This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<wsdl:definitions xmlns:gloria.rt.entity.device="http://gloria.eu/rt/entity/device"
xmlns:gloria.rti.fault="http://gloria.eu/rti/fault" xmlns:rti="http://gloria.eu/rti"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="gloria_rti"
targetNamespace="http://gloria.eu/rti">
  <wsdl:types>
    <xsd:schema xmlns:gloria.rt.entity.device="http://gloria.eu/rt/entity/device"
    xmlns:gloria.rti.fault="http://gloria.eu/rti/fault"
    xmlns:rti="http://gloria.eu/rti" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://gloria.eu/rti">
      <xsd:import namespace="http://gloria.eu/rti/fault"
      schemaLocation="https://altamira.asu.cas.cz:8444/RTI/services/gloria_rtiSOAP?
      xsd=gloria_rti_fault.xsd"/>
      <xsd:import namespace="http://gloria.eu/rt/entity/device"
      schemaLocation="https://altamira.asu.cas.cz:8444/RTI/services/gloria_rtiSOAP?
      xsd=gloria_rti_device.xsd"/>
      <!--  BASIC TYPE RESPONSES - BEGIN  -->
      <xsd:element name="voidResponse">
        <xsd:complexType>
          <xsd:sequence></xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="stringResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="booleanResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:boolean"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="intResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:int"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="longResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:long"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="doubleResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:double"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
```

```
<!--  BASIC TYPE RESPONSES - END  -->
<!--  SYSTEM ELEMENTS - BEGIN  -->
<xsd:simpleType name="sessionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="TOKEN"/>
    <xsd:enumeration value="COOKIE"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="loginCert">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="user" type="xsd:string"/>
      <xsd:element name="time" type="xsd:date"/>
      <xsd:element name="sign" type="xsd:string"/>
      <xsd:element name="sessionType" type="rti:sessionType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="login">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="user" type="xsd:string"/>
      <xsd:element name="pw" type="xsd:string"/>
      <xsd:element name="sessionType" type="rti:sessionType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="logout">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  SYSTEM ELEMENTS - END  -->
<!--  COMMON DEVICE ELEMENTS - BEGIN  -->
<xsd:element name="devGetConfiguration">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDeviceProperties">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDevicePropertiesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:deviceProperty"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDeviceProperty">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
```

```
      <xsd:element name="name" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDevicePropertyResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:deviceProperty"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDevices">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="allProperties" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDevicesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:device"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devIsConnected">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDevice">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="allProperties" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devGetDeviceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:device"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devUpdateDeviceProperty">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="value"
      type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devUpdateDevicePropertyAsync">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="value"
      type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devConnect">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devDisconnect">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="devIsBlocked">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  COMMON DEVICE ELEMENTS - END  -->
<!--  CAMERA ELEMENTS - BEGIN  -->
<xsd:element name="camImageReady">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCameraType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCameraTypeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:cameraType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetXsize">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="camGetYSize">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camCanAbortExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camCanAsymetricBin">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camCanGetCoolerPower">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camCanSetCCDTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camCanControlTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camCanStopExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCoolerPower">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCoolerPowerResponse">
```

```
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetElectronsPerADU">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetFullWellCapacity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasShutter">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasBrightness">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasConstrast">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasGain">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasGamma">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasSubframe">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHasExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camHeatSinkTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camIsPulseGuiding">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetLastError">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetLastExposureDuration">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetLastExposureStart">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetLastExposureStartResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:date"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetMaxAdu">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetMaxBinX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetMaxBinY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetPixelSizeX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetPixelSizeY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetAcquisitionMode">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetAcquisitionModeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out"
      type="gloria.rt.entity.device:cameraAcquisitionMode"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetFPS">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetFPSResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="camGetDigitilizingMode">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetDigitilizingModeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out"
      type="gloria.rt.entity.device:cameraDigitizingMode"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetBinX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetBinX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetBinY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetBinY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camIsCoolerOn">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetCoolerOn">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:boolean"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetROINumX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetROINumX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetROINumY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetROINumY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetROIStartX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetROIStartX">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetROIStartY">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetROIStartY">
  <xsd:complexType>
```

