



PHILIPS
www.philips.com

An Overview of Scripting in Pinnacle for Workflow Customization

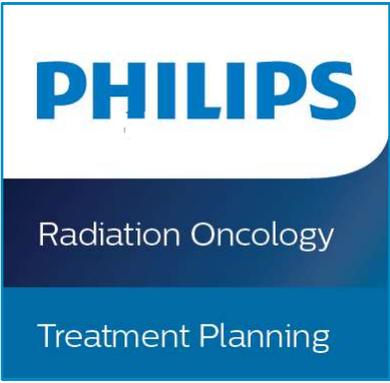
Michael Meltsner, PhD
Philips Radiation Oncology Systems
July 15, 2019

innovation  you

Conflicts of Interest



- I am a Senior Research Scientist with Philips Radiation Oncology Systems



PHILIPS
Radiation Oncology
Treatment Planning

Disclaimer



Philips provides these script files, instructions for use and all other information (written or verbal) communicated therein ("Information") "AS IS", without warranty of any kind, express or implied. Any and all use of the Information provided in this Script Portal or in reliance on the Information, including without limitation, creating, managing, supporting or using scripting in Pinnacle or otherwise, is at the sole risk and responsibility of Portal User. Portal User is solely responsible to confirm the script in their workflow. In no event will Philips be liable or responsible for any losses, injuries, demands or damages of any kind (direct, indirect, incidental, consequential or otherwise) however caused and on any theory of liability arising in any way from or relating to use or access of this portal, use of the Information by Portal User or any other person or organization, or from the creation, managing, supporting and/or use of scripts in Pinnacle or otherwise. Further, the Scripts are not intended to, and shall not be construed to, substitute or replace the Instructions for Use for Pinnacle.

3

Introduction



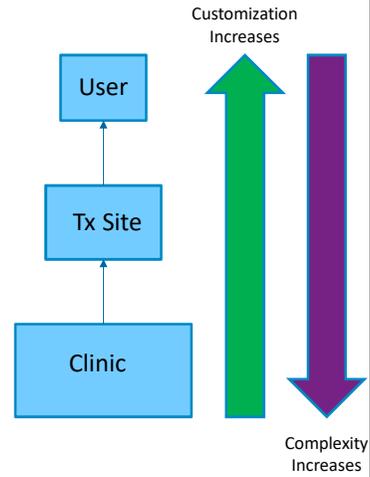
- A Pinnacle script is a small program that can mimic or replace manual interactions with the treatment planning software
- More than 90% of Pinnacle users engage in scripting to some degree
- 2/3 of Pinnacle users plan with the help of scripts regularly



Why Script?



- Every clinic has different needs
 - Specific workflows
 - Different policies for Quality Control
- Each treatment site may have unique requirements
- Every user has preferences and ways of working
- Often, tasks can be made more efficient
- Scripting can automate and facilitate much of the treatment planning process to satisfy these needs

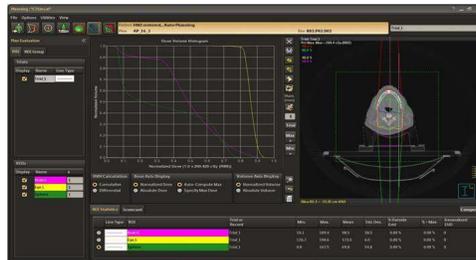


5

Types of Scripts: Customization



- Customization in treatment planning can be defined as enhancing the performance or presentation of the software
 - Basically, changing look and feel of software to user's preferences
- Examples include:
 - Defining contour line thickness
 - Changing default views of images
 - Setting isodose lines or DVHs
 - Minor repetitive tasks
- Users can change for same plan
 - Dosimetrists
 - Physicists
 - Doctors



6

Types of Scripts: Supplemental Functionality



- Even though we try, not all software can meet the current and future needs of all users at all times
- Scripting supplements the existing tools for the specific user
- Examples include:
 - Enhanced data export not found in DICOM
 - Clinic-specific metrics for plan quality
 - Additional tools to aid in modeling/planning/delivery for new linacs
 - Formatting/exporting/importing data to aid in 3rd party communication



