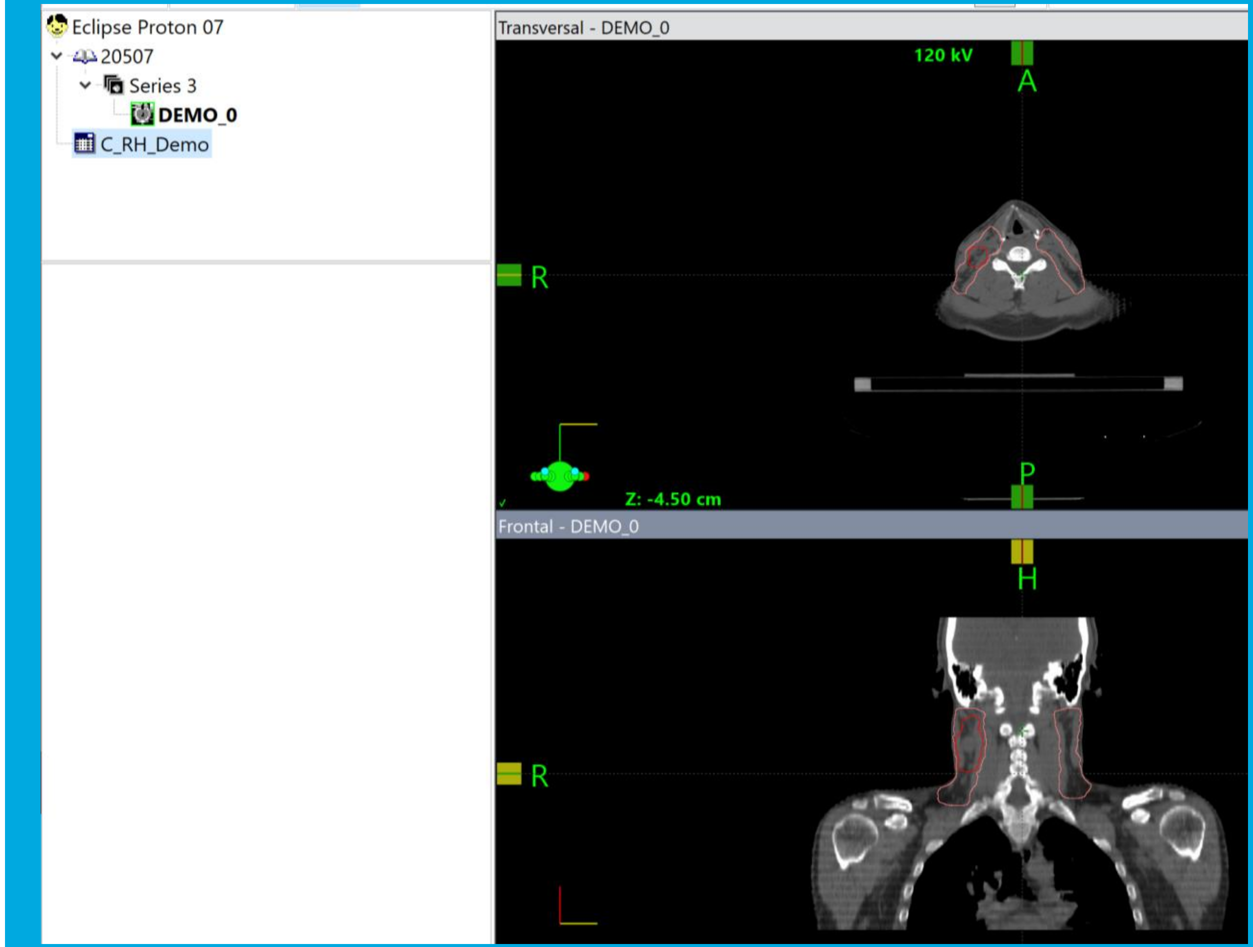
- From Varian plan library
  - medicalaffairs.varian.com/probeam-case-studies
  - HNC Patient Data & RapidPlan Model
- Generic ProBeam machine & beam data.



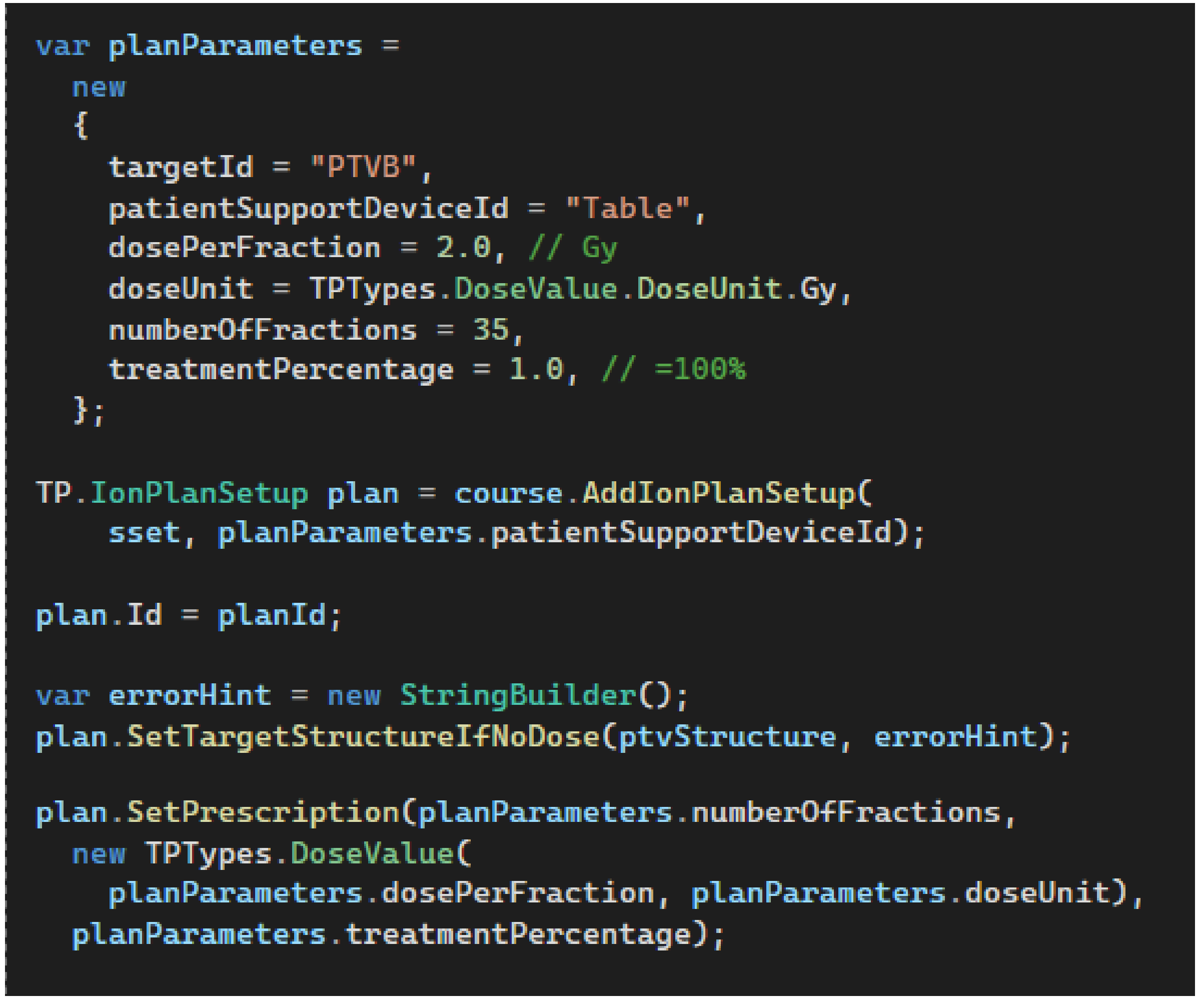
```
[STAThread]
0 references | 0 changes | 0 authors, 0 changes
static void Main(string[] args)
{
  Console.WriteLine("Hello, World.");

  (string patientId, string courseId,
   string planId, string ssetId) =
   (args[0], args[1], args[2], args[3]);

  try
  {
    using (var App = TP.Application.CreateApplication())
    {
      TP.Patient patient = App.OpenPatientById(patientId);
      patient.BeginModifications();

      TP.StructureSet sset =
        patient.StructureSets.First(x => x.Id == ssetId);

      TP.Structure ptvStructure =
        sset.Structures.First(x => x.Id == "PTVB");

      TP.Course course = patient.AddCourse();
      course.Id = courseId;
```

Accessing Patient Data

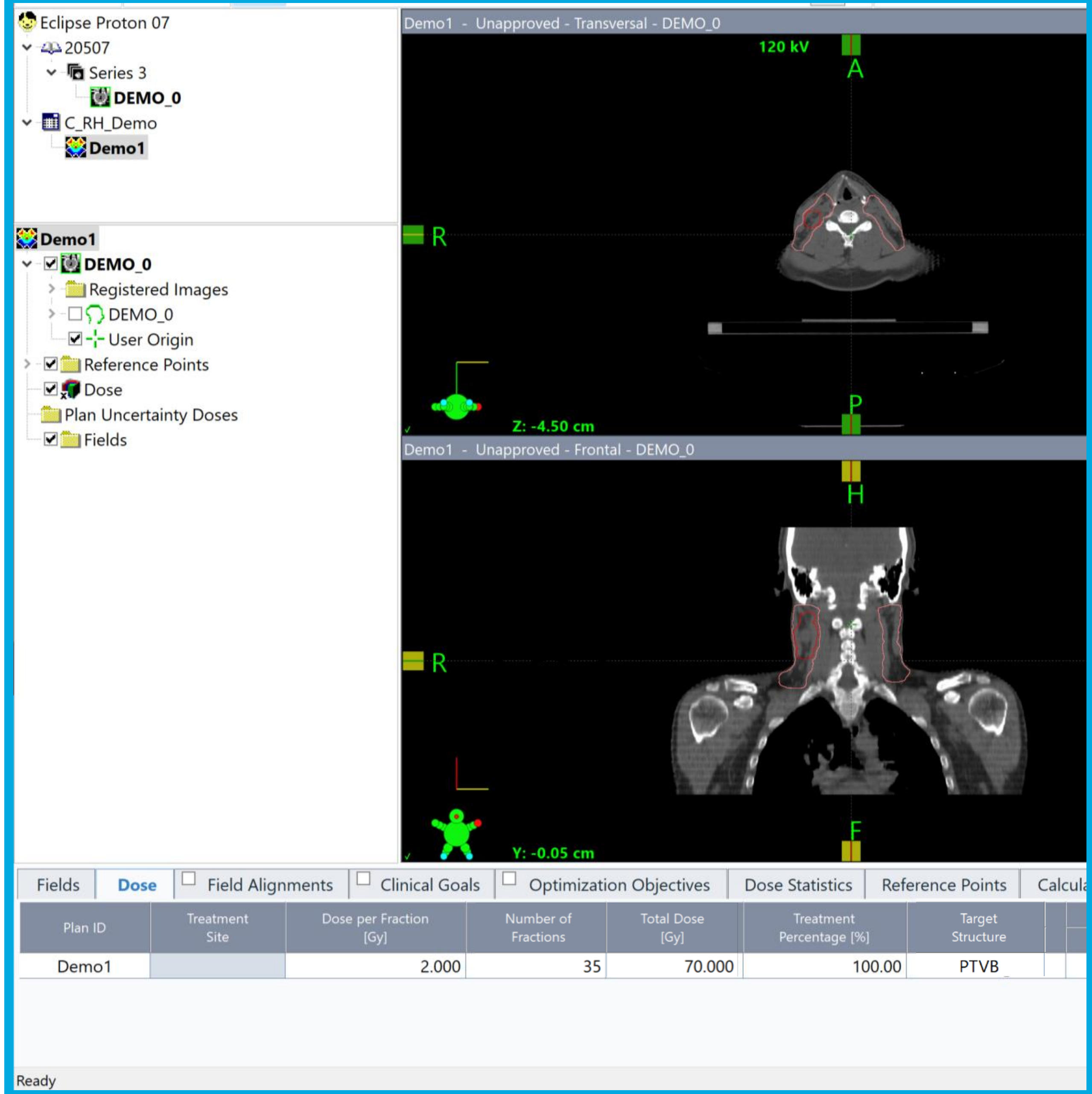
- Open data for read/write access.
- Create a new course for us.



