An Overview of Scripting in Pinnacle for Workflow Customization

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

Conflicts of Interest

• I am a Senior Research Scientist with Philips Radiation Oncology Systems
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.

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

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
Types of Scripts: Automation
Standardization

- Site-specific repetitive tasks, class solutions and templates can be implemented via script

- Scripting plan creation and setup can:
  - Reduce time to create the plan
  - Create consistency in plan quality
  - Aid inexperienced users
  - Reduce errors by:
    - Ensure colors, names of beams, ROIs follow clinical conventions
    - POIs, objectives, display, etc. are consistent for reporting and review

Types of Scripts: Automation
Example

- Dual planning script automatically generates a Photon and an IMPT plan
  - Automates planning process
  - Facilitates comparison and modality choice
  - Creates backup plans for proton site
  - Aids sites without proton training to determine best Tx
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

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
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

Types of Scripts: Error-Checking/QC Examples

Types of Scripts: Error-Checking/QC

Examples

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

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
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

Pinnacle MVC Architecture
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

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
• 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.

Example Messages

• Reference by name of index
  TrialList.Current.BeamList."#2".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

• IF - THEN - ELSE numeric:
  IF.TrialList.Count.EQUALTO."#1".THEN.
    TrialList.CreateChild.ELSE.TrialList.Last.MakeCurrent=1;
Example Messages

- Run another script via message
  \texttt{ExecuteNow = "/home/pinnbeta/MyScript.Script";}

- Execute command in terminal while Pinnacle continues to run
  \texttt{SpawnCommandNoWait = "xterm \-e OSCommand";}

- Creating Variables
  \texttt{Store.FloatAt.MyFloat = 1.23;}
  \texttt{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.
  - \texttt{LaunchpadInit} - all commands are run after the database is loaded
  - \texttt{PinnacleInit} - all commands in the script are run after the plan is loaded

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

• We do offer scripting/plugin classes with documentation and support

InCenter Pinnacle Scripts Portal

**Pinnacle\(^3\) Scripts Portal** is part of InCenter

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

**Pinnacle\(^3\) 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!