Examples of 3rd party communication utilizing scripting (RadCalc + PlanIQ)

9

Types of Scripts: Error-Checking/QC



*The most common origins of near-miss errors were: **treatment planning (35%), patient assessment/orders (16%), equipment issues (13%), and simulation (12%).***

Novak et al. Red Journal 2014

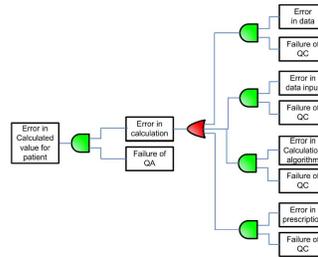
- Systematic vs. Random Errors
- Scripting can mainly reduce random errors
 - Errors of distraction
 - Overwork
 - Multitasking
 - Transcription
- Be careful of systematic errors introduced by writing a script

10

Types of Scripts: Error-Checking/QC



- Scripting can act as a highly effective Quality Control measure
- Scripts can check for incorrect:
 - Beam angles
 - Deliverability concerns
 - Units: Gy/cGy, mm/cm
 - SSD, energies, wedges, Modalities
 - Prescriptions
 - Naming conventions
- Scripts can aid:
 - Data mining and performance metrics
 - Efficiency and customization of report generation and setup
 - Support and implement error analysis tools



RADIATION ONCOLOGY®
INCIDENT LEARNING SYSTEM
Sponsored by ASTRO and AAPM

The report of Task Group 100 of the AAPM: Application of risk analysis methods to radiation therapy quality management

Types of Scripts: Error-Checking/QC Examples

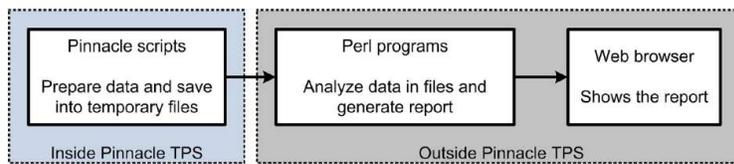
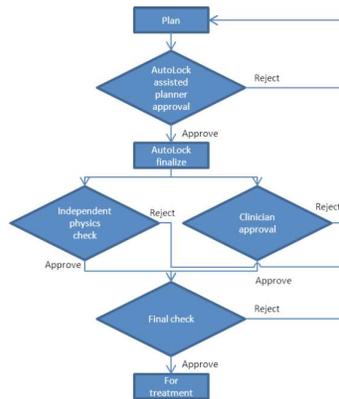


Table 1. List of IMRT plan data checked by plan integrity checking program, the asterisked items will be further covered in the following sections.

Plan data	Integrity checking
Patient information and setup	Report patient identification, position, and scan orientation to the user for verification
General plan setup	(1) Outside patient-air density threshold (2) Proper couch removal (3) Proper localization (4) Proper coordinate system (5) Presence of empty containers
Regions of interest (ROIs)	(1) Density overrides (2) Gaps between serial contour slices
Beams	(1) Completeness of basic beam information (beam name convention, field ID, etc.) (2) Consistency of beam isocenter, delivery machine name, prescription ownership (3) Beam type, energy selection (4) Jaw spacing (5) Monitor units range (6) Bolus assignment (7) MLC analysis (in-field abutments)
Prescriptions	(1) Empty prescriptions without beams (2) Invalid prescription definition
Dose calculation parameters	(1) Dose calculation grid and resolution (2) Heterogeneity corrections (3) Computation algorithm selection (4) Max dose versus prescription dose (5) Max dose points inside PTV
IMRT optimization	(1) DMPO used on all beams (2) Segments MUs > 4 (3) Segments per field (4) Total number of segments (5) Any possible conflicts in optimization objectives (see text) (6) ROI is used in objectives but is partially outside the dose grid (7) Energy ROIs used in objectives? (8) IMRT objective ranking