Creating IMPT Plan

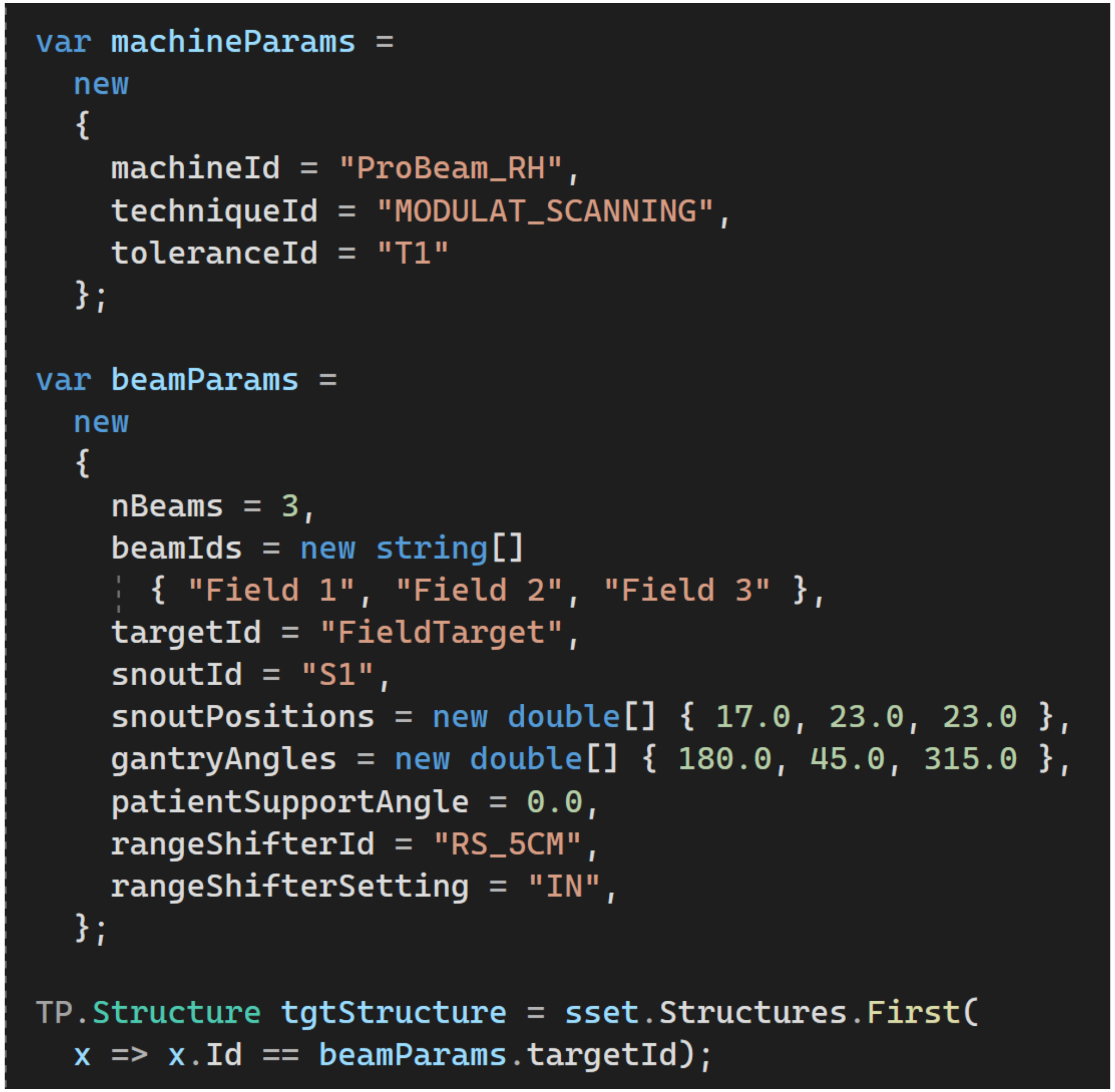


```
var planParameters =
  new
  {
    targetId = "PTVB",
    patientSupportDeviceId = "Table",
    dosePerFraction = 2.0, // Gy
    doseUnit = TPTypes.DoseValue.DoseUnit.Gy,
    numberOfFractions = 35,
    treatmentPercentage = 1.0, // =100%
  };

TP.IonPlanSetup plan = course.AddIonPlanSetup(
    sset, planParameters.patientSupportDeviceId);

plan.Id = planId;

var errorHint = new StringBuilder();
plan.SetTargetStructureIfNoDose(ptvStructure, errorHint);

plan.SetPrescription(planParameters.numberOfFractions,
  new TPTypes.DoseValue(
    planParameters.dosePerFraction, planParameters.doseUnit),
  planParameters.treatmentPercentage);
```

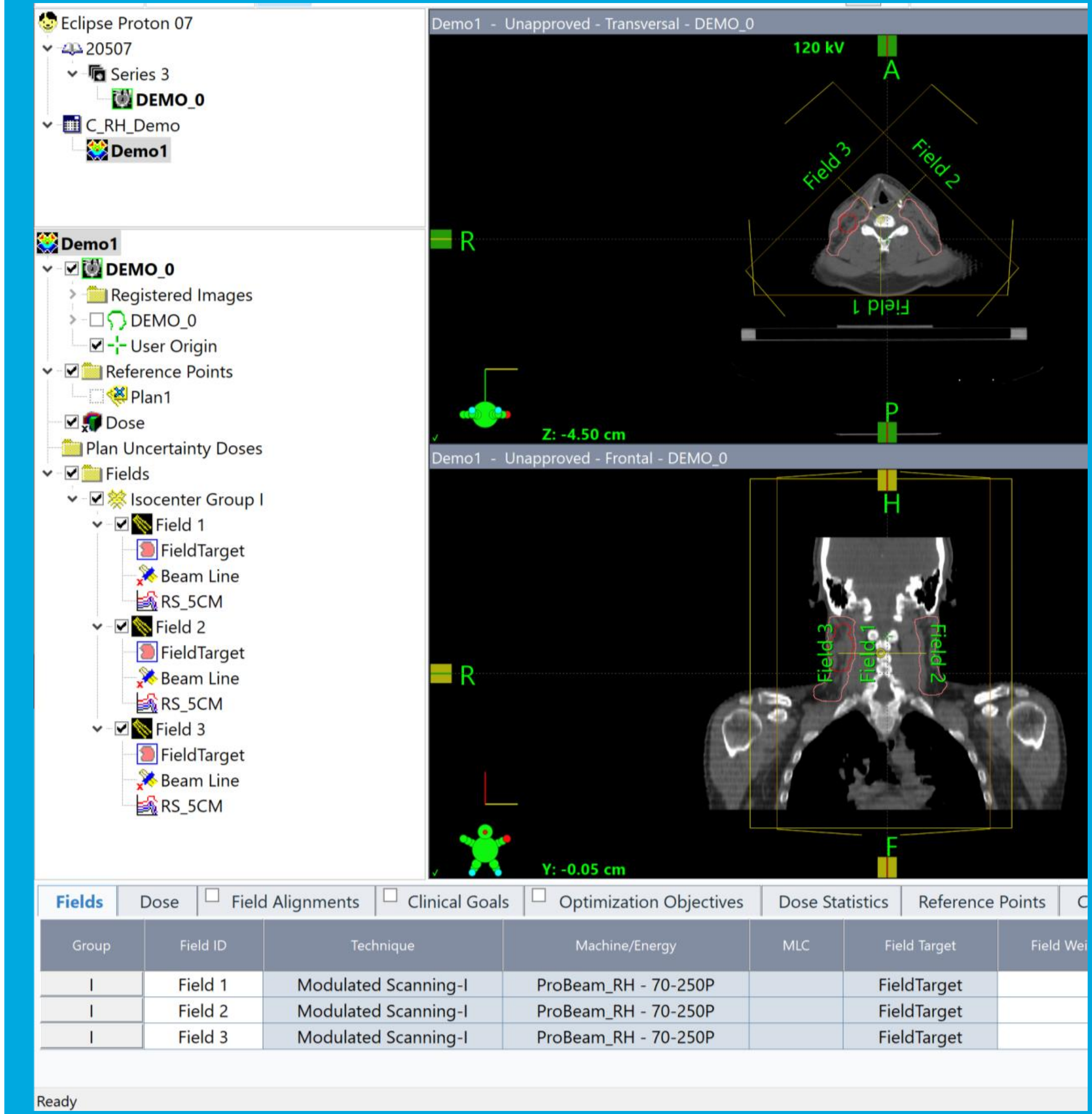


Plan ID	Treatment Site	Dose per Fraction [Gy]	Number of Fractions	Total Dose [Gy]	Treatment Percentage [%]	Target Structure
Demo1		2.000	35	70.000	100.00	PTVB

Adding Fields (1/2)



```
var machineParams =
  new
  {
    machineId = "ProBeam_RH",
    techniqueId = "MODULAT_SCANNING",
    toleranceId = "T1"
  };

var beamParams =
  new
  {
    nBeams = 3,
    beamIds = new string[]
      { "Field 1", "Field 2", "Field 3" },
    targetId = "FieldTarget",
    snoutId = "S1",
    snoutPositions = new double[] { 17.0, 23.0, 23.0 },
    gantryAngles = new double[] { 180.0, 45.0, 315.0 },
    patientSupportAngle = 0.0,
    rangeShifterId = "RS_5CM",
    rangeShifterSetting = "IN",
  };

TP.Structure tgtStructure = sset.Structures.First(
  x => x.Id == beamParams.targetId);
```



```
Eclipse Proton 07
  20507
    Series 3
      DEMO_0
  C_RH_Demo
    Demo1

Demo1
  ☑ DEMO_0
      Registered Images
      □ DEMO_0
      ☑ User Origin
  ☑ Reference Points
      Plan1
  ☑ Dose
  Plan Uncertainty Doses
  ☑ Fields
    ☑ Isocenter Group I
      ☑ Field 1
          FieldTarget
          Beam Line
          RS_5CM
      ☑ Field 2
          FieldTarget
          Beam Line
          RS_5CM
      ☑ Field 3
          FieldTarget
          Beam Line
          RS_5CM
```

Demo1 - Unapproved - Transversal - DEMO_0

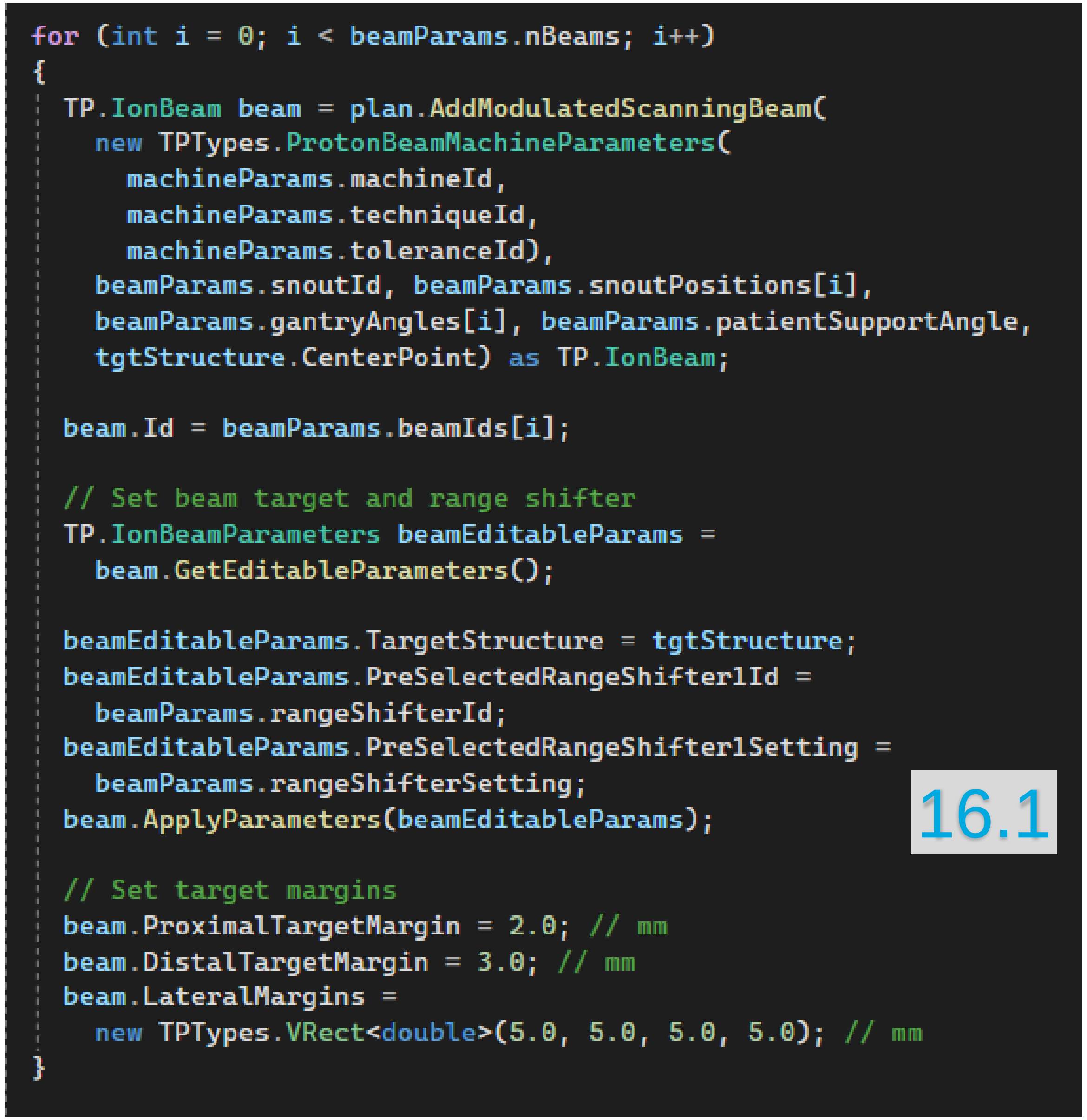
Demo1 - Unapproved - Frontal - DEMO_0

Fields | Dose | □ Field Alignments | □ Clinical Goals | □ Optimization Objectives | Dose Statistics | Reference Points | C

