SCRIPTING IN RAYSTATION

Scripts can be recorded or programmed, can be executed across multiple patients, can read and write all data in RayStation. Furthermore, using the convenient interface to Microsoft .NET and Windows Presentation Foundation, scripts can, for instance, create and display windows with graphical interface components, create pdf reports and interface directly with other scriptable applications such as Microsoft Excel. The scripting language is IronPython, which is a python variety which allows access to .NET.

SCRIPTING CAPABILITIES

Automation

In the day to day clinical practice, there are often numerous check lists which must be executed manually at various stages in the process. For instance, quality related properties to check in a plan before approval. These procedures vary from clinic to clinic and cannot therefore be fully supported by the treatment planning system out-of-the-box. But through scripting, the clinic specific procedures can be automated. For instance, a script can check for properties in a plan such as small segments, disconnected target volumes, hotspots, undesirable gantry and couch angles or similarly and display a warning message or create a report.

Script controls

RayStation in itself supports automation of plan creation, structure definitions, beam placement, definitions of objectives and clinical goals through the planning protocols. Scripting can also be used to automate these steps, and it is a matter of preference which method to use. Scripting allows for more customization whereas planning protocols are easier to get started with.
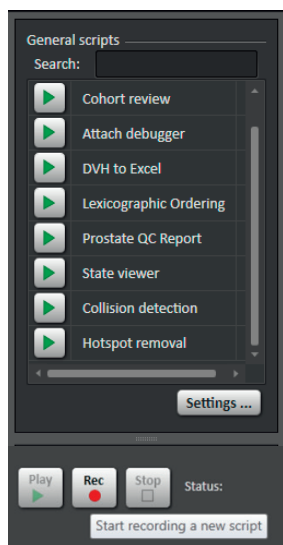
Connectivity

Many clinics have at least some non-standard software components in its ecosystem, for instance for independent MU calculation and quality assurance. Although many systems interact seamlessly though DICOM, there are often some incompatibilities which need to be resolved. Scripting provides a way to customize the interaction between RayStation and other systems for the scenarios where DICOM is not sufficient.

The script language, IronPython, is a complete programming language which also has access to all capabilities of the operating system and other applications, for instance to write files, start processes, communicate with other computers, and control other scriptable applications such as Microsoft Office.

Flexibility

Scripting can be used to create new capabilities which are not available in RayStation. For instance, automatic marker detection can be integrated. Also, scripts can be used to access options which are not available in the graphical user interface such as exporting of images of non-standard dose planes and images of all control points.

Scripting on multiple patients

Through its ability to execute over multiple patients, scripting is an ideal tool for retrospective data analysis and for evaluating new treatment techniques. For instance, a script can go through a cohort of patients and systematically alter some parameter and record the effect or just extract data such as dose statistics.

Research

The above mentioned capabilities of scripting are exceptional research tools. Automatically applying a range of parameter values in multiple plans and multiple patients, applying analysis tools on multiple patients, import and export to, for instance, Matlab, are all different ways to enable research in a more convenient and powerful way that would otherwise be possible.

In non-clinical versions of RayStation, there are additional capabilities, such as interfaces to integrate custom objective functions and external dose engines, and to export intermediate calculation results, and change non-standard options.

Cohort review
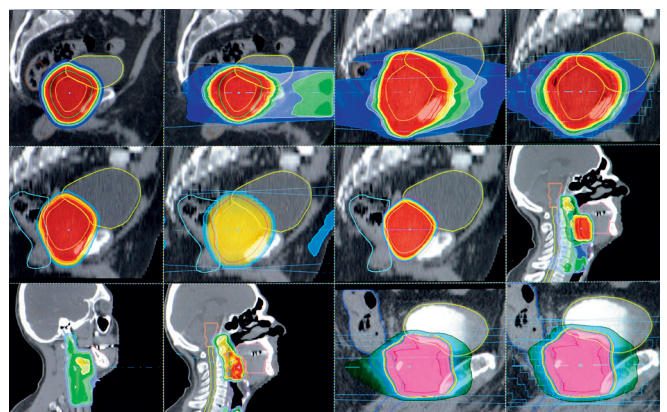
A script *has executed over several patients and created its own graphical user interface.*

WORKING WITH SCRIPTING

Recording
The easiest way to get started with scripting is to record one. Press the record button in the scripting tab and perform the actions you wish to script. When the recording is stopped, the resulting script is displayed and can be edited and saved for future use.

Debugging
It is possible to launch the scripts through a debugger available in Python Tools for Visual Studio. Script execution can be stepped, break points can be used and local variables inspected. Furthermore, the immediate window can be used to directly interact with RayStation while executing a script.

Programming
Only one initial line is required to connect to RayStation. After that, scripts can be written from scratch, or recorded scripts can be modified to, for instance, loop over multiple plans or patients, interact with the user through dialogues or perform any other task.

Exceptions
If an error occurs in the script itself, or if the script causes a non-fatal error in RayStation, an exception will be thrown within the script. The script can choose to handle the exception, or, if unhandled, the script execution will stop. Since the script executes in a different process from RayStation, unhandled exceptions will not influence RayStation.

Publishing
Once a script is finished, it can be imported into the database and marked as validated. This makes it available to all users in the clinic.

Online documentation
The data model with which the scripting interacts can be browsed through a script called State Viewer. It displays all available data properties and also all available actions along with documentation for all available parameters. The documentation can also be accessed through the interactive console or through a python editor such as Visual Studio.

Debugging in Visual Studio

State Viewer

Debugger *available in Python Tools for Visual Studio.*

State Viewer *displaying all available data properties and all available actions and parameters.*

CONCLUSION
Through scripting, the clinic specific procedures can be implemented and automated. Scripting in RayStation provides automation, connectivity and flexibility beyond what is possible in the ordinary graphical user interface.

RaySearch Laboratories