Types of Scripts: Error-Checking/QC Examples



FORMS document set	TP4-FORMS.1
AutoLock Plan Lock Sheet	

PLAN LOCK SHEET

Patient name:	Patient ID:
Plan name: RIGHT BREAST	Planner:
Lock status: The plan was locked by	with user name at

Acknowledged Fails

Reviewed Items

1.6	Overrides check: 1 overrides found. Outside BEAM is overridden to 0.0.	
1.8	ROIs inside dose grid: 1 ROIs are outside the dose grid: OUTSIDE_BEAM	
2.1	Prescription dose: Prescription: 1: 4000.0cGy in 15 fractions	
2.2	Prescription method: Prescription: 1 is prescribed to mean of FTV_EVAL	
2.4	Seedless: Isodose set to absolute 1110.0% LOS 0% 55.0% 90.0% 55.0% 55.0% 35.0% of 4000.0cGy	
3.1	Consistent machine and timeramp: All beams set to 0kBravo 2 in Clinical Version 8.6 with correct timeramp.	
3.4	Consistent heterogeneity: All beams set to heterogeneous with LockV parameter of density table	

Passed

1.1	Single trial named final: Trial name is FINAL	
-----	---	--

Dewhurst et al. Red Journal 2015



Pinnacle Architecture

How To Interact With Pinnacle



- Three main ways to communicate with Pinnacle:
 - Conventional Scripting
 - Custom language used in clinical versions of Pinnacle
 - Python-based Scripting
 - Currently in research mode only, WIP for clinical release
 - C++ based Plugins
 - Research mode only

15

What is a Script?



Overview

- Scripting in Pinnacle gives the user the ability to communicate to/from the Pinnacle core without having access to the code directly
- Each script is an ASCII file that contains a collection of messages that communicate sequentially with Pinnacle
- These messages allow the user to mimic interactions with mouse clicks and keyboard input
- They are analogous to macros in other software like Excel

PHILIPS

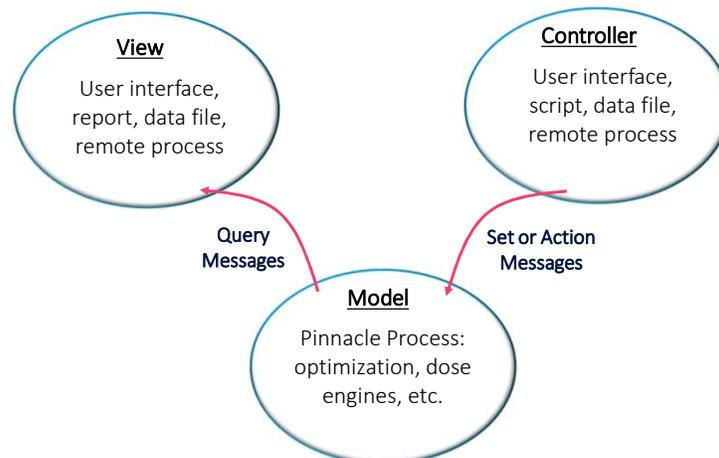
Three Basic Types of Messages



- Pinnacle responds to 3 basic types of Scripting Messages:
 - 1) **Set command:** tells Pinnacle to change an underlying attribute
 - Beam angle, prescription name
 - 2) **Query request:** asks Pinnacle to return the value of an attribute
 - Number of beams, Patient's name
 - 3) **Action command:** tells Pinnacle to perform a task
 - Add a beam, Delete an ROI

PHILIPS

Pinnacle MVC Architecture

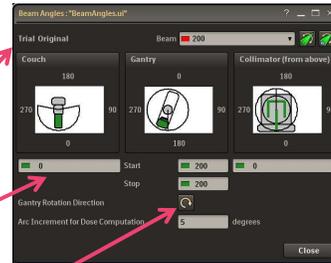


PHILIPS

How Pinnacle Widgets Work



- The Pinnacle GUI is essentially an extensive set of messages hidden behind widgets
- Each widget in the GUI has 1 or more of the three types of messages associated with it
- The label widget displays the result of a **Query** request
- The text box displays a **Query** request as its value and sends a **Set** message after the user enters a new value
- The button widget in Pinnacle typically performs an **Action**



