

Memorial Sloan Kettering Cancer Center


Scripting Applications in Clinic – That is an button...

Chengyu Shi, Ph.D.
Memorial Sloan Kettering Cancer Center
AAPM Annual Meeting 2019

Disclosure


I owe to my colleagues for their excellent works on this talk...

Peng Zhang
Maria Chan
Jun Lin
MP Computer Services
Treatment Planning Group
Dosimetry Group
Margie Hunt
Joe Deasy




Memorial Sloan Kettering Cancer Center


This talks introduces you different types of scripting in RT




Varian Script ESAPI



Pinnacle Script Basic




MIM Workflow Design



Memorial Sloan Kettering Cancer Center

Why we need scripting?

- Part of automation
- Extend TPS function
- Customize TPS
- Prevent errors
- Improve efficiency



Memorial Sloan Kettering Cancer Center

What is Script?

A **script** or **scripting language** is a computer language with a series of **commands** within a file that is capable of being executed without being compiled. Good examples of **server-side scripting** languages include **Perl**, **PHP**, and **Python**. The best example of a client side scripting language is **JavaScript**. A full list of scripting languages and other programming languages can be found through our **programming language** definition.


Advantages of scripts

- **Open source**, allowing users to view and edit the **script** if needed.
- Does not require the file to be **compiled**, but may be when necessary.
- Easy to learn and write.
- Easy to **port** between different **operating systems**.
- Much faster to develop than an actual program - some individuals and companies write scripts as a prototype for actual programs.

Disadvantages of scripts

- **Open source**, allows others to view source code, which may be prohibited by some companies.
- Requires the user to install an **interpreter** or separate program before the script can be run.
- In some situations, they may be slower than a compiled program.

<https://www.computerhope.com/jargon/s/script.htm>



Memorial Sloan Kettering Cancer Center

What is ESAPI?

ESAPI: Eclipse Scripting Application Programming Interface, which is built into the Eclipse™ treatment planning system. C#.NET scripts, DLLs, and programs can be developed to read and operate on the patient data loaded in Eclipse™.

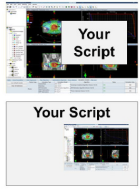
Two approaches for Eclipse Scripts

Plugin Script

- Eclipse calls you!
- Operates on current patient.


Standalone EXE

- You call Eclipse!
- Operates on any number of patients.



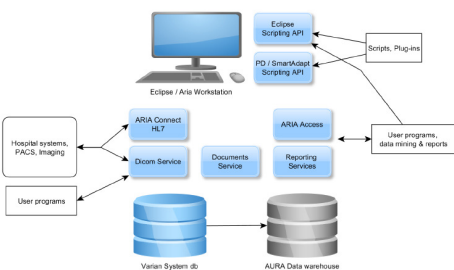
varian

Reference: Varian APIs, a handbook for programming in the Varian oncology software system
Eds. Joakim Pyry and Wayne Keranen



Memorial Sloan Kettering Cancer Center

How to exchange information with Varian database?



What do you need to start?

- Enable ESAPI Dev System: RT Administration->System Properties->Database in Research Mode
- Microsoft Visual Studio
- VMS.TPS.Common.Model.Types.dll
- VMS.TPS.Common.Model.API.dll
- What kind of script do you want?
 - Single-file plug-in
 - Binary plug-in
 - Stand-alone executable
 - Visual Scripting Action Pack
- C#.NET, Python language knowledge
- Mathematics, computer graphic basic knowledge

Hello world-Hello XXX

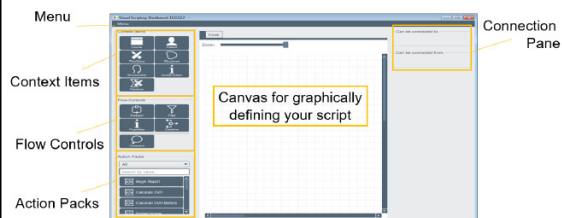
Single file plugin
Eclipse version 15.1 and higher
Using Eclipse Script Wizard
Save the following code into a file

```
public void Execute(ScriptContext context /*, System.Windows.Window window, ScriptEnvironment environment*/)
{
    // TODO : Add here the code that is called when the script is launched from Eclipse.
    MessageBox.Show("Hello" + context.CurrentUser.Name + ",\nloaded_patient_id:" + context.Patient.Name);
}
```

Open a patient in Eclipse
Run the script by Tools/Scripts



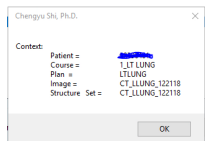
Not easy? How about Visual Scripting?