```
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetBrightness">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetBrightness">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetContrast">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetContrast">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetGain">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetGain">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetGamma">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="camGetGamma">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetObjectExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="object" type="xsd:string"/>
      <xsd:element name="filterType" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetObjectExposureTimeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetCCDTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCCDTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCCDTemperatureResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:float"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCCDCurrentTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetCCDCurrentTemperatureResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camAbortExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camPulseGuide">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="direction" type="xsd:int"/>
      <xsd:element name="duration" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camStartExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="light" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camStopExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetImage">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="format" type="gloria.rt.entity.device:imageFormat"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetImageResponse">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="out" type="gloria.rt.entity.device:Image"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetOneShotModeImageFormats">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetOneShotModeImageFormatsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:imageFormat"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetContinueModeImageFormats">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetContinueModeImageFormatsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:imageFormat"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetImageDataType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetImageDataTypeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:imageContentType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetBitDepth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetBitDepth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="bits" type="xsd:int"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetContinueModeQuality">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetContinueModeQuality">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetOneShotModeQuality">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetOneShotModeQuality">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetContinueModeImagePath">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetOneShotModeImagePath">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetAutoGain">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetAutoGain">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
```

```
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetAutoExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camSetAutoExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camStartContinueMode">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camStopContinueMode">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetImageURL">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="uid" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetImageURLProperFormat">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="uid" type="xsd:string"/>
      <xsd:element name="format" type="gloria.rt.entity.device:imageFormat"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetFilterWheels">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="camGetFilterWheelsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="camGetFocuser">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  CAMERA ELEMENTS - END  -->
<!--  DOME ELEMENTS - BEGIN  -->
<xsd:element name="domGetNumberElement">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domCanSetAltitude">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domCanSetAzimuth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domCanSetPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domIsAtHome">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domIsAtPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="domGetAltitude">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domGetAzimuth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domOpen">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="element" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domClose">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="element" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domGoHome">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domSetPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="altitude" type="xsd:double"/>
      <xsd:element name="azimuth" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domMoveAzimuth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
```

```
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="azimuth" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domMoveAltitude">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="altitude" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domSetTracking">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domGetTracking">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="domSlewObject">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="object" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  DOME ELEMENTS - END  -->
<!--  MOUNT ELEMENTS - BEGIN  -->
<xsd:element name="mntGetUtcClock">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetUtcClockResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:date"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSetUtcClock">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="date" type="xsd:date"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="mntGetSiderealDate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntIsAtHome">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntIsParked">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetTargetRightAscension">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetTargetDeclination">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetTrackingDeclinationRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetTrackingAscensionRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetTrackingRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="mntGetTrackingRateResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:trackingRateType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSetTrackingRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="rate" type="gloria.rt.entity.device:trackingRateType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetTracking">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSetTracking">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetGuideRateDeclination">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetDeclinationRateRightAscension">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntIsSlewing">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetPosAxis2">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="mntGetPosAxis3">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetPosAxis1">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanPulseGuide">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSetGuideRates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSetPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSetPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="ascension" type="xsd:double"/>
      <xsd:element name="declination" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetALTParkPos">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetAZParkPos">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="mntCanSetTracking">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSetTrackingRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSlewCoordinates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSlewCoordinatesAsync">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSlewObject">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSlewAltAz">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanSlewAzAsync">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntCanMoveAzis">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="mntAxisRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntAxisRateResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:axisRateType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntTrackingRates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntTrackingRatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:trackingRateType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGoHome">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntPark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntUnpark">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSlewToAltAz">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="azimuth" type="xsd:double"/>
      <xsd:element name="altitude" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="mntSlewToAltAzAsync">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="azimuth" type="xsd:double"/>
      <xsd:element name="altitude" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSlewToCoordinates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="ascension" type="xsd:double"/>
      <xsd:element name="declination" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSlewToCoordinatesAsync">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="ascension" type="xsd:double"/>
      <xsd:element name="declination" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntMoveAxis">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="axisType" type="xsd:int"/>
      <xsd:element name="rate" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntPulseGuide">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="guideDirection" type="xsd:int"/>
      <xsd:element name="duration" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntStopSlewAxis">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="axisType" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntStopSlew">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetPointingModel">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetPointingModelResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:mountPointingModel"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntMoveNorth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntMoveSouth">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntMoveEast">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntMoveWest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSetSlewRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="rate" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntGetSlewRate">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
```

```
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntSlewObject">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="object" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntIsPointingAtObject">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="object" type="xsd:string"/>
      <xsd:element name="raError" type="xsd:double"/>
      <xsd:element name="decError" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="mntIsPointingAtCoordinates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="ra" type="xsd:double"/>
      <xsd:element name="dec" type="xsd:double"/>
      <xsd:element name="raError" type="xsd:double"/>
      <xsd:element name="decError" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  MOUNT ELEMENTS - END  -->
<!--  FOCUSER ELEMENTS - BEGIN  -->
<xsd:element name="focGetCamera">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focIsAbsolute">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focGetStepSize">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focGetMaxIncrement">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focGetMaxStep">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focGetMinStep">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focGetPosition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focIsTempCompAvailable">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focGetTemperature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focSetTempComp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="trackingMode" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focHalt">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="focMove">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
```