Group	Field ID	Technique	Machine/Energy	MLC	Field Target	Field Wei
I	Field 1	Modulated Scanning-I	ProBeam_RH - 70-250P		FieldTarget	
I	Field 2	Modulated Scanning-I	ProBeam_RH - 70-250P		FieldTarget	
I	Field 3	Modulated Scanning-I	ProBeam_RH - 70-250P		FieldTarget	

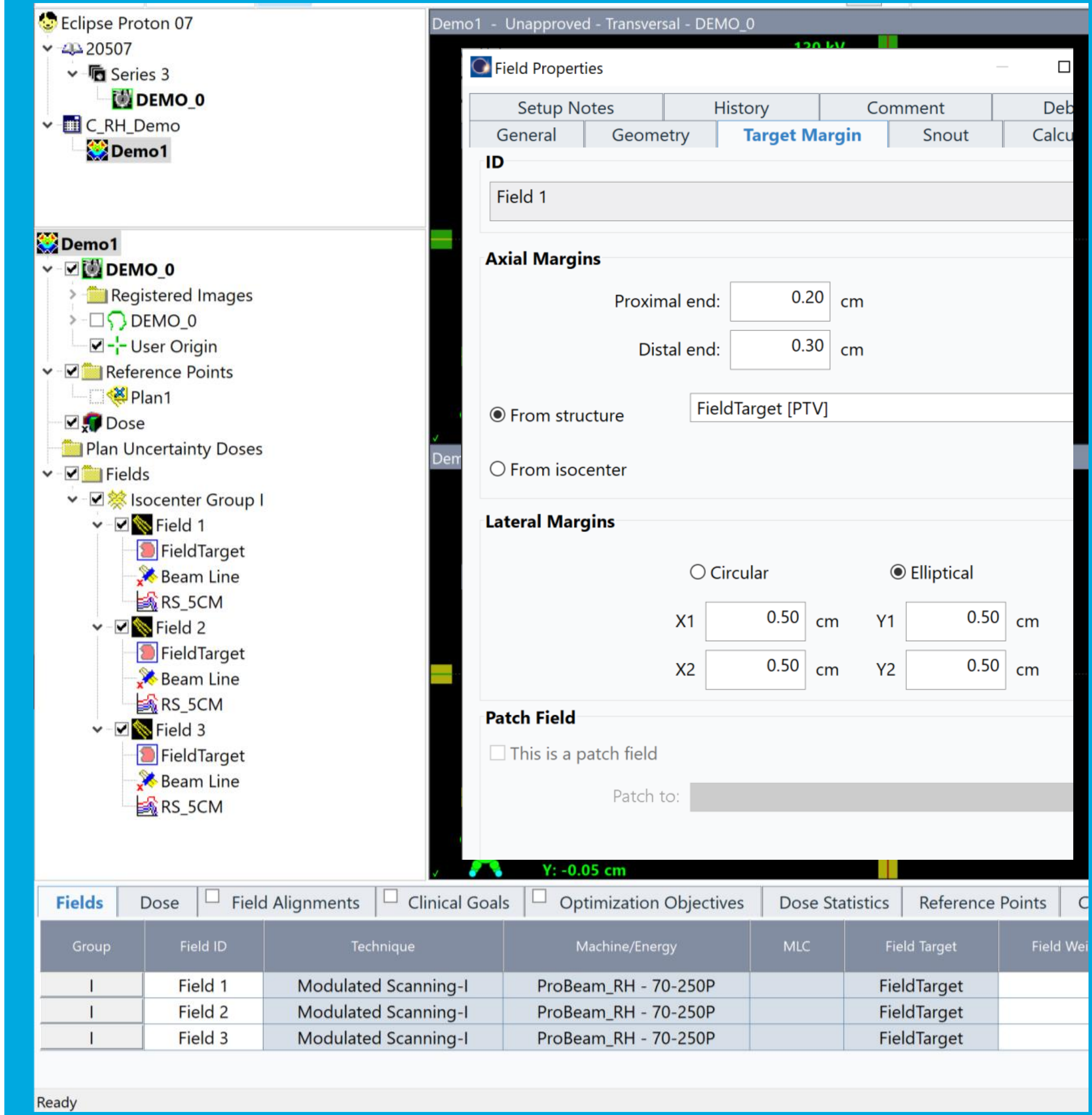
Ready

Adding Fields (2/2)



```
for (int i = 0; i < beamParams.nBeams; i++)
{
  TP.IonBeam beam = plan.AddModulatedScanningBeam(
    new TPTypes.ProtonBeamMachineParameters(
      machineParams.machineId,
      machineParams.techniqueId,
      machineParams.toleranceId),
    beamParams.snoutId, beamParams.snoutPositions[i],
    beamParams.gantryAngles[i], beamParams.patientSupportAngle,
    tgtStructure.CenterPoint) as TP.IonBeam;

  beam.Id = beamParams.beamIds[i];

  // Set beam target and range shifter
  TP.IonBeamParameters beamEditableParams =
    beam.GetEditableParameters();

  beamEditableParams.TargetStructure = tgtStructure;
  beamEditableParams.PreSelectedRangeShifter1Id =
    beamParams.rangeShifterId;
  beamEditableParams.PreSelectedRangeShifter1Setting =
    beamParams.rangeShifterSetting;
  beam.ApplyParameters(beamEditableParams);

  // Set target margins
  beam.ProximalTargetMargin = 2.0; // mm
  beam.DistalTargetMargin = 3.0; // mm
  beam.LateralMargins =
    new TPTypes.VRect<double>(5.0, 5.0, 5.0, 5.0); // mm
}
```

16.1



Field Properties

Setup Notes | History | Comment | Deb
General | Geometry | **Target Margin** | Snout | Calcu

ID: Field 1

Axial Margins

Proximal end: 0.20 cm

Distal end: 0.30 cm

● From structure: FieldTarget [PTV]

○ From isocenter

Lateral Margins

○ Circular ● Elliptical

X1 0.50 cm Y1 0.50 cm

X2 0.50 cm Y2 0.50 cm

Patch Field

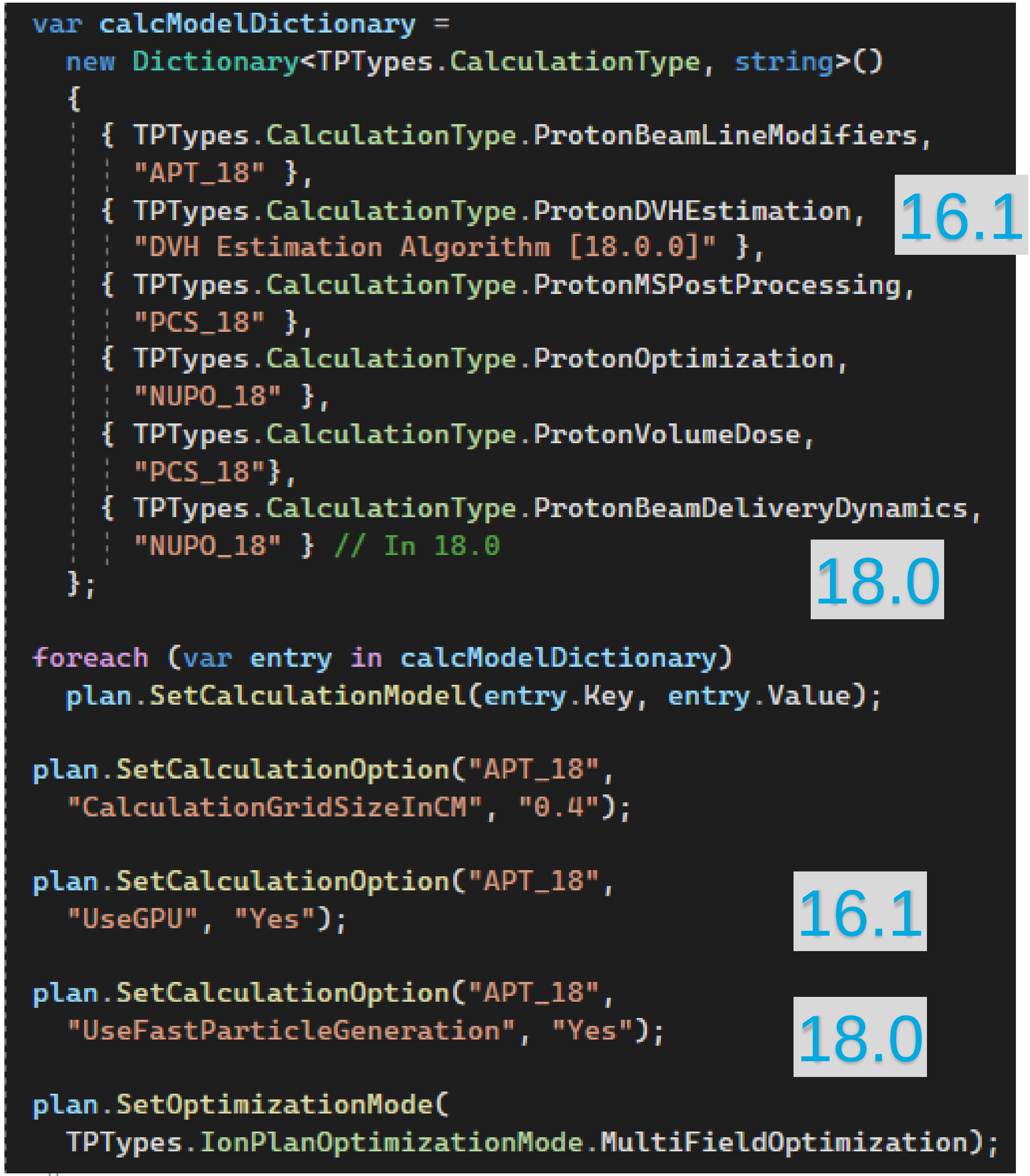
□ This is a patch field

Patch to:

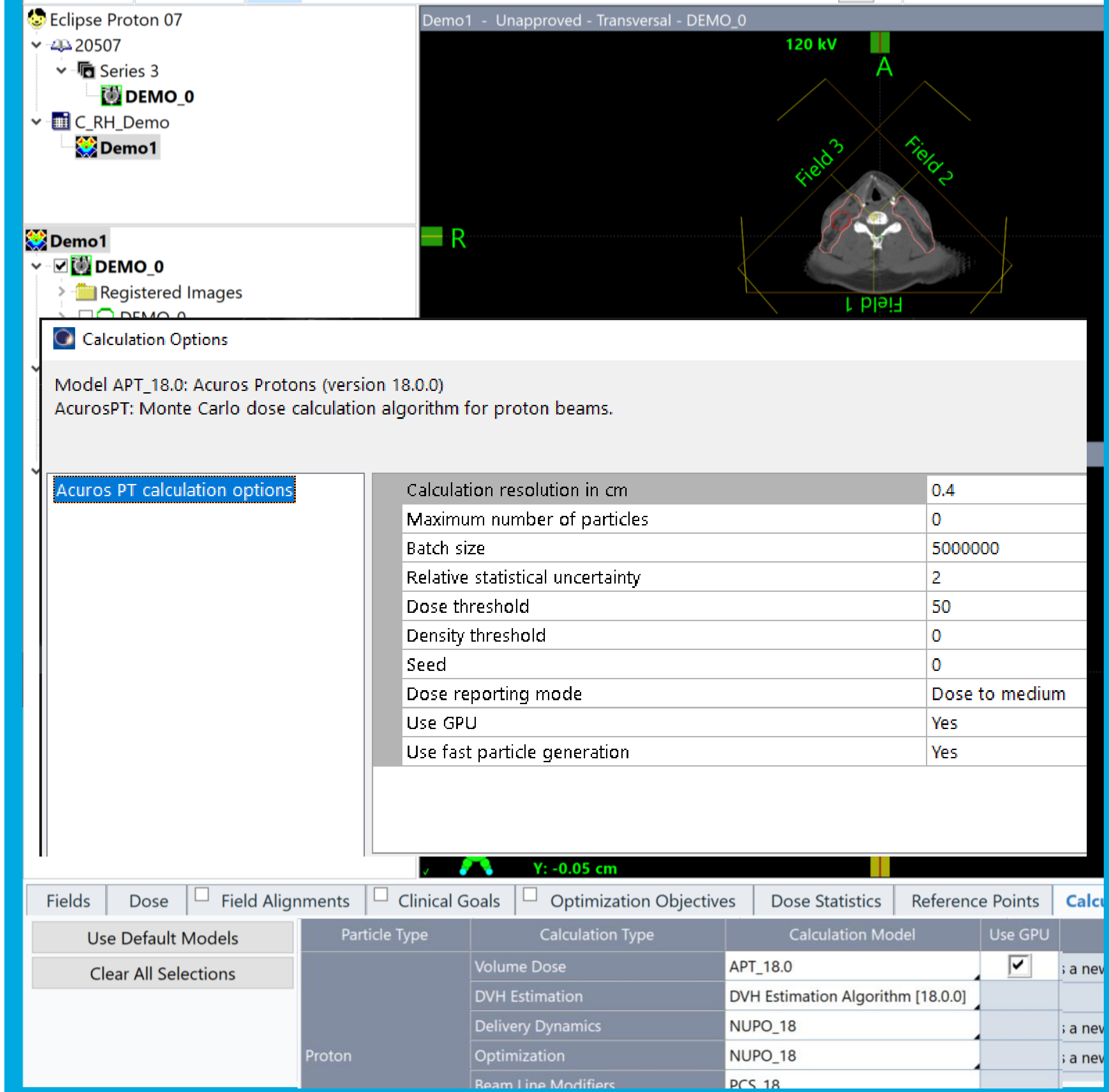
Fields | Dose | □ Field Alignments | □ Clinical Goals | □ Optimization Objectives | Dose Statistics | Reference Points | C

