

VMAT TBI autoplanning scripts quick install and run guide

E. A. Simiele

(Revised March 2, 2022)

Current script versions

The below guide is valid and accurate for script versions:

Binary plug-in: 1.7_beta

Executable: 1.2_beta

Changes from v1.6 and v1.1

Addressed the MLC model ID issue regarding the use of multiple linacs (issue description: <https://github.com/esimiele/VMAT-TBI/issues/1>)

Binary plug-in quick install and run

1. Download the latest version of the autoplanning scripts from GitHub
2. Place the file tree in a location of your choosing
3. Open visual studio
4. Open the solution file for the plug-in script (located at:
//path/to/VMAT-TBI/ESAPI/Projects/VMATTBIautoPlan/VMATTBIautoPlan.sln)
5. Once the solution is open, right click on the project file and select properties
6. Change the name of the build (what the dll will be called) to a name of your preference (I recommend leaving it as VMATTBIautoPlan_vXX.esapi.dll where XX is the current version number on GitHub)
7. Select Build->Rebuild Solution

- This will ensure the build does not try to use any cached data from the previous build on my institution's system
 - If the code is having trouble finding the VMS.TPS dll files. You can re-add them to the code by deleting the existing references to these dll files, right clicking on references→add. Browse to the VMAT-TBI directory and there should be a folder there called 'VMS_TPS_resources' with the dll files for both v16.1 and v15.6 of ESAPI.
8. Assuming no issues, the compiled dll should now be located in //path/to/VMAT-TBI/ESAPI/plugins
 9. Inside this directory, there should also be a folder called 'configuration' and inside that folder there should be a file called VMAT_TBI.config.ini
 - This VMAT_TBI.config.ini file should be a copy of the VMAT_TBI.config.ini file in the main VMAT-TBI directory
 10. Modify the .ini file inside the configuration folder:
 - Change the documentation path location to //path/to/VMAT-TBI/documentation
 - Modify the linac names ('add linac{ }') to use linacs from your clinic (if you only need one linac, delete the second add linac entry)
 - Modify the available beam energies for VMAT TBI (we recommend 6MV and 10MV as options)
 - Modify the calculation and optimization model entries to match the model names used in your clinic
 - NOTE: IT IS IMPORTANT THAT THERE ARE NO EMPTY LINES OR SPACES BEFORE/AFTER THE '=' CHARACTER (see Section 7 of the VMAT_TBI_guide.pdf for more details)
 11. Download the example patient data from GitHub
 - There are two anonymized patient data sets available for testing: one VMAT-only case and one VMAT with AP/PA cases
 - import the CT and structure sets into a research Eclipse box (you won't be able to import the dose and RT plan files)
 12. In a research Eclipse box (i.e., test box), create/import a VMAT TBI patient CT scan and contour the lungs, kidneys, brain, bowel, lenses, and couch
 - If a research box is not available at your institution, you can use the clinical system. However, you will have to deal with script approvals which is annoying when testing scripts
 - Make sure the left and right structures (e.g., kidneys, lungs) are union-ed together into a single structure called, e.g., 'Lungs' or 'Kidneys'

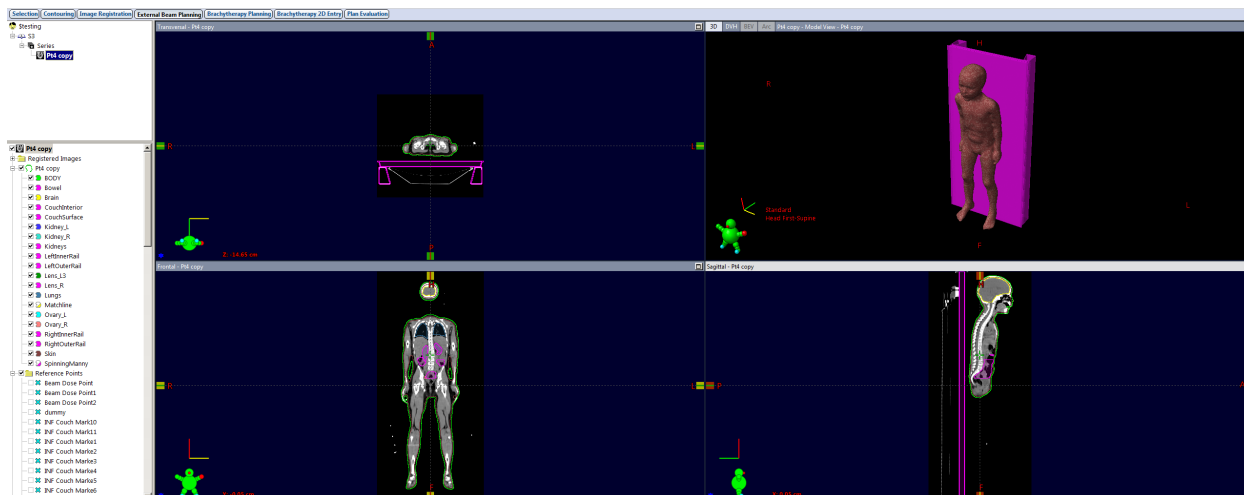


Figure 1: Initial setup for launching the VMAT TBI binary plug-in script.