```
      <xsd:element name="position" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  MOUNT ELEMENTS - END  -->
<!--  DEW REMOVER ELEMENTS - BEGIN  -->
<xsd:element name="dwrGetChannelsNumber">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dwrGetChannelType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="channel" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dwrGetChannelTypeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out"
      type="gloria.rt.entity.device:dewRemoverChannelType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dwrSetTemperatureThreshold">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="channel" type="xsd:int"/>
      <xsd:element name="temperature" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="dwrSetCycleThreshold">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="channel" type="xsd:int"/>
      <xsd:element name="cycle" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  DEW REMOVER ELEMENTS - END  -->
<!--  PHOTOMETER ELEMENTS - BEGIN  -->
<xsd:element name="fhtGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fhtGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fhtGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fhtSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fhtGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fhtGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  PHOTOMETER ELEMENTS - END  -->
<!--  RAIN DETECTOR ELEMENTS - BEGIN  -->
<xsd:element name="rndGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rndGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rndGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rndSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rndGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rndGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rndIsRaining">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  RAIN DETECTOR ELEMENTS - END  -->
<!--  RHSENSOR ELEMENTS - BEGIN  -->
<xsd:element name="rhsGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rhsGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rhsGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rhsSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="rhsGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rhsGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  RHSENSOR ELEMENTS - END  -->
<!--  BAROMETER ELEMENTS - BEGIN  -->
<xsd:element name="barGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="barGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="barGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="barSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="barGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="barGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  BAROMETER ELEMENTS - END  -->
<!--  TEMPERATURE SENSOR ELEMENTS - BEGIN  -->
<xsd:element name="tempGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tempGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tempGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tempSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tempGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="tempGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  TEMPERATURE SENSOR - END  -->
<!--  STORM SENSOR ELEMENTS - BEGIN  -->
<xsd:element name="strIsAvailableOrientation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetDistanceMeasureUnit">
```

```
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetDistanceMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetDistanceMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetDegrees">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetAbosoluteDegrees">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strSetDistanceMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalLong"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetDistanceMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetDistanceMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalLong"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strSetOrientationMeasureStates">
```

```
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetOrientationMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="strGetOrientationMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  STORM SENSOR ELEMENTS - END  -->
<!--  WEATHER VANE ELEMENTS - BEGIN  -->
<xsd:element name="wvnGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wvnGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wvnGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wvnGetAbosluteDegrees">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wvnSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wvnGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wvnGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  WEATHER VANE ELEMENTS - END  -->
<!--  WIND SPEED ELEMENTS - BEGINS  -->
<xsd:element name="wspGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wspGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wspGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wspSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wspGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="wspGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  WIND SPEED ELEMENTS - END  -->
<!--  CLOUD DETECTOR ELEMENTS - BEGINS  -->
<xsd:element name="cldGetMeasureUnit">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="cldGetMeasureUnitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:measureUnit"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="cldGetMeasure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="cldSetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="states"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="cldGetMeasureStates">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="cldGetMeasureStatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="gloria.rt.entity.device:sensorStateIntervalDouble"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  CLOUD DETECTOR ELEMENTS - END  -->
<!--  FILTER WHEEL ELEMENTS - BEGIN  -->
<xsd:element name="fwGetCamera">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="fwGetFilterList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGetFilterListResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="out"
      type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGetPositionNumber">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGetSpeedSwitching">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGetFilterSize">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGetFilterSizeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGetFilterKind">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwIsAtHome">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwSetOffset">
```

```
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="positions"
      type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwSelectFilterKind">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="kind" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwSelectFilterPosition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="position" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="fwGoHome">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  FILTER WHEEL ELEMENTS - END  -->
<!--  ROTATOR ELEMENTS - BEGIN  -->
<xsd:element name="rttGetCurrentPosition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="rttSetTargetPosition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="position" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  ROTATOR ELEMENTS - END  -->
<!--  SURVEILLANCE CAMERA ELEMENTS - BEGIN  -->
<xsd:element name="scamHasBrightness">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamHasContrast">
```

```
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamAcquisitionMode">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamAcquisitionModeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out"
      type="gloria.rt.entity.device:cameraAcquisitionMode"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetFPS">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetFPSResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetDigitizingMode">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetDigitizingModeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out"
      type="gloria.rt.entity.device:cameraDigitizingMode"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamIsPTSupported">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamIsZoomSupported">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetPanMin">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetPanMax">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetTiltMin">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetTiltMax">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetZoomMax">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamSetExposureTime">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamIsImageReady">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetVideoStreamingURL">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetImageURL">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamSetBrightness">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetBrightness">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamSetContrast">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetContrast">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetPanRotation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamSetPanRotation">
  <xsd:complexType>
```