Group	Field ID	Technique	Machine/Energy	MLC	Field Target	Field Wei
I	Field 1	Modulated Scanning-I	ProBeam_RH - 70-250P		FieldTarget	
I	Field 2	Modulated Scanning-I	ProBeam_RH - 70-250P		FieldTarget	
I	Field 3	Modulated Scanning-I	ProBeam_RH - 70-250P		FieldTarget	

Setting Calculation Models



```
var calcModelDictionary =
  new Dictionary<TPTypes.CalculationType, string>()
  {
    { TPTypes.CalculationType.ProtonBeamLineModifiers,
      "APT_18" },
    { TPTypes.CalculationType.ProtonDVHEstimation,
      "DVH Estimation Algorithm [18.0.0]" },
    { TPTypes.CalculationType.ProtonMSPostProcessing,
      "PCS_18" },
    { TPTypes.CalculationType.ProtonOptimization,
      "NUPO_18" },
    { TPTypes.CalculationType.ProtonVolumeDose,
      "PCS_18"},
    { TPTypes.CalculationType.ProtonBeamDeliveryDynamics,
      "NUPO_18" } // In 18.0
  };

foreach (var entry in calcModelDictionary)
  plan.SetCalculationModel(entry.Key, entry.Value);

plan.SetCalculationOption("APT_18",
  "CalculationGridSizeInCM", "0.4");

plan.SetCalculationOption("APT_18",
  "UseGPU", "Yes");

plan.SetCalculationOption("APT_18",
  "UseFastParticleGeneration", "Yes");

plan.SetOptimizationMode(
  TPTypes.IonPlanOptimizationMode.MultiFieldOptimization);
```



Calculation Options

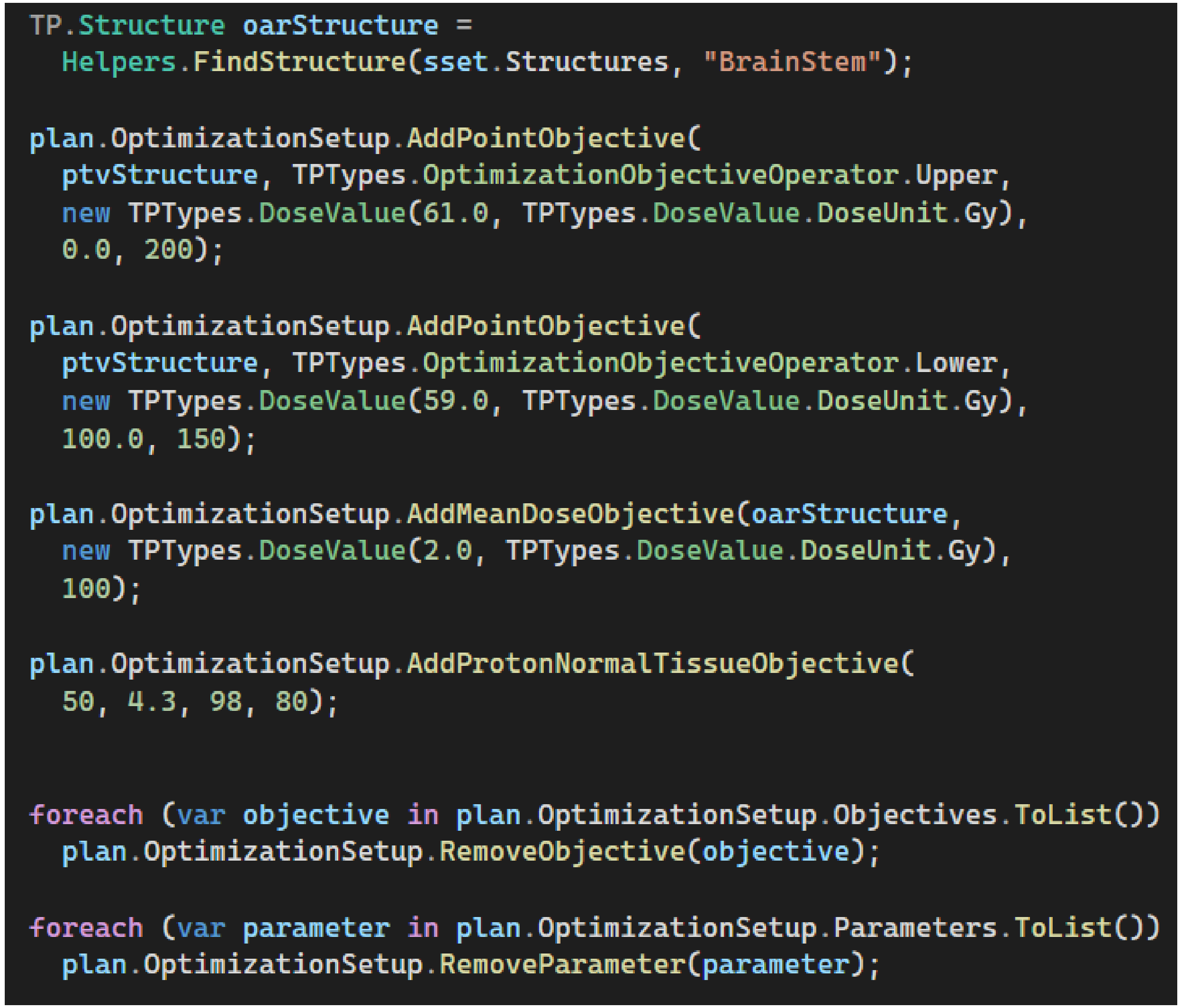
Model APT_18.0: Acuros Protons (version 18.0.0)
AcurosPT: Monte Carlo dose calculation algorithm for proton beams.

Acuros PT calculation options

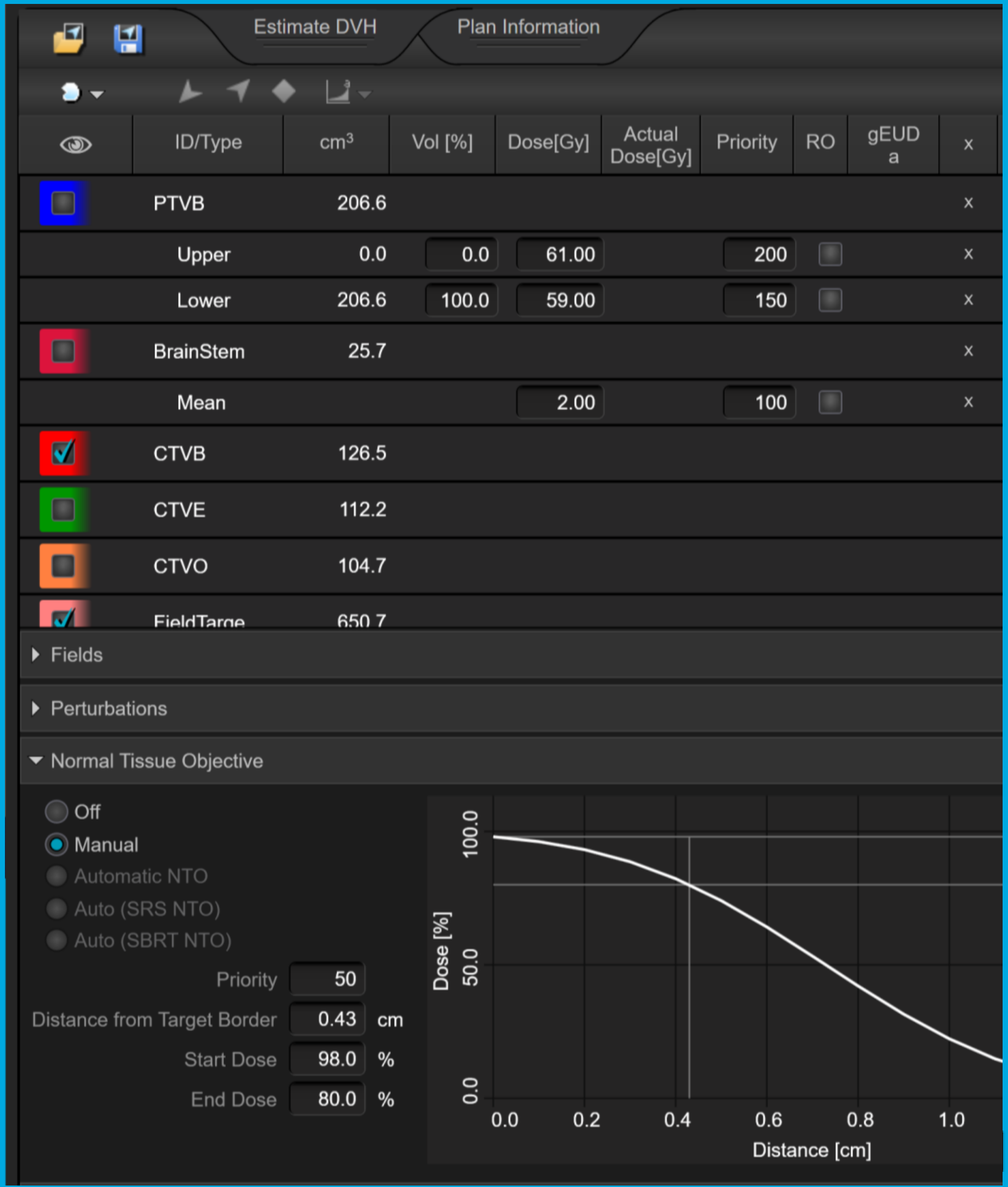
Calculation resolution in cm	0.4
Maximum number of particles	0
Batch size	5000000
Relative statistical uncertainty	2
Dose threshold	50
Density threshold	0
Seed	0
Dose reporting mode	Dose to medium
Use GPU	Yes
Use fast particle generation	Yes

Particle Type	Calculation Type	Calculation Model	Use GPU	
Proton	Volume Dose	APT_18.0	☑	; a nev
	DVH Estimation	DVH Estimation Algorithm [18.0.0]		
	Delivery Dynamics	NUPO_18		; a nev
	Optimization	NUPO_18		; a nev
	Beam Line Modifiers	PCS_18		

Adding Optimization Objectives



```
TP.Structure oarStructure =
  Helpers.FindStructure(sset.Structures, "BrainStem");

plan.OptimizationSetup.AddPointObjective(
  ptvStructure, TPTypes.OptimizationObjectiveOperator.Upper,
  new TPTypes.DoseValue(61.0, TPTypes.DoseValue.DoseUnit.Gy),
  0.0, 200);

plan.OptimizationSetup.AddPointObjective(
  ptvStructure, TPTypes.OptimizationObjectiveOperator.Lower,
  new TPTypes.DoseValue(59.0, TPTypes.DoseValue.DoseUnit.Gy),
  100.0, 150);

plan.OptimizationSetup.AddMeanDoseObjective(oarStructure,
  new TPTypes.DoseValue(2.0, TPTypes.DoseValue.DoseUnit.Gy),
  100);

plan.OptimizationSetup.AddProtonNormalTissueObjective(
  50, 4.3, 98, 80);


foreach (var objective in plan.OptimizationSetup.Objectives.ToList())
  plan.OptimizationSetup.RemoveObjective(objective);

foreach (var parameter in plan.OptimizationSetup.Parameters.ToList())
  plan.OptimizationSetup.RemoveParameter(parameter);
```



	ID/Type	cm³	Vol [%]	Dose[Gy]	Actual Dose[Gy]	Priority	RO	gEUD a	x
□	PTVB	206.6							x
	Upper	0.0	0.0	61.00		200	□		x
	Lower	206.6	100.0	59.00		150	□		x
□	BrainStem	25.7							x
	Mean			2.00		100	□		x
☑	CTVB	126.5							
□	CTVE	112.2							
□	CTVO	104.7							
☑	FieldTarge	650.7							