- We are working to add example patient CT scans and structure sets to the GitHub repository. Stay tuned

- Open the patient structure set in external beam planning (Figure 1)
- Go to tools → scripts
- Change the default location from 'System Scripts' to 'Folder' and navigate to the plugins directory at `//path/to/VMAT-TBI/ESAPI/plugins`
- Hit ok, then the compiled dll should show up
- Run the script and ensure the GUI appears with no warnings. If so, everything initialized correctly!
- Go to the 'Script Configuration' tab and ensure the displayed parameters match the items in the .ini file
- Run a few test cases with different prescriptions to ensure the script can properly create and setup VMAT TBI plans for optimization (be sure to include both VMAT-only and VMAT with AP/PA plans)
 - TIP: binary plug-in scripts can't write to the Aria database, so you need to manually save after running the script if you want to keep the changes
 - This makes it easy to test the script since you can just reload the patient after each run of the script
- Prepare one patient dataset for optimization using one of the template treatment regimens (I recommend the myeloablative regimen)

Executable plug-in quick install and run

1. Open the executable solution file in Visual Studio
`//path/to/VMAT-TBI/ESAPI_MT/VMATTTBI_optLoopMT/VMATTTBI_optLoopMT.sln`
 - Ensure the Visual Studio you are using is linked to the research Eclipse system! If not, you will need to go into the clinical Eclipse system and approve the executable as a script that can modify the database
2. Right click on the project properties, then change the name of the build to a name of your preference (again, I recommend leaving it as the current version name in GitHub)
3. Select Build→Rebuild Solution
4. Assuming no issues, the compiled .exe should now be located in
`//path/to/VMAT-TBI/ESAPI_MT/VMATTTBI_optLoopMT/VMATTTBI_optLoopMT/bin/debug`
 - NOTE: if you are on v16.1 of Eclipse, the .exe file will be located in
`.../VMATTTBI_optLoopMT/bin/x64/debug`
5. Inside this directory, there should also be a folder called 'configuration' and inside that folder there should be a file called VMAT_TBI_config.ini. Again, this should be a copy of the .ini file in the main VMAT-TBI directory
6. Modify the .ini file inside the configuration folder:
 - Change the documentation path location to `//path/to/VMAT-TBI/documentation` (NOTE: There are two 'documentation path' entries in this file!)
 - Change the log file path to `//path/to/VMAT-TBI/log_files`
 - Set 'demo' to 'true'
 - Modify the MLC model entry to the name of the MLC model used on your clinic's linacs (i.e., the linacs that will be used for VMAT TBI)
 - NOTE: IT IS IMPORTANT THAT THERE ARE NO EMPTY LINES OR SPACES BEFORE/AFTER THE '=' CHARACTER (see Section 7 of the VMAT_TBI_guide.pdf for more details)
7. Hit the 'Start' button at the top of Visual Studio page. This will run the executable within Visual Studio
8. Ensure the GUI appears with no warnings. If so, everything initialized correctly!
9. Go to the 'Script Configuration' tab and ensure the displayed parameters match the items in the .ini file
10. Enter the MRN number of the patient you just prepared for VMAT TBI optimization and hit 'Open Patient'

11. Ensure the prescription information and optimization parameters obtained from the plan match the parameters you just set in the '_VMAT TBI' plan
12. Hit 'Confirm Constraints and Begin Optimization'
 - Since the code is in demo mode, it should only take 1-2 minutes to complete
 - Let the optimization loop fully complete. This ensures everything is working properly within the code with regards to data passing, no null references, etc.
13. Once the optimization loop completes, close the window and close the program. Nothing should have been saved to the patient plan or Aria database
14. **NOTE: THE FOLLOW STEPS WILL TAKE SEVERAL HOURS AS IT WILL ACTUALLY PERFORM AN OPTIMIZATION LOOP ON THE TEST PATIENT! PROCEED ONLY IF YOU WANT TO PERFORM THIS TEST NOW**
 - Although it will take a long time, this test is important as it ensures everything is working properly with regards to dose calculation, optimization, writing to the database, etc.
15. In the VMAT_TBI.config.ini file, set demo to 'false' and save the file
16. Hit the 'Start' button at the top of Visual Studio page and reopen the patient once the GUI pops up (ensure demo is set to false on the 'Script Configuration' tab)
17. Hit 'Confirm Constraints and Begin Optimization'
18. Check in on the optimization periodically to ensure it is still running and didn't crash or get stuck
 - If the optimization loop gets stuck, pop up windows from Eclipse are usually the culprit. Close these windows and the optimization loop should resume
 - If the program crashes, you can look at the debugging window in Visual Studio for more info and look at the log text file to see where the program crashed
19. If the program completes successfully, close the program and open Eclipse to evaluate the optimized plan
20. If you've made it this far, great! This means the scripts are working in your environment!
21. Now you should perform more thorough testing and commissioning of the scripts and modify them to your clinical needs (you should really only need to modify the .ini configuration file)