

# Viewer Interface

---

## Link with the Interface

As described in the section "Using the ReportGUI" of [Using the Checker Framework](#), it is possible to show issues in a 3D Viewer, if they have a valid position link. In general QC4OpenX doesn't provide a 3D Viewer but a C-interface to connect your own 3D Viewer to the ReportGUI. The ReportGUI loads viewer plugins in the form of shared libraries from the folder `bin/plugin`. A plugin needs to implement the interface `include/viewer/iConnector.h` and provide it via common C-API `__declspec(dllexport)` mechanism. The ReportGUI will load the library during startup and use your viewer to show issues graphically.

The C-interface declares the following functions, which are called by the ReportGUI:

```
bool StartViewer();
bool Initialize(const char* xoscPath, const char* xodrPath);
bool AddIssue(void * issueToAdd);
bool ShowIssue(void * itemToShow, void* locationToShow);
const char* GetName();
bool CloseViewer();
const char* GetLastErrorMessage();
```

This functions should be implemented inside your 3D viewer library to make your 3D viewer compatible with the Report GUI. If the shared library is not loadable or one of the functions isn't defined, the ReportGUI will show an error and the viewer won't be available.

It is not possible to implement and start multiple viewers. Starting a second viewer automatically closes the currently active one. A viewer is started via the context menu **File -> Start Viewer**.

If you start a viewer the following functions are called in this order:

1. StartViewer
2. Initialize
3. AddIssue (will be called for each issue found)

If an error occurs during the startup process, the ReportGUI will call GetLastErrorMessage to print out the error in the ReportGUI itself.

ShowIssue is triggered if you click on a RoadLocation issue in the ReportGUI. It will send the clicked issue and its location to the viewer.

If the ReportGUI is closed a currently active Viewer receives the closeViewer call.

## Viewer Example

A demo viewer is provided as source code under `examples/viewer_example`. When you open the GUI, it looks like in the image above and you can start the viewer from the File menu. Click on issues that represent a

3D error and find the function calls inside the console window. Your own viewer interface implementation can use this information to show the error in 3D.