▸ Fields

▸ Perturbations

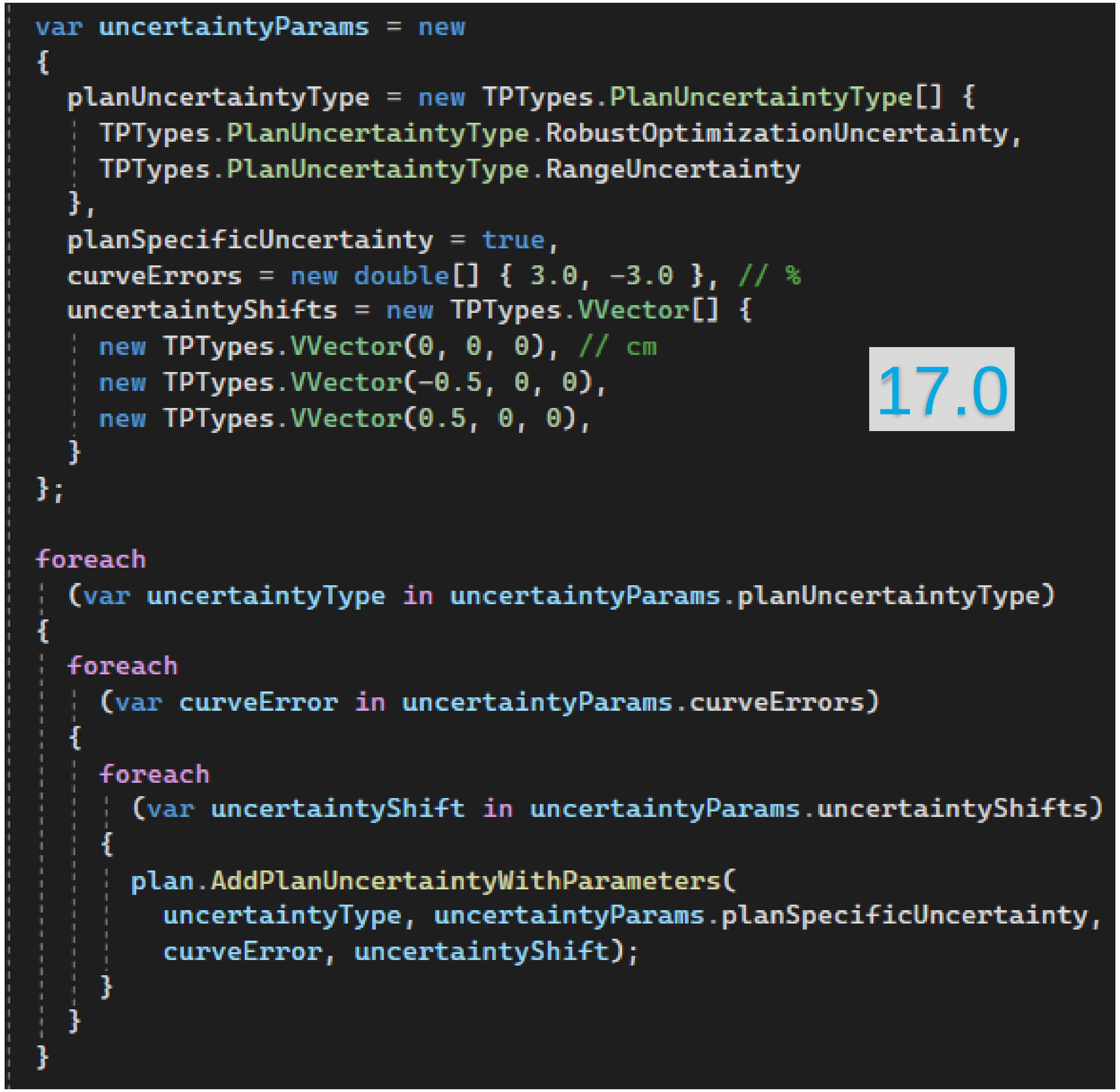
▾ Normal Tissue Objective

○ Off
● Manual
○ Automatic NTO
○ Auto (SRS NTO)
○ Auto (SBRT NTO)

Priority 50
Distance from Target Border 0.43 cm
Start Dose 98.0 %
End Dose 80.0 %

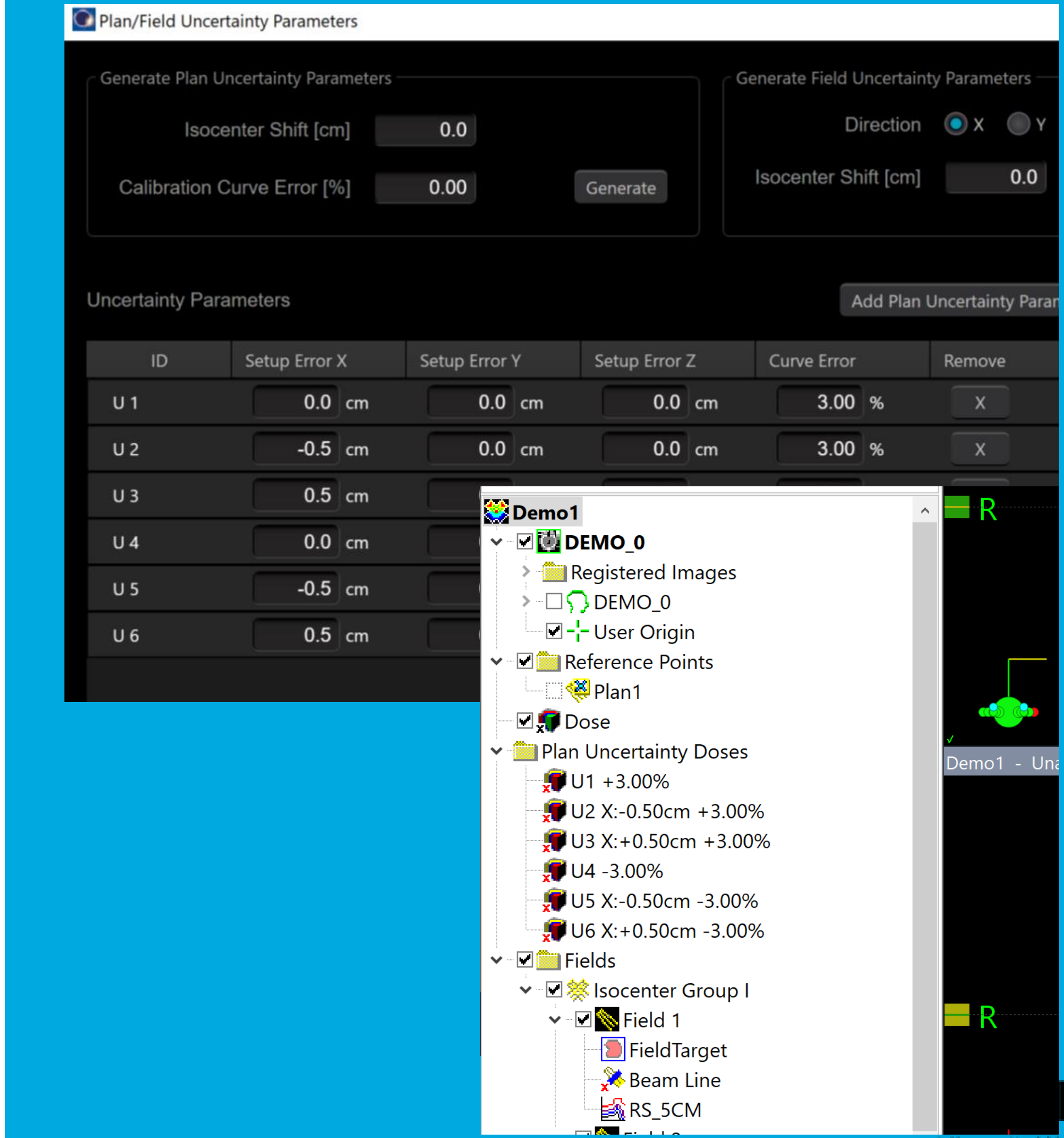
A Siemens Healthineers Company

Adding Robustness Scenarios



```
var uncertaintyParams = new
{
  planUncertaintyType = new TPTypes.PlanUncertaintyType[] {
    TPTypes.PlanUncertaintyType.RobustOptimizationUncertainty,
    TPTypes.PlanUncertaintyType.RangeUncertainty
  },
  planSpecificUncertainty = true,
  curveErrors = new double[] { 3.0, -3.0 }, // %
  uncertaintyShifts = new TPTypes.VVector[] {
    new TPTypes.VVector(0, 0, 0), // cm
    new TPTypes.VVector(-0.5, 0, 0),
    new TPTypes.VVector(0.5, 0, 0),
  }
};

foreach
  (var uncertaintyType in uncertaintyParams.planUncertaintyType)
{
  foreach
    (var curveError in uncertaintyParams.curveErrors)
  {
    foreach
      (var uncertaintyShift in uncertaintyParams.uncertaintyShifts)
    {
      plan.AddPlanUncertaintyWithParameters(
        uncertaintyType, uncertaintyParams.planSpecificUncertainty,
        curveError, uncertaintyShift);
    }
  }
}
```