```
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetTiltRotation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamSetTiltRotation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetZoom">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamSetZoom">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="value" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamAbortExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamStartExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
      <xsd:element name="light" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamStopExposure">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="scamGetImage">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="deviceId" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="scamGetImageResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:Image"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  SURVEILLANCE CAMERA ELEMENTS - END  -->
<!--  EXECUTOR ELEMENTS - BEGIN  -->
<xsd:element name="execStartOp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
      <xsd:element name="uuidOp" type="xsd:string"/>
      <xsd:element name="user" type="xsd:string"/>
      <xsd:element name="seconds" type="xsd:long"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="execStopOp">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="execGetInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="execGetInfoResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:executorInfo"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="execGetLastOpInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="session" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="execGetLastOpInfoResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="out" type="gloria.rt.entity.device:opInfo"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<!--  EXECUTOR ELEMENTS - END  -->
</xsd:schema>
```

```
</wsdl:types>
<wsdl:message name="mntSlewToCoordinatesAsyncResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devConnectRequest">
  <wsdl:part element="rti:devConnect" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barGetMeasureStatesRequest">
  <wsdl:part element="rti:barGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveWestRequest">
  <wsdl:part element="rti:mntMoveWest" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetTiltMaxResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamIsZoomSupportedResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetAltitudeResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewCoordinatesAsyncResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devUpdateDevicePropertyAsyncResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFocuserResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempSetMeasureStatesRequest">
  <wsdl:part element="rti:tempSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetSpeedSwitchingResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageDataTypeRequest">
  <wsdl:part element="rti:camGetImageDataType" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetAbosluteDegreesRequest">
  <wsdl:part element="rti:wvnGetAbosluteDegrees" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDegreesResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camPulseGuideRequest">
  <wsdl:part element="rti:camPulseGuide" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROINumYRequest">
  <wsdl:part element="rti:camSetROINumY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBinYResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wspSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBinXResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domSlewObjectResponse">
```

```
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focSetTempCompRequest">
  <wsdl:part element="rti:focSetTempComp" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtGetMeasureUnitResponse">
  <wsdl:part element="rti:fhtGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetAutoExposureTimeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetContrastResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetPanMaxResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="loginCertResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetPanMinRequest">
  <wsdl:part element="rti:scamGetPanMin" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focMoveRequest">
  <wsdl:part element="rti:focMove" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROINumYResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntPulseGuideResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strSetOrientationMeasureStatesRequest">
  <wsdl:part element="rti:strSetOrientationMeasureStates" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="camGetYSizeResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrGetChannelsNumberRequest">
  <wsdl:part element="rti:dwrGetChannelsNumber" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetGuideRateDeclinationRequest">
  <wsdl:part element="rti:mntGetGuideRateDeclination" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camPulseGuideResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetGammaResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsPointingAtObjectRequest">
  <wsdl:part element="rti:mntIsPointingAtObject" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focIsAbsoluteRequest">
  <wsdl:part element="rti:focIsAbsolute" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domMoveAltitudeRequest">
  <wsdl:part element="rti:domMoveAltitude" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntAxisRateRequest">
  <wsdl:part element="rti:mntAxisRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsGetMeasureRequest">
  <wsdl:part element="rti:rhsGetMeasure" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="devUpdateDevicePropertyRequest">
  <wsdl:part element="rti:devUpdateDeviceProperty" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetMaxAduRequest">
  <wsdl:part element="rti:camGetMaxAdu" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCanSetAltitudeRequest">
  <wsdl:part element="rti:domCanSetAltitude" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingRateRequest">
  <wsdl:part element="rti:mntGetTrackingRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetVideoStreamingURLResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPosAxis2Response">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focHaltResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBrightnessRequest">
  <wsdl:part element="rti:camGetBrightness" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntStopSlewAxisResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetBrightnessRequest">
  <wsdl:part element="rti:scamSetBrightness" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROIStartYRequest">
  <wsdl:part element="rti:camSetROIStartY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetDigitizingModeRequest">
  <wsdl:part element="rti:scamGetDigitizingMode" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBitDepthRequest">
  <wsdl:part element="rti:camSetBitDepth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetSlewRateResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContinueModeQualityResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetElectronsPerADURequest">
  <wsdl:part element="rti:camGetElectronsPerADU" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetExposureTimeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devIsConnectedResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContinueModeImagePathResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execGetLastOpInfoRequest">
  <wsdl:part element="rti:execGetLastOpInfo" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetObjectExposureTimeResponse">
```