PHILIPS

Pinnacle Internal Design



- Pinnacle is designed in an object-oriented manner
- Every object has associated attributes and functions
- The hierarchy of the Pinnacle objects is important and directly relates to the functionality of the software and hence is mimicked in the scripting language
- Objects of the same class are stored in a container simply called **ObjectList**.
 - Each object within that list may contain sub-objects including sub-lists

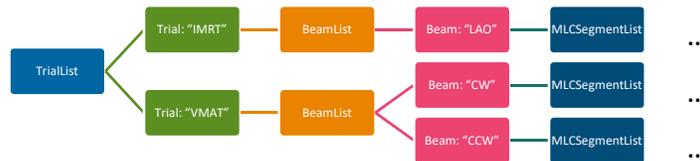


PHILIPS

Pinnacle Internal Object Hierarchy



- The hierarchy of Pinnacle (and thus Scripting) follows a logical sense of design
 - For example, each plan may contain multiple **Trials**. Each **Trial** may contain multiple **Beams**. Each **Beam** contains multiple **MLC Leaves** and so on.



- You can **Set, Query, or Act** on any object by following the hierarchical structure of the Pinnacle design starting from the root level, in this case **TrialList**.

PHILIPS

Example Messages



- Reference by name of index


```
TrialList.Current.BeamList.Lateral.Couch = 180.0;
TrialList.Current.BeamList.# "#2".Couch =
    TrialList.Current.BeamList.Lateral.Couch;
```
- Special character * loops over all objects in list


```
TrialList.Current.BeamList.# "*" .Couch = 180.0;
```
- Sort the beam list by Gantry angle in ascending order


```
TrialList.Current.BeamList.SortBy.Gantry = "";
```
- IF - THEN - ELSE numeric:


```
IF.TrialList.Count.EQUALTO.# "#1".THEN.
    TrialList.CreateChild.ELSE.TrialList.Last.MakeCurrent=1;
```

PHILIPS

Example Messages



- Run another script via message
`ExecuteNow = "/home/pinnbeta/MyScript.Script";`
- Execute command in terminal while Pinnacle continues to run
`SpawnCommandNoWait = "xterm -e OSCCommand";`
- Creating Variables
`Store.FloatAt.MyFloat = 1.23;`
`Store.At.MyFloat.Add = 23.0; //Adds 23 to MyFloat`
- There are initialization scripts that can be created to run at Pinnacle and Launchpad startup.
 - **LaunchpadInit** - all commands are run after the database is loaded
 - **PinnacleInit** - all commands in the script are run after the plan is loaded

PHILIPS

How to Learn Scripting



Learning Scripting



- There are many easy ways to find and learn scripting messages
- The easiest ways include:
 - Record and Playback
 - All actions performed in the GUI are recorded to a text file to play back
 - Writing out an object will place all the persisted object's values and messages into a text file
 - **TrialList.Current.BeamList.Current.Save = "/files/rtp/Beam.out";**
- We do offer scripting/plugin classes with documentation and support



25

InCenter Pinnacle Scripts Portal



Pinnacle³ Scripts Portal is part of InCenter

- **Scripts Portal** for PROS RightFit service agreement customers via **InCenter**.

Pinnacle³ Scripts Portal provides:

- **Scripts:** Users can view, download Pinnacle scripts created by PROS engineers
- **Resource:** centralized location to exchange scripts.
- **Ease of use:** compared to email sharing or browsing listserv
- **Security:** password protected user login
- **Sanity check:** Scripts are reviewed by PROS experts for system capability
- **Support:** Scripts downloaded from the Portal are supported by the PROS Customer Service team





Please come visit us at our booth.
Thank you!