17.0



Plan/Field Uncertainty Parameters

Generate Plan Uncertainty Parameters

Isocenter Shift [cm]: 0.0

Calibration Curve Error [%]: 0.00

Generate Field Uncertainty Parameters

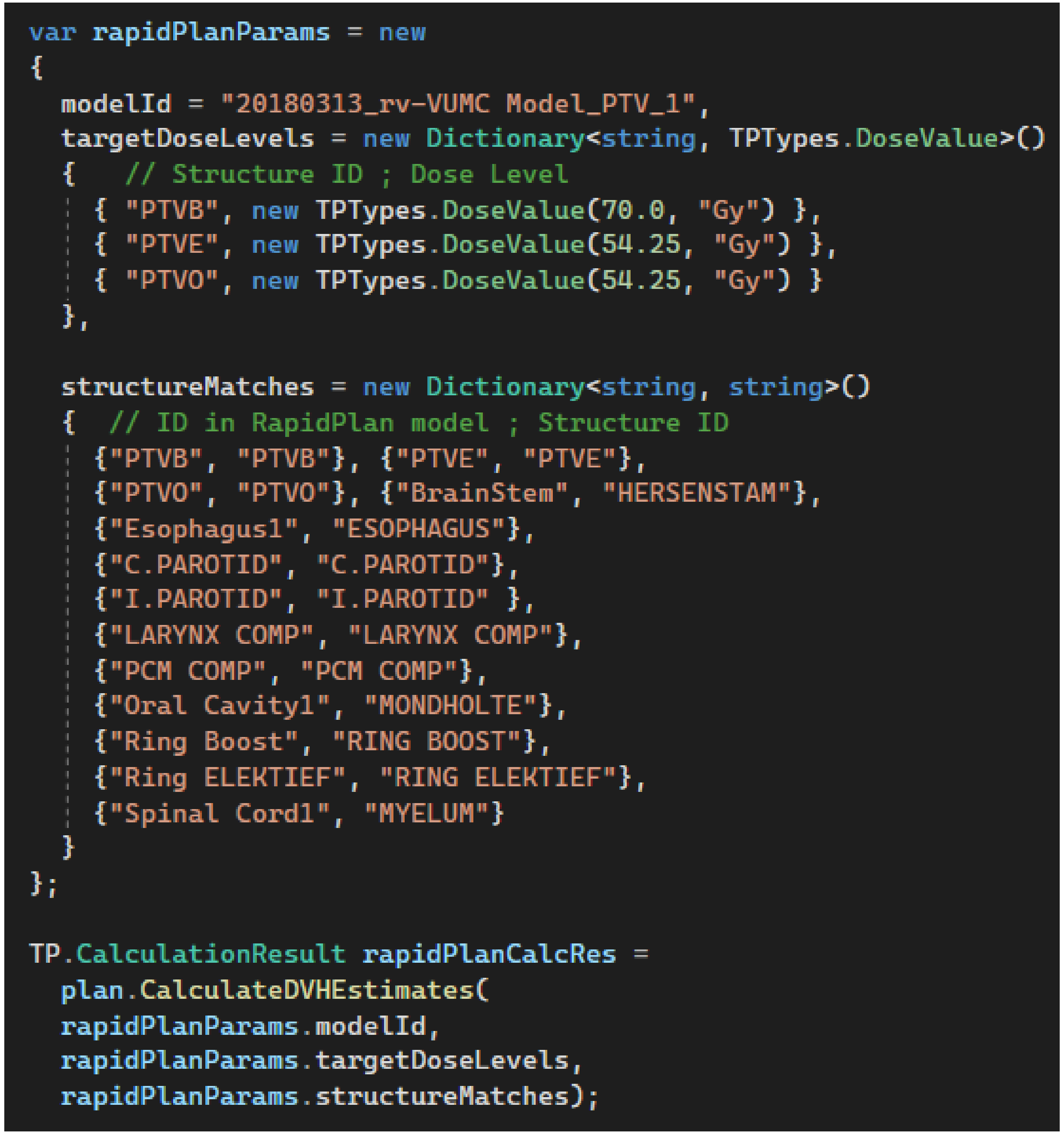
Direction: ● X ○ Y

Isocenter Shift [cm]: 0.0

Uncertainty Parameters

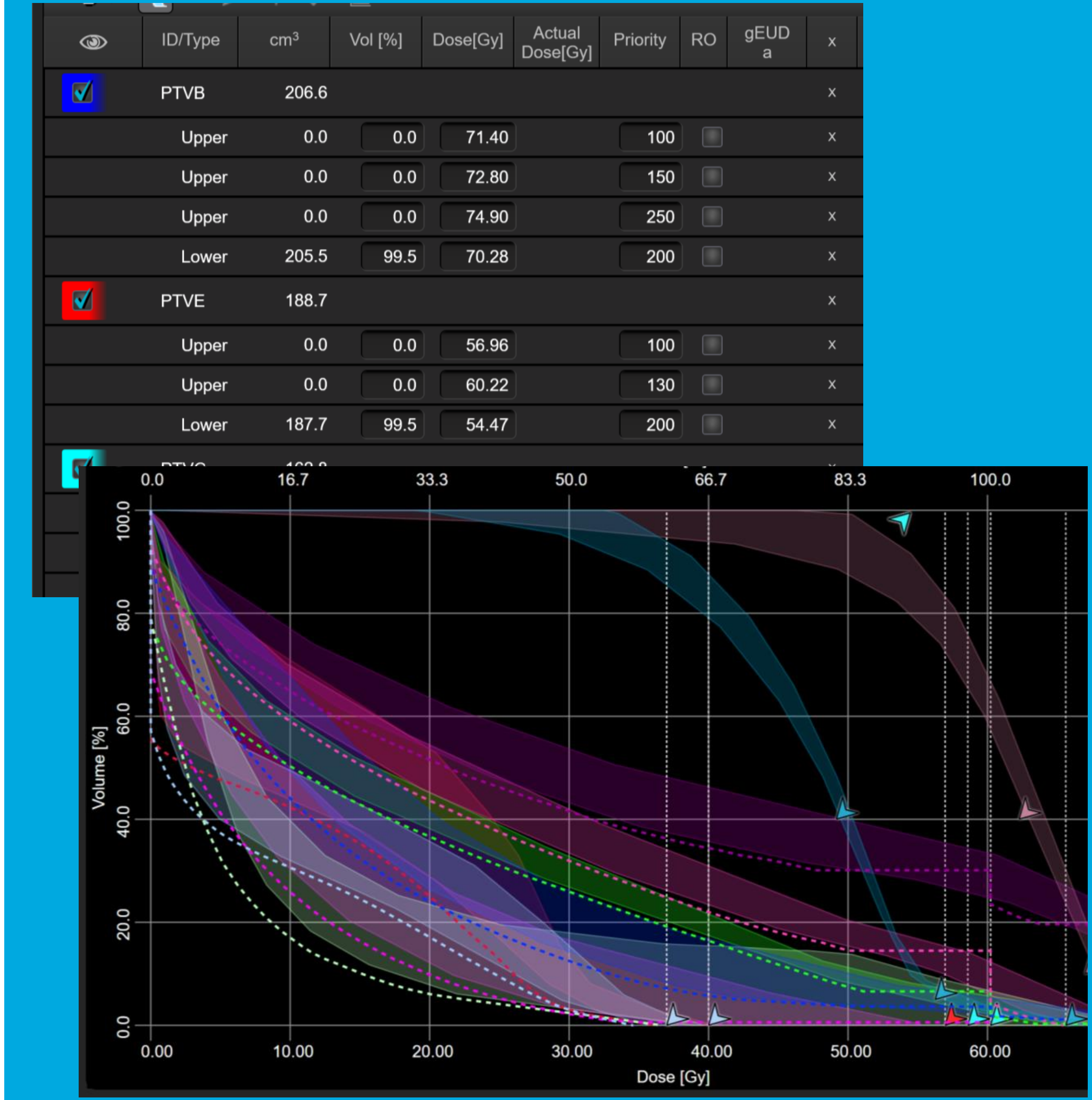
ID	Setup Error X	Setup Error Y	Setup Error Z	Curve Error	Remove
U 1	0.0 cm	0.0 cm	0.0 cm	3.00 %	X
U 2	-0.5 cm	0.0 cm	0.0 cm	3.00 %	X
U 3	0.5 cm				
U 4	0.0 cm				
U 5	-0.5 cm				
U 6	0.5 cm				

- Demo1
  - ☑ DEMO_0
    - Registered Images
    - □ DEMO_0
    - ☑ User Origin
  - ☑ Reference Points
    - Plan1
  - ☑ Dose
  - Plan Uncertainty Doses
    - U1 +3.00%
    - U2 X:-0.50cm +3.00%
    - U3 X:+0.50cm +3.00%
    - U4 -3.00%
    - U5 X:-0.50cm -3.00%
    - U6 X:+0.50cm -3.00%
  - ☑ Fields
    - ☑ Isocenter Group I
      - ☑ Field 1
        - FieldTarget
        - Beam Line
        - RS_5CM



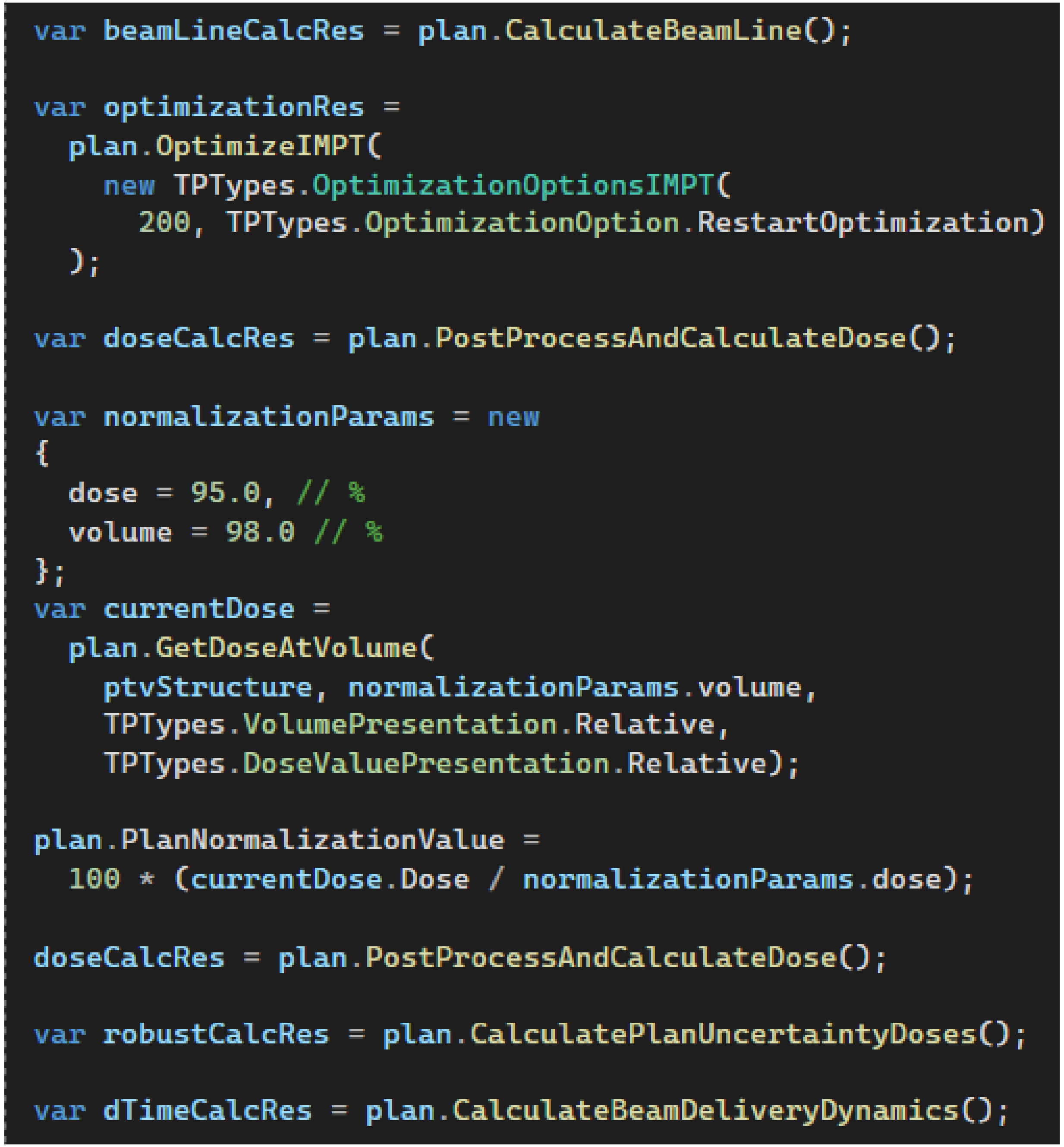
```
var rapidPlanParams = new
{
  modelId = "20180313_rv-VUMC Model_PTV_1",
  targetDoseLevels = new Dictionary<string, TPTypes.DoseValue>()
  {   // Structure ID ; Dose Level
    { "PTVB", new TPTypes.DoseValue(70.0, "Gy") },
    { "PTVE", new TPTypes.DoseValue(54.25, "Gy") },
    { "PTVO", new TPTypes.DoseValue(54.25, "Gy") }
  },

  structureMatches = new Dictionary<string, string>()
  {  // ID in RapidPlan model ; Structure ID
    {"PTVB", "PTVB"}, {"PTVE", "PTVE"},
    {"PTVO", "PTVO"}, {"BrainStem", "HERSENSTAM"},
    {"Esophagus1", "ESOPHAGUS"},
    {"C.PAROTID", "C.PAROTID"},
    {"I.PAROTID", "I.PAROTID" },
    {"LARYNX COMP", "LARYNX COMP"},
    {"PCM COMP", "PCM COMP"},
    {"Oral Cavity1", "MONDHOLTE"},
    {"Ring Boost", "RING BOOST"},
    {"Ring ELEKTIEF", "RING ELEKTIEF"},
    {"Spinal Cord1", "MYELUM"}
  }
};

TP.CalculationResult rapidPlanCalcRes =
  plan.CalculateDVHEstimates(
  rapidPlanParams.modelId,
  rapidPlanParams.targetDoseLevels,
  rapidPlanParams.structureMatches);
```

RapidPlanning

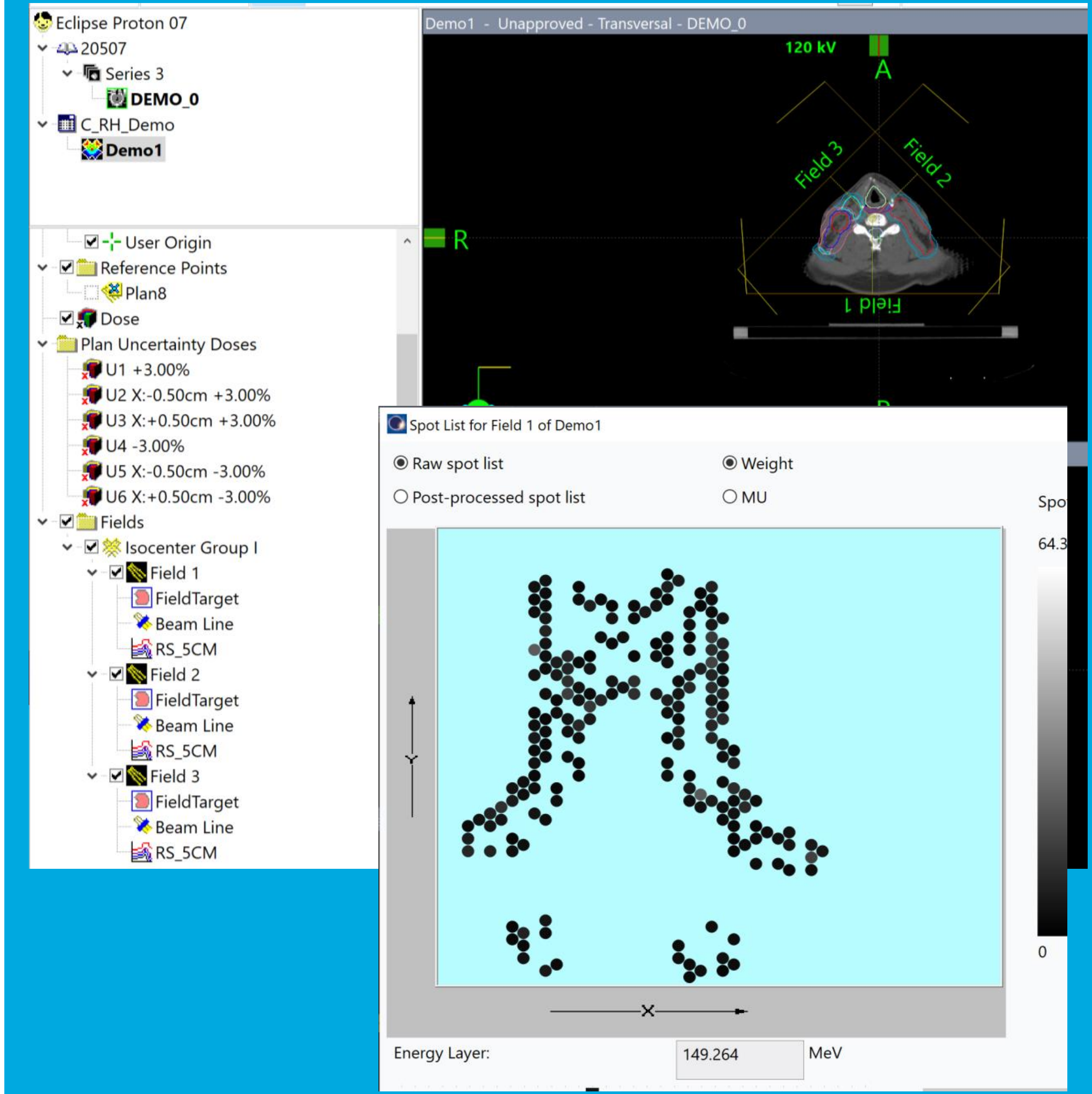


👁	ID/Type	cm³	Vol [%]	Dose[Gy]	Actual Dose[Gy]	Priority	RO	gEUD a	x
☑	PTVB	206.6							x
	Upper	0.0	0.0	71.40		100	□		x
	Upper	0.0	0.0	72.80		150	□		x
	Upper	0.0	0.0	74.90		250	□		x
	Lower	205.5	99.5	70.28		200	□		x
☑	PTVE	188.7							x
	Upper	0.0	0.0	56.96		100	□		x
	Upper	0.0	0.0	60.22		130	□		x
	Lower	187.7	99.5	54.47		200	□		x