```
  <wsdl:part element="rti:camGetObjectExposureTimeResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="fwGoHomeRequest">
  <wsdl:part element="rti:fwGoHome" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGoHomeRequest">
  <wsdl:part element="rti:domGoHome" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetAltitudeRequest">
  <wsdl:part element="rti:domGetAltitude" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetGainRequest">
  <wsdl:part element="rti:camSetGain" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="logoutResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFullWellCapacityRequest">
  <wsdl:part element="rti:camGetFullWellCapacity" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveSouthResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamAcquisitionModeRequest">
  <wsdl:part element="rti:scamAcquisitionMode" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetZoomMaxRequest">
  <wsdl:part element="rti:scamGetZoomMax" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetFilterKindRequest">
  <wsdl:part element="rti:fwGetFilterKind" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempGetMeasureUnitRequest">
  <wsdl:part element="rti:tempGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetCameraResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domSetTrackingResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBinYResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanAbortExposureRequest">
  <wsdl:part element="rti:camCanAbortExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasGammaRequest">
  <wsdl:part element="rti:camHasGamma" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camAbortExposureResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetConfigurationResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetTiltRotationRequest">
  <wsdl:part element="rti:scamGetTiltRotation" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetMaxBinYRequest">
  <wsdl:part element="rti:camGetMaxBinY" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="camSetAutoGainRequest">
  <wsdl:part element="rti:camSetAutoGain" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCoolerPowerResponse">
  <wsdl:part element="rti:camGetCoolerPowerResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveNorthResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devConnectResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanAsymetricBinResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewCoordinatesRequest">
  <wsdl:part element="rti:mntCanSlewCoordinates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveEastResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetGainRequest">
  <wsdl:part element="rti:camGetGain" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGoHomeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldGetMeasureStatesResponse">
  <wsdl:part element="rti:cldGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetCCDTemperatureResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsPointingAtCoordinatesResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetOneShotModeQualityRequest">
  <wsdl:part element="rti:camSetOneShotModeQuality" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetNumberElementResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsGetMeasureStatesResponse">
  <wsdl:part element="rti:rhsGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetTiltRotationResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetCameraRequest">
  <wsdl:part element="rti:fwGetCamera" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStartExposureRequest">
  <wsdl:part element="rti:camStartExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetDigitilizingModeRequest">
  <wsdl:part element="rti:camGetDigitilizingMode" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetTrackingRequest">
  <wsdl:part element="rti:domGetTracking" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveAxisResponse">
```