A little more complex example

```
using System;
using System.Text;
using System.Windows;
using VMS.TPS.Common.Model.API;
using VMS.TPS.Common.Model.Types;
namespace VMS.TPS
{
    public class Script
    {
        public void Execute (ScriptContext context)
        {
            Patient = context.Patient;
            Course = context.Course;
            PlanSetup plan = context.PlanSetup;
            Image image3D = context.Image;
            StructureSet structureSet = context.StructureSet;

            string msg = string.Format(
                "Context:\n",
                "patient = {0}\n",
                "course = {1}\n",
                "plan = {2}\n",
                "image = {3}\n",
                "structure Set = {4}\n",
                "(patient != null) ? patient.Id : \"not loaded \",
                (course != null) ? course.Id : \"not loaded \",
                (plan != null) ? plan.Id : \"not loaded \",
                (image3D != null) ? image3D.Id : \"not loaded \",
                (structureSet != null) ? structureSet.Id : \"not loaded \");
            MessageBox.Show (msg, "Chengyu Shi, Ph.D.");
        }
    }
}
```



I am not C#.NET or Python user, I am a MATLAB guy


Extract the data file out of the system
Import them into MATLAB environment to manipulate them
Convert them to DICOM object
Import them back to Eclipse Treatment Planning System

```
clear;
clc;
%%Accessing the TPS class
dllPath = fullfile('G:', 'My Documents\MATLAB\ESAPI', 'VMS.TPS.Common.Model.Types.dll');
ESAPI=NET.addAssembly(dllPath);

dllPath = fullfile('G:', 'My Documents\MATLAB\ESAPI', 'VMS.TPS.Common.Model.API.dll');
ESTYPE=NET.addAssembly(dllPath);
```

ESAPI can do

- Extract treatment planning data
 - Dose and image profile
 - Structure or segment profile
 - DVH
 - Automation
 - Estimation
 - Further extended calculation, such as gamma index, BED etc.




How about Pinnacle?

<SetMessage> = <Value>;

Where <Value> can be:

- A string enclosed in double quotes.
- A floating point or integer number.
- A Query message.



Examples

For example: export dose grid (A string enclosed in double quotes.)


`TrialList.Current.DoseGrid.SaveVolumeData = "home/p3rtp/dosegrid";`

Another example: set the current beam gantry to 180.0 degree (A floating point or integer number)

`TrialList.Current.BeamList.Current.Gantry=180.0;`

Set Couch angle to be equal to the gantry angle: (A Query message)

`TrialList.Current.BeamList.Current.Couch =
TrialList.Current.BeamList.Current.Gantry;`



How about MIM?

Building a workflow just needs three steps!

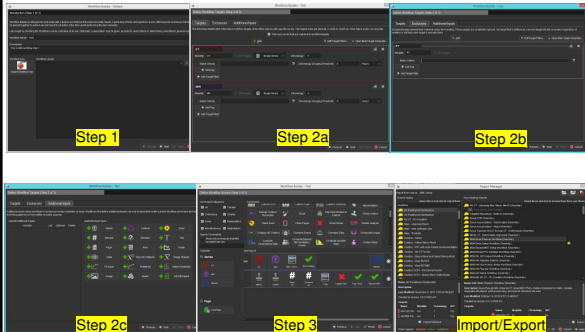
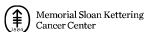
1. Give a name
2. Select targets
3. Build commands

All GUI based





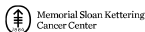
How about MIM?

Where to start to learn scripting?

Rome wasn't built in a day.
It went up very slowly, truth to tell.
The moral of the story, so they say,
If something's worth the doing,
Do it well.

- Adam Tucker



Summary

Intro Eclipse scripting basic

Intro Pinnacle scripting basic

Intro MIM scripting basic