Calculations



```
var beamLineCalcRes = plan.CalculateBeamLine();

var optimizationRes =
  plan.OptimizeIMPT(
    new TPTypes.OptimizationOptionsIMPT(
      200, TPTypes.OptimizationOption.RestartOptimization)
  );

var doseCalcRes = plan.PostProcessAndCalculateDose();

var normalizationParams = new
{
  dose = 95.0, // %
  volume = 98.0 // %
};
var currentDose =
  plan.GetDoseAtVolume(
    ptvStructure, normalizationParams.volume,
    TPTypes.VolumePresentation.Relative,
    TPTypes.DoseValuePresentation.Relative);

plan.PlanNormalizationValue =
  100 * (currentDose.Dose / normalizationParams.dose);

doseCalcRes = plan.PostProcessAndCalculateDose();

var robustCalcRes = plan.CalculatePlanUncertaintyDoses();

var dTimeCalcRes = plan.CalculateBeamDeliveryDynamics();
```



Eclipse Proton 07
- 20507
  - Series 3
    - DEMO_0
  - C_RH_Demo
    - Demo1

- ☑ User Origin
- ☑ Reference Points
  - □ Plan8
- ☑ Dose
- Plan Uncertainty Doses
  - U1 +3.00%
  - U2 X:-0.50cm +3.00%
  - U3 X:+0.50cm +3.00%
  - U4 -3.00%
  - U5 X:-0.50cm -3.00%
  - U6 X:+0.50cm -3.00%
- ☑ Fields
  - ☑ Isocenter Group I
    - ☑ Field 1
      - FieldTarget
      - Beam Line
      - RS_5CM
    - ☑ Field 2
      - FieldTarget
      - Beam Line
      - RS_5CM
    - ☑ Field 3
      - FieldTarget
      - Beam Line
      - RS_5CM

Demo1 - Unapproved - Transversal - DEMO_0

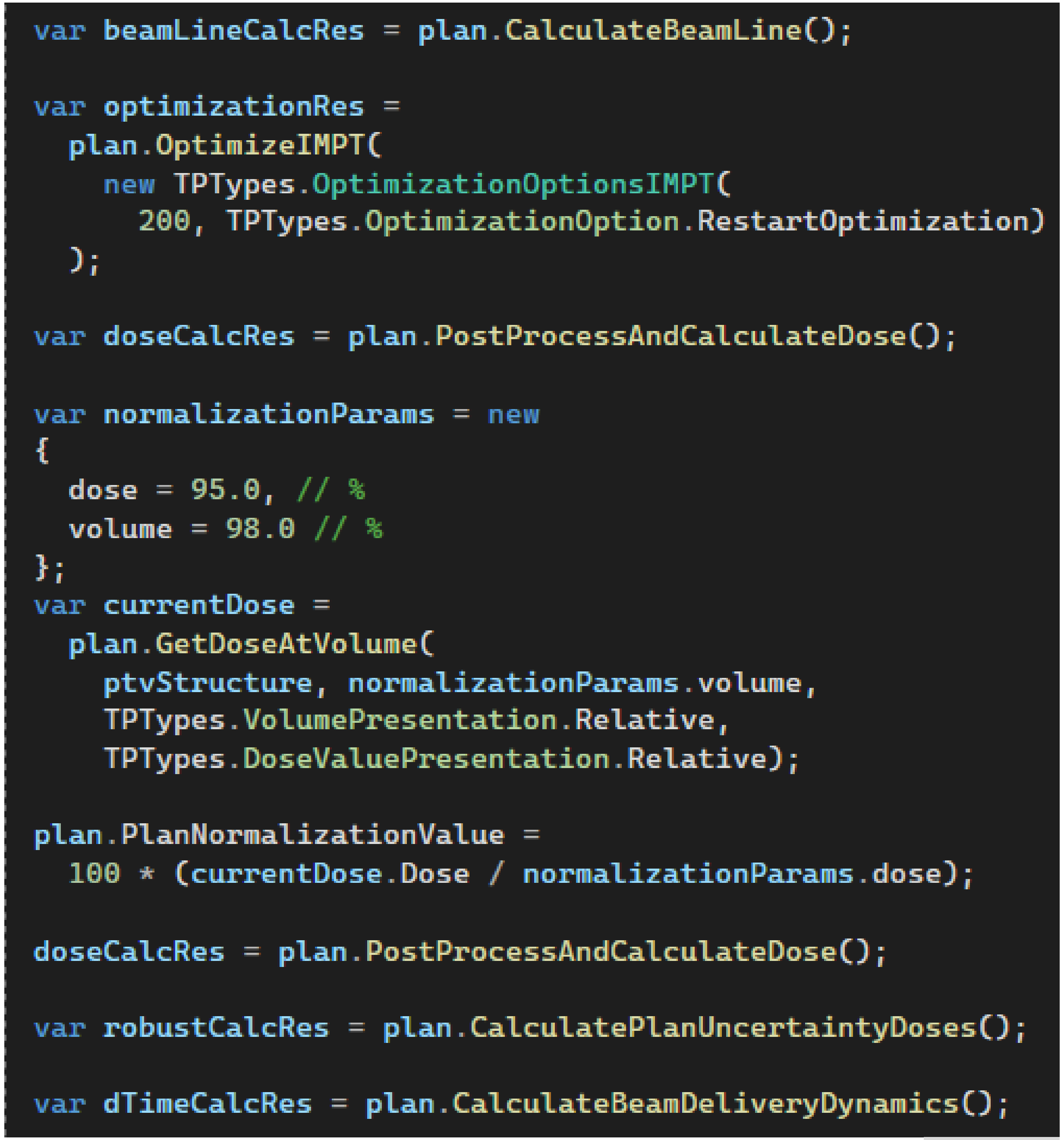
Spot List for Field 1 of Demo1

● Raw spot list ● Weight

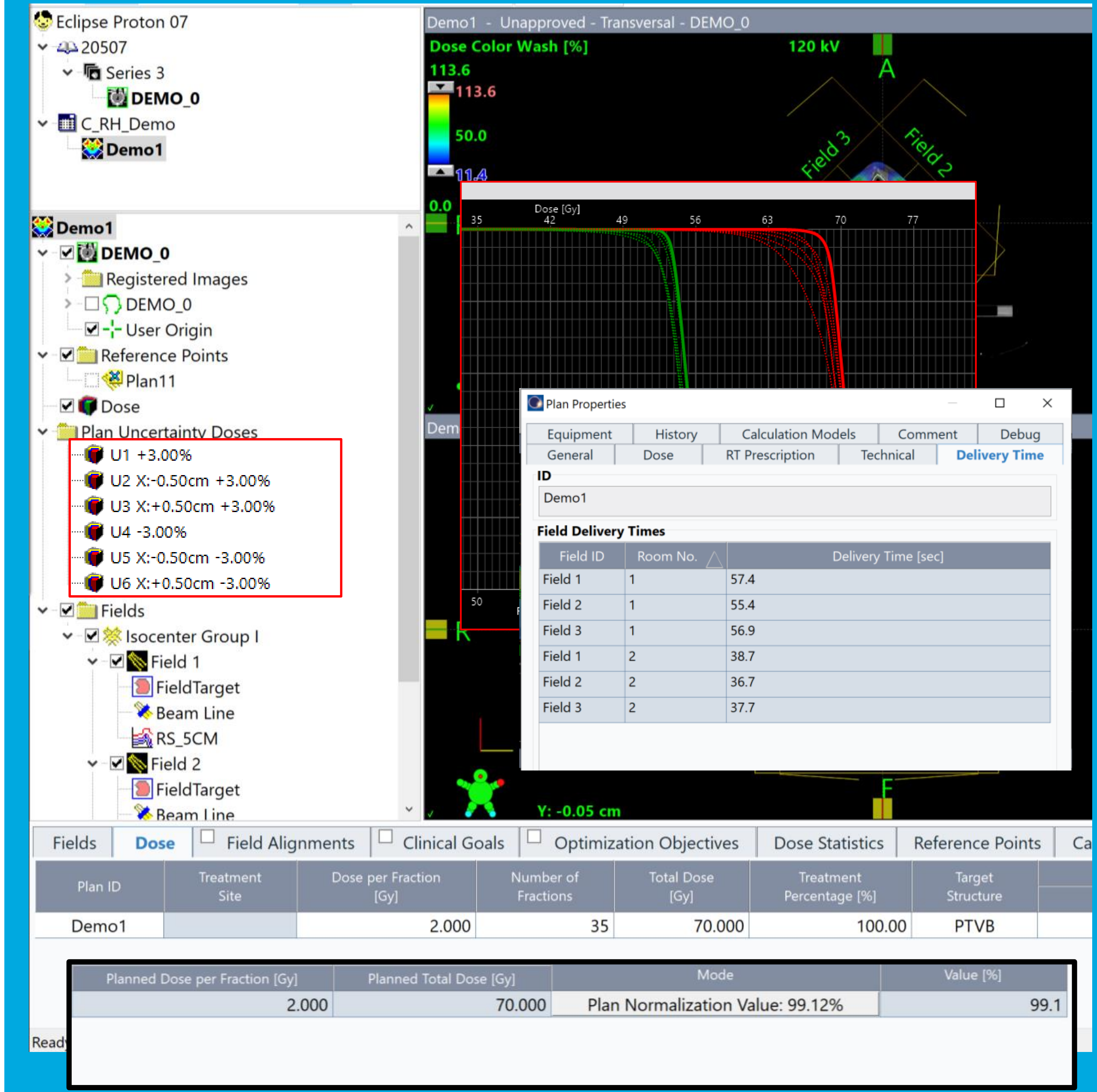
○ Post-processed spot list ○ MU

Energy Layer: 149.264 MeV

Calculations



```
var beamLineCalcRes = plan.CalculateBeamLine();

var optimizationRes =
  plan.OptimizeIMPT(
    new TPTypes.OptimizationOptionsIMPT(
      200, TPTypes.OptimizationOption.RestartOptimization)
  );

var doseCalcRes = plan.PostProcessAndCalculateDose();

var normalizationParams = new
{
  dose = 95.0, // %
  volume = 98.0 // %
};
var currentDose =
  plan.GetDoseAtVolume(
    ptvStructure, normalizationParams.volume,
    TPTypes.VolumePresentation.Relative,
    TPTypes.DoseValuePresentation.Relative);

plan.PlanNormalizationValue =
  100 * (currentDose.Dose / normalizationParams.dose);

doseCalcRes = plan.PostProcessAndCalculateDose();

var robustCalcRes = plan.CalculatePlanUncertaintyDoses();

var dTimeCalcRes = plan.CalculateBeamDeliveryDynamics();
```



ID

Demo1

Field Delivery Times

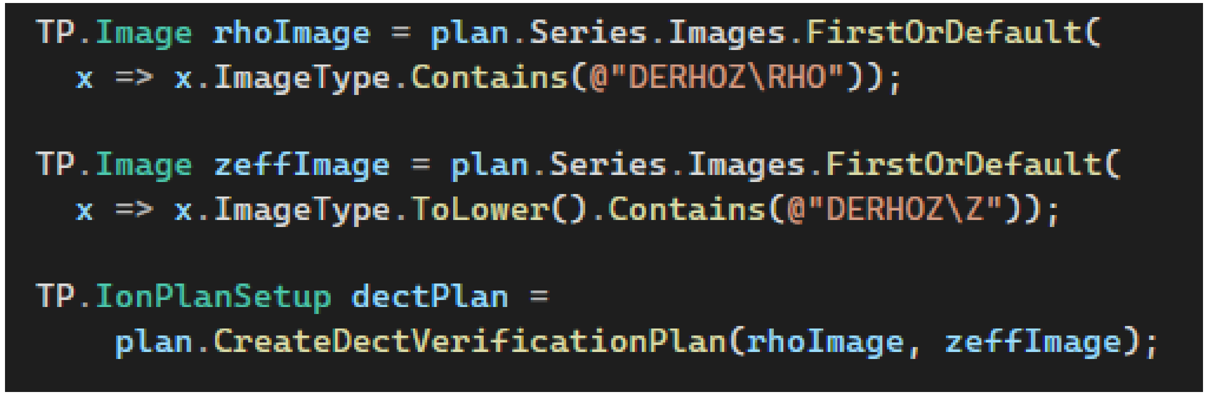
Field ID	Room No.	Delivery Time [sec]
Field 1	1	57.4
Field 2	1	55.4
Field 3	1	56.9
Field 1	2	38.7
Field 2	2	36.7
Field 3	2	37.7

Plan ID	Treatment Site	Dose per Fraction [Gy]	Number of Fractions	Total Dose [Gy]	Treatment Percentage [%]	Target Structure
Demo1		2.000	35	70.000	100.00	PTVB