```
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsAtHomeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetAutoExposureTimeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToAltAzAsyncResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamStopExposureRequest">
  <wsdl:part element="rti:scamStopExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetAzimuthResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetALTParkPosRequest">
  <wsdl:part element="rti:mntGetALTParkPos" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveSouthRequest">
  <wsdl:part element="rti:mntMoveSouth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBrightnessResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROINumYRequest">
  <wsdl:part element="rti:camGetROINumY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCCDTemperatureResponse">
  <wsdl:part element="rti:camGetCCDTemperatureResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="focGetPositionResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBinYRequest">
  <wsdl:part element="rti:camGetBinY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetSpeedSwitchingRequest">
  <wsdl:part element="rti:fwGetSpeedSwitching" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasBrightnessResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domIsAtParkResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPointingModelResponse">
  <wsdl:part element="rti:mntGetPointingModelResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageURLResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetStepSizeResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetMaxIncrementRequest">
  <wsdl:part element="rti:focGetMaxIncrement" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROIStartXResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetTiltRotationRequest">
  <wsdl:part element="rti:scamSetTiltRotation" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="camSetCCDTemperatureRequest">
  <wsdl:part element="rti:camSetCCDTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rttGetCurrentPositionResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrSetCycleThresholdResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtGetMeasureStatesResponse">
  <wsdl:part element="rti:fhtGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamAcquisitionModeResponse">
  <wsdl:part element="rti:scamAcquisitionModeResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStartContinueModeResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetTrackingRateResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetBrightnessResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntUnparkRequest">
  <wsdl:part element="rti:mntUnpark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPosAxis1Request">
  <wsdl:part element="rti:mntGetPosAxis1" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetImageURLResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetGammaRequest">
  <wsdl:part element="rti:camSetGamma" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetExposureTimeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsPointingAtObjectResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetParkRequest">
  <wsdl:part element="rti:mntCanSetPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetAcquisitionModeResponse">
  <wsdl:part element="rti:camGetAcquisitionModeResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="domCanSetAzimuthResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwSelectFilterPositionRequest">
  <wsdl:part element="rti:fwSelectFilterPosition" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewAltAzRequest">
  <wsdl:part element="rti:mntCanSlewAltAz" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetYSizeRequest">
  <wsdl:part element="rti:camGetYSize" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasExposureTimeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="camGetLastExposureDurationRequest">
  <wsdl:part element="rti:camGetLastExposureDuration" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempGetMeasureRequest">
  <wsdl:part element="rti:tempGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrGetChannelTypeRequest">
  <wsdl:part element="rti:dwrGetChannelType" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsSetMeasureStatesRequest">
  <wsdl:part element="rti:rhsSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndGetMeasureStatesResponse">
  <wsdl:part element="rti:rndGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetFPSRequest">
  <wsdl:part element="rti:scamGetFPS" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrSetTemperatureThresholdRequest">
  <wsdl:part element="rti:dwrSetTemperatureThreshold" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBrightnessResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetOneShotModeQualityRequest">
  <wsdl:part element="rti:camGetOneShotModeQuality" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devIsConnectedRequest">
  <wsdl:part element="rti:devIsConnected" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetCameraRequest">
  <wsdl:part element="rti:focGetCamera" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFilterWheelsRequest">
  <wsdl:part element="rti:camGetFilterWheels" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToCoordinatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStartExposureResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrSetCycleThresholdRequest">
  <wsdl:part element="rti:dwrSetCycleThreshold" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strIsAvailableOrientationResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBrightnessRequest">
  <wsdl:part element="rti:camSetBrightness" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageURLProperFormatResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetSiderealDateResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetFPSResponse">
  <wsdl:part element="rti:scamGetFPSResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetSlewRateResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBitDepthResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="camCanStopExposureResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetSiderealDateRequest">
  <wsdl:part element="rti:mntGetSiderealDate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wspGetMeasureStatesRequest">
  <wsdl:part element="rti:wspGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barGetMeasureUnitRequest">
  <wsdl:part element="rti:barGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDevicePropertyRequest">
  <wsdl:part element="rti:devGetDeviceProperty" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntStopSlewResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetUtcClockResponse">
  <wsdl:part element="rti:mntGetUtcClockResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetTrackingRateRequest">
  <wsdl:part element="rti:mntCanSetTrackingRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBitDepthRequest">
  <wsdl:part element="rti:camGetBitDepth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamIsImageReadyRequest">
  <wsdl:part element="rti:scamIsImageReady" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTargetDeclinationRequest">
  <wsdl:part element="rti:mntGetTargetDeclination" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetFilterListRequest">
  <wsdl:part element="rti:fwGetFilterList" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROINumXRequest">
  <wsdl:part element="rti:camSetROINumX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldGetMeasureRequest">
  <wsdl:part element="rti:cldGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBinXRequest">
  <wsdl:part element="rti:camSetBinX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntTrackingRatesResponse">
  <wsdl:part element="rti:mntTrackingRatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanGetCoolerPowerResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetPositionNumberRequest">
  <wsdl:part element="rti:fwGetPositionNumber" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domOpenResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetBrightnessResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDevicePropertiesRequest">
  <wsdl:part element="rti:devGetDeviceProperties" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetFilterSizeRequest">
```

```
  <wsdl:part element="rti:fwGetFilterSize" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamStartExposureRequest">
  <wsdl:part element="rti:scamStartExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewAzAsyncRequest">
  <wsdl:part element="rti:mntCanSlewAzAsync" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetExposureTimeRequest">
  <wsdl:part element="rti:camSetExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetDigitizingModeResponse">
  <wsdl:part element="rti:scamGetDigitizingModeResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDistanceMeasureStatesRequest">
  <wsdl:part element="rti:strGetDistanceMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStopExposureRequest">
  <wsdl:part element="rti:camStopExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="loginResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetGuideRatesRequest">
  <wsdl:part element="rti:mntCanSetGuideRates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetPanRotationResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domIsAtParkRequest">
  <wsdl:part element="rti:domIsAtPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetMaxBinXResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetMaxStepResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domParkRequest">
  <wsdl:part element="rti:domPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domSlewObjectRequest">
  <wsdl:part element="rti:domSlewObject" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanSetCCDTemperatureRequest">
  <wsdl:part element="rti:camCanSetCCDTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamIsImageReadyResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROIStartYResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rtiError">
  <wsdl:part element="gloria.rti.fault:errorDetail" name="rtiError"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDevicePropertiesResponse">
  <wsdl:part element="rti:devGetDevicePropertiesResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingRateResponse">
```

