

# File Formats

---

## Configuration File (\*.xml)

The runtime configuration settings are stored in a configuration file. This file defines which CheckerBundles and what checkers are used, how they are parameterized and whether the issues are warnings or errors. If a CheckerBundle outputs errors that are not configured in this file, the result pooling removes them later on. This way only the relevant results are included in the overall result. Furthermore, it can be configured which ReportModules are started after all CheckerBundles finish execution.

```
<Config>
<Param name="XodrFile" value="e"myTrack.xodr">
<CheckerBundle application="SemanticChecker">
  <Param name="GlobalAngleTolerance" value="0.1"/>
  <Checker checkerId="elevationChecker" minLevel="1" maxLevel="3">
    <Param name="LocalAngleTolerance" value="0.2"/>
  </Checker>
</CheckerBundle>
<ReportModule application="TextReport">
</Config>
```

The runtime parses this file, then executes the configured CheckerBundles, and hands them the configuration file. The parameter "application" to the XML tags `<CheckerBundle/>` and `<ReportModule/>` specifies the name of the executable to be used.

Notes for the paths:

- "application" should contain the path of your executable relative to the path of the ConfigGUI binary - it is recommended to place all executables in the `./bin` directory.
- Results will be stored in the directory from which the call is made.

## Result File (\*.xqar)

The results of a checker are stored persistently as XML in the form of a result file (\*.xqar). The [Base library](#) can be used to write a report. A report consists of problems (called issues) which contain syntactic or semantic flaws.

- An issue consists of the following parts:
  - Identifier: Unique number for an issue
  - Level: Indicates whether it is an information, a warning, or an error.
  - Description: Description in the form of a text
  - Optional description of where the problem lies, if applicable several grouped locations inside a **Locations** tag:
    - **FileLocation** (typically generated automatically by the ResultPooling)
      - A reference to a file with row and column

- Example: `<FileLocation column="0" fileType="1" row="223124"/>`, with `fileType="1"` for XODR, `fileType="2"` for XOSC
- **XmlLocation**
  - Addressing in a XML file with help of a XPath expression
  - Example: `<XMLLocation xpath="/OpenDRIVE/road[@id='1']"/>`
- **RoadLocation**
  - Position in road coordinates, the angles are calculated based on the road orientation
  - Example: `<RoadLocation roadId="502066" s="0.5" t="0.0"/>`
- **InertialLocation**
  - Position in inertial coordinates
  - Example: `<InertialLocation h="0.0" p="0.0" r="0.3" x="120.998703" y="0.0" z="-0.29552"/>`
- Optional external files (e. g. Images of generated graphs such as speed over distance). Currently not supported.

The following example shows the results obtained by running two CheckerBundles, one called SyntaxChecker and one SemanticChecker.

```
<CheckerResults version="1.0.0">
  <CheckerBundle name="SyntaxChecker" description="Syntax checker bundle"
summary="Found 1 incident" build_date="23.05.2019" version="1.0">
    <Param name="XodrFile" value="myTrack.xodr"/>
    <Param name="globalParameter" value="0.1"/>
    <Checker checkerId="xsdSchemaChecker" description="Checks the xsd
validity" summary="Found 1 issue">
      <Issue description="Row:8693 Column:12 invalid schema" issueId="1"
level="1">
        <Locations description="empty content is not valid for content
model '(lane+,userData*,include*)'">
          <FileLocation column="12" fileType="1" row="8693"/>
        </Locations>
      </Issue>
    </Checker>
  </CheckerBundle>
  <CheckerBundle name="SemanticChecker" description="Semantic checker bundle"
summary="Found some incidents" build_date="23.05.2019" version="1.0">
    <Param name="XodrFile" value="myTrack.xodr"/>
    <Checker checkerId="roadIdChecker" description="Checks validity of the
roadIds" summary="Found 1 issue">
      <Issue description="Road with id=5 invalid" issueId="2" level="1">
        <Locations description="Road id is defined multiple times">
          <XMLLocation xpath="/OpenDRIVE/road[@id='5']"/>
          <FileLocation column="25" fileType="1" row="505"/>
          <RoadLocation roadId="5" s="0.0" t="0.0"/>
        </Locations>
      </Issue>
    </Checker>
  </CheckerBundle>
</CheckerResults>
```

Regarding parametrization here an example of the section of the .xqar file where the default parameters are shown:

```
<CheckerResults version="1.0.0">
  <CheckerBundle description="Checks the..." name="StaticXodrChecker"
summary="173 issues..."> <!-- attribute file deleted! -->
    <Param name="GlobalAngleTolerance" value="0.2"/>
    <Param name="ElevationTolerance" value="0.1"/>
    (... more Params ...)
    <Checker checkerId="elevationChecker" description="Verify ..." summary="68
issues found...">
      <Issue description="#34: delta ..." issueId="0" level="2">
        <Locations description="Init section of scenario">
          <XMLLocation ... />
          (... more explicit location types ...)
        </Locations>
        (... more Locations ...)
      </Issue>
      (... more Issues...)
    </Checker>
    (... more Checkers...)
  </CheckerBundle>
  (... more CheckerBundles...)
</CheckerResults>
```