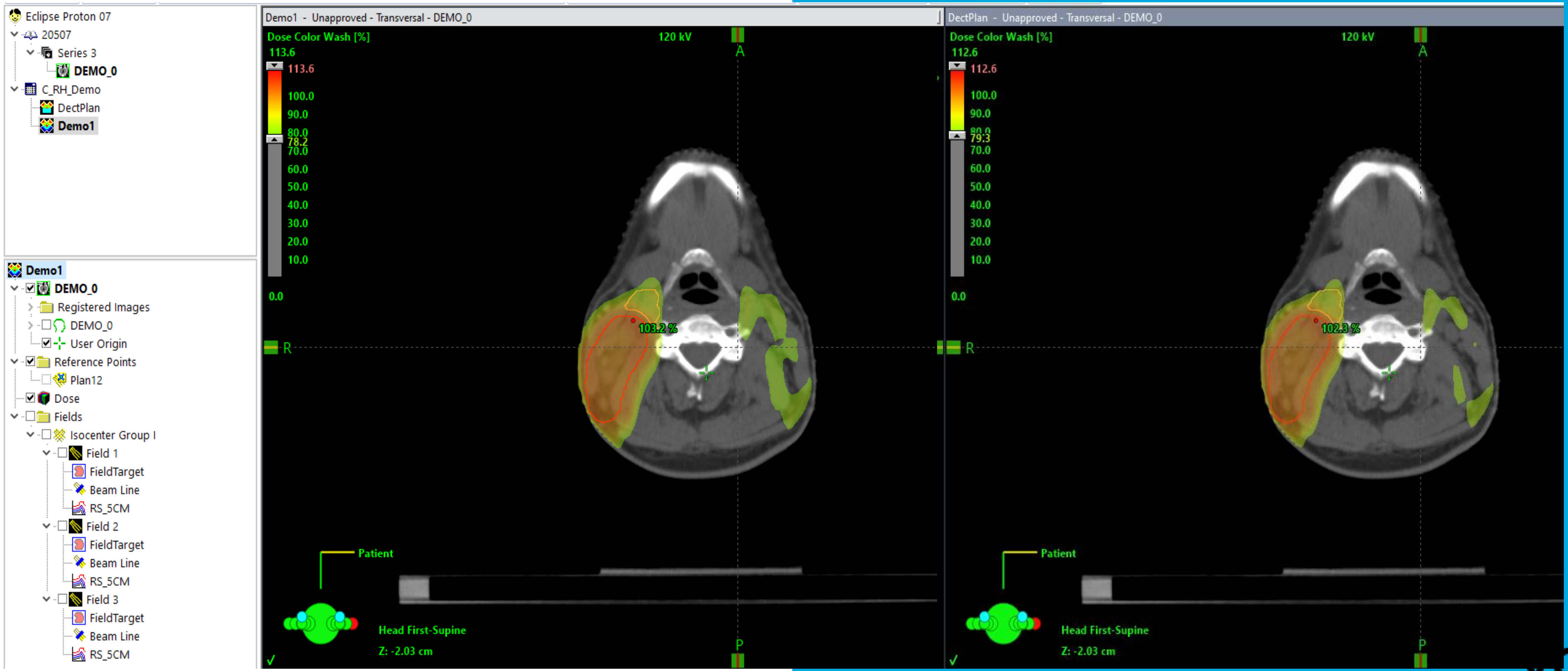
Planned Dose per Fraction [Gy]	Planned Total Dose [Gy]	Mode	Value [%]
2.000	70.000	Plan Normalization Value: 99.12%	99.1

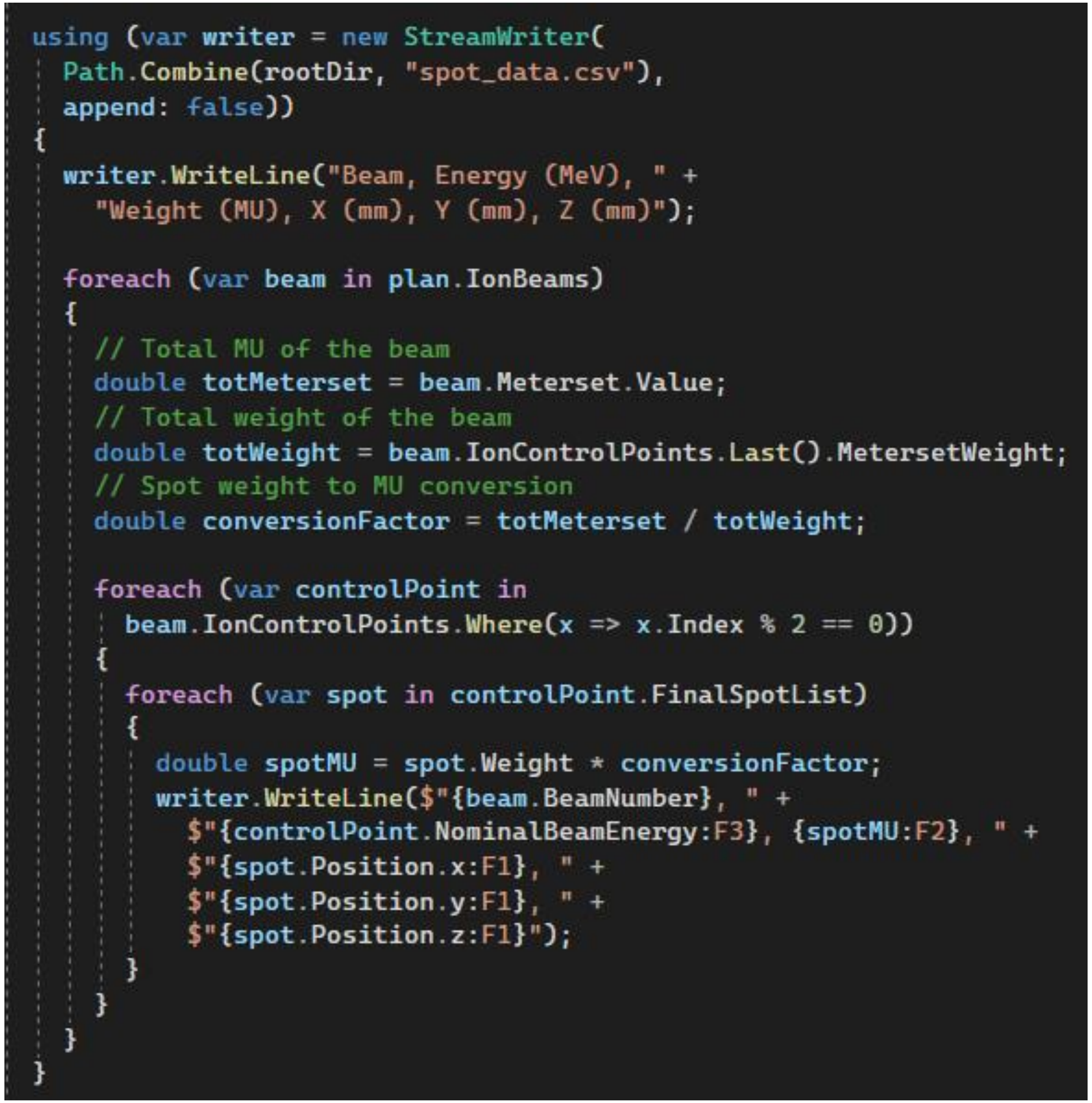
18.0

DECT Verification Plan

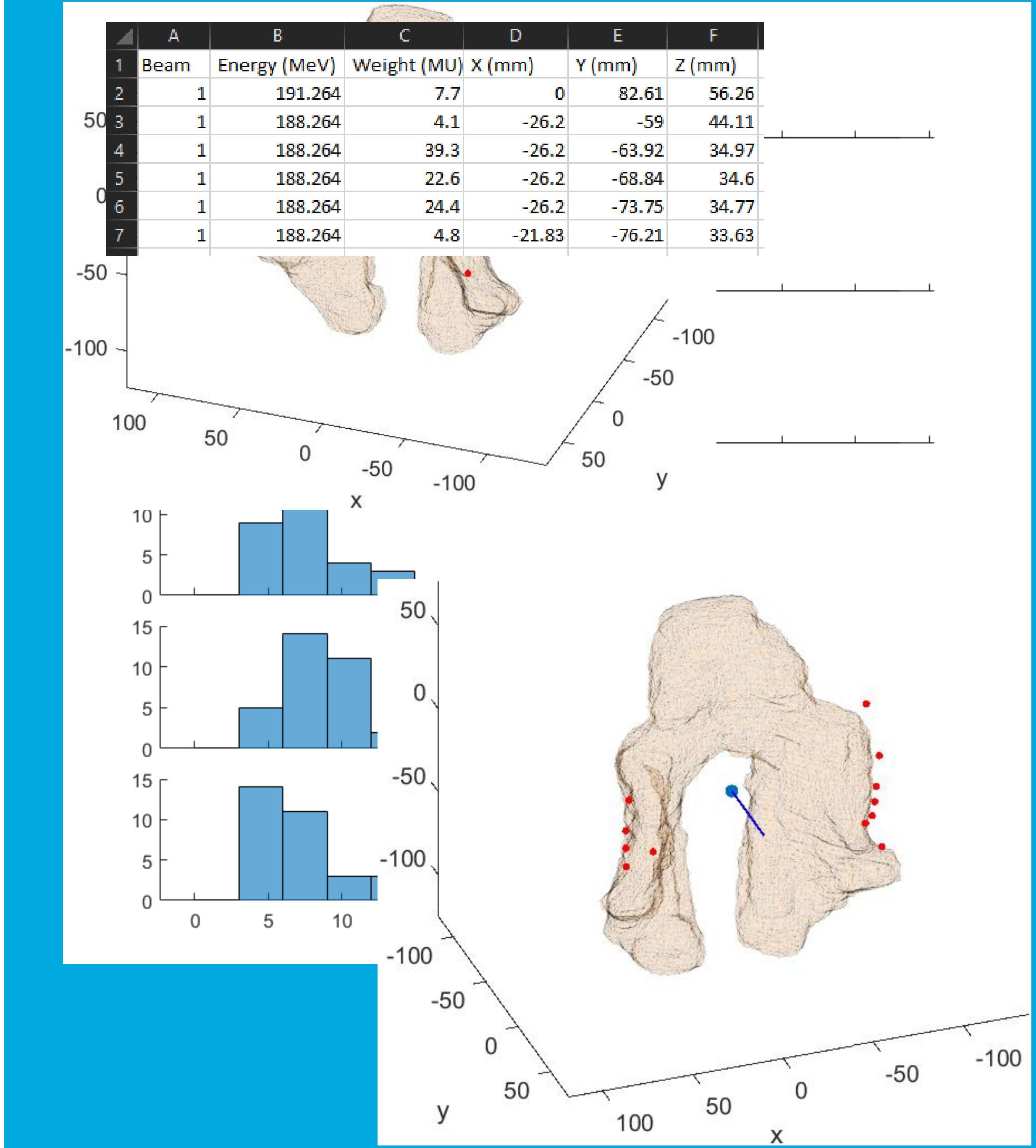


```
TP.Image rhoImage = plan.Series.Images.FirstOrDefault(
  x => x.ImageType.Contains(@"DERHOZ\RHO"));

TP.Image zeffImage = plan.Series.Images.FirstOrDefault(
  x => x.ImageType.ToLower().Contains(@"DERHOZ\Z"));

TP.IonPlanSetup dectPlan =
    plan.CreateDectVerificationPlan(rhoImage, zeffImage);
```





```
using (var writer = new StreamWriter(
  Path.Combine(rootDir, "spot_data.csv"),
  append: false))
{
  writer.WriteLine("Beam, Energy (MeV), " +
    "Weight (MU), X (mm), Y (mm), Z (mm)");

  foreach (var beam in plan.IonBeams)
  {
    // Total MU of the beam
    double totMeterset = beam.Meterset.Value;
    // Total weight of the beam
    double totWeight = beam.IonControlPoints.Last().MetersetWeight;
    // Spot weight to MU conversion
    double conversionFactor = totMeterset / totWeight;

    foreach (var controlPoint in
      beam.IonControlPoints.Where(x => x.Index % 2 == 0))
    {
      foreach (var spot in controlPoint.FinalSpotList)
      {
        double spotMU = spot.Weight * conversionFactor;
        writer.WriteLine($"{beam.BeamNumber}, " +
          $"{controlPoint.NominalBeamEnergy:F3}, {spotMU:F2}, " +
          $"{spot.Position.x:F1}, " +
          $"{spot.Position.y:F1}, " +
          $"{spot.Position.z:F1}");
      }
    }
  }
}
```



	A	B	C	D	E	F
1	Beam	Energy (MeV)	Weight (MU)	X (mm)	Y (mm)	Z (mm)
2	1	191.264	7.7	0	82.61	56.26
3	1	188.264	4.1	-26.2	-59	44.11
4	1	188.264	39.3	-26.2	-63.92	34.97
5	1	188.264	22.6	-26.2	-68.84	34.6
6	1	188.264	24.4	-26.2	-73.75	34.77
7	1	188.264	4.8	-21.83	-76.21	33.63

(Thank you)

roni.hytonen@varian.com