```
  <wsdl:part element="rti:mntGetTrackingRateResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanSetCCDTemperatureResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devIsBlockedRequest">
  <wsdl:part element="rti:devIsBlocked" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntUnparkResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwSelectFilterPositionResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barGetMeasureStatesResponse">
  <wsdl:part element="rti:barGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageRequest">
  <wsdl:part element="rti:camGetImage" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetPixelSizeXRequest">
  <wsdl:part element="rti:camGetPixelSizeX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetMeasureRequest">
  <wsdl:part element="rti:wvnGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetAzimuthRequest">
  <wsdl:part element="rti:domGetAzimuth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetImageURLRequest">
  <wsdl:part element="rti:scamGetImageURL" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetMeasureUnitResponse">
  <wsdl:part element="rti:wvnGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToAltAzAsyncRequest">
  <wsdl:part element="rti:mntSlewToAltAzAsync" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wspGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="loginRequest">
  <wsdl:part element="rti:login" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempGetMeasureStatesResponse">
  <wsdl:part element="rti:tempGetMeasureStatesResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="camSetCoolerOnResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetDeclinationRateRightAscensionRequest">
  <wsdl:part element="rti:mntGetDeclinationRateRightAscension" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="strGetAbosoluteDegreesResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetTemperatureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domSetParkRequest">
```

```
  <wsdl:part element="rti:domSetPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execGetLastOpInfoResponse">
  <wsdl:part element="rti:execGetLastOpInfoResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetImageResponse">
  <wsdl:part element="rti:scamGetImageResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strSetDistanceMeasureStatesRequest">
  <wsdl:part element="rti:strSetDistanceMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPosAxis3Request">
  <wsdl:part element="rti:mntGetPosAxis3" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execGetInfoRequest">
  <wsdl:part element="rti:execGetInfo" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanControlTemperatureResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetMaxBinYResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROINumYResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveAxisRequest">
  <wsdl:part element="rti:mntMoveAxis" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domOpenRequest">
  <wsdl:part element="rti:domOpen" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingAscensionRateRequest">
  <wsdl:part element="rti:mntGetTrackingAscensionRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetPixelSizeYRequest">
  <wsdl:part element="rti:camGetPixelSizeY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContinueModeImageFormatsResponse">
  <wsdl:part element="rti:camGetContinueModeImageFormatsResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewObjectResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntParkRequest">
  <wsdl:part element="rti:mntPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrSetTemperatureThresholdResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingDeclinationRateResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domSetTrackingRequest">
  <wsdl:part element="rti:domSetTracking" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamHasContrastRequest">
  <wsdl:part element="rti:scamHasContrast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barGetMeasureUnitResponse">
  <wsdl:part element="rti:barGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamAbortExposureResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="camGetROINumXRequest">
  <wsdl:part element="rti:camGetROINumX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanGetCoolerPowerRequest">
  <wsdl:part element="rti:camCanGetCoolerPower" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCCDTemperatureRequest">
  <wsdl:part element="rti:camGetCCDTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldGetMeasureUnitResponse">
  <wsdl:part element="rti:cldGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFullWellCapacityResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingAscensionRateResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamIsPTSupportedResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageURLRequest">
  <wsdl:part element="rti:camGetImageURL" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROIStartXRequest">
  <wsdl:part element="rti:camSetROIStartX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devUpdateDevicePropertyResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtGetMeasureRequest">
  <wsdl:part element="rti:fhtGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStopExposureResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasGainRequest">
  <wsdl:part element="rti:camHasGain" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rttSetTargetPositionRequest">
  <wsdl:part element="rti:rttSetTargetPosition" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetPixelSizeYResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetAZParkPosResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageURLProperFormatRequest">
  <wsdl:part element="rti:camGetImageURLProperFormat" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetTiltRotationResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFilterWheelsResponse">
  <wsdl:part element="rti:camGetFilterWheelsResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetTemperatureRequest">
  <wsdl:part element="rti:focGetTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCCDCurrentTemperatureResponse">
```

```
  <wsdl:part element="rti:camGetCCDCurrentTemperatureResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="focGetCameraResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewCoordinatesAsyncRequest">
  <wsdl:part element="rti:mntCanSlewCoordinatesAsync" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewAltAzResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetUtcClockRequest">
  <wsdl:part element="rti:mntGetUtcClock" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamAbortExposureRequest">
  <wsdl:part element="rti:scamAbortExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToAltAzResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsGetMeasureStatesRequest">
  <wsdl:part element="rti:rhsGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetLastErrorRequest">
  <wsdl:part element="rti:camGetLastError" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetPanRotationResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveWestResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrGetChannelTypeResponse">
  <wsdl:part element="rti:dwrGetChannelTypeResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetBrightnessRequest">
  <wsdl:part element="rti:scamGetBrightness" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContrastResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetBinYRequest">
  <wsdl:part element="rti:camSetBinY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFPSResponse">
  <wsdl:part element="rti:camGetFPSResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetTiltMinResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetFilterKindResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamStartExposureResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntTrackingRatesRequest">
  <wsdl:part element="rti:mntTrackingRates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execStopOpResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="mntMoveNorthRequest">
  <wsdl:part element="rti:mntMoveNorth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndIsRainingRequest">
  <wsdl:part element="rti:rndIsRaining" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camImageReadyRequest">
  <wsdl:part element="rti:camImageReady" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetZoomRequest">
  <wsdl:part element="rti:scamGetZoom" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetMaxStepRequest">
  <wsdl:part element="rti:focGetMaxStep" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetExposureTimeRequest">
  <wsdl:part element="rti:camGetExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetDigitilizingModeResponse">
  <wsdl:part element="rti:camGetDigitilizingModeResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="scamHasBrightnessRequest">
  <wsdl:part element="rti:scamHasBrightness" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetElectronsPerADUResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasBrightnessRequest">
  <wsdl:part element="rti:camHasBrightness" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="domIsAtHomeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetUtcClockResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFPSRequest">
  <wsdl:part element="rti:camGetFPS" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContinueModeImageFormatsRequest">
  <wsdl:part element="rti:camGetContinueModeImageFormats" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDevicesRequest">
  <wsdl:part element="rti:devGetDevices" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetPixelSizeXResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execStartOpRequest">
  <wsdl:part element="rti:execStartOp" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCoolerPowerRequest">
  <wsdl:part element="rti:camGetCoolerPower" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetExposureTimeResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetParkRequest">
  <wsdl:part element="rti:mntSetPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwIsAtHomeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="camGetOneShotModeImageFormatsResponse">
  <wsdl:part element="rti:camGetOneShotModeImageFormatsResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="domGetNumberElementRequest">
  <wsdl:part element="rti:domGetNumberElement" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCCDCurrentTemperatureRequest">
  <wsdl:part element="rti:camGetCCDCurrentTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndGetMeasureStatesRequest">
  <wsdl:part element="rti:rndGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDistanceMeasureUnitRequest">
  <wsdl:part element="rti:strGetDistanceMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntStopSlewAxisRequest">
  <wsdl:part element="rti:mntStopSlewAxis" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTargetDeclinationResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCloseRequest">
  <wsdl:part element="rti:domClose" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rttSetTargetPositionResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsSlewingResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHeatSinkTemperatureRequest">
  <wsdl:part element="rti:camHeatSinkTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetTrackingRequest">
  <wsdl:part element="rti:mntSetTracking" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strSetOrientationMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDeviceRequest">
  <wsdl:part element="rti:devGetDevice" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camIsPulseGuidingRequest">
  <wsdl:part element="rti:camIsPulseGuiding" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetContrastRequest">
  <wsdl:part element="rti:camSetContrast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingDeclinationRateRequest">
  <wsdl:part element="rti:mntGetTrackingDeclinationRate" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="domMoveAzimuthResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanPulseGuideRequest">
  <wsdl:part element="rti:mntCanPulseGuide" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDistanceMeasureResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="camIsCoolerOnRequest">
  <wsdl:part element="rti:camIsCoolerOn" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtGetMeasureUnitRequest">
  <wsdl:part element="rti:fhtGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewObjectResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewAzAsyncResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasShutterResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetAutoGainRequest">
  <wsdl:part element="rti:camGetAutoGain" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDistanceMeasureStatesResponse">
  <wsdl:part element="rti:strGetDistanceMeasureStatesResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsParkedResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanAsymetricBinRequest">
  <wsdl:part element="rti:camCanAsymetricBin" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetGuideRateDeclinationResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempGetMeasureUnitResponse">
  <wsdl:part element="rti:tempGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPosAxis3Response">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPosAxis1Response">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasGammaResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasShutterRequest">
  <wsdl:part element="rti:camHasShutter" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetMeasureUnitRequest">
  <wsdl:part element="rti:wvnGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetMeasureStatesRequest">
  <wsdl:part element="rti:wvnGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetTrackingRateResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetOneShotModeImagePathResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldGetMeasureStatesRequest">
  <wsdl:part element="rti:cldGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domIsAtHomeRequest">
  <wsdl:part element="rti:domIsAtHome" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="fwSetOffsetRequest">
  <wsdl:part element="rti:fwSetOffset" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camImageReadyResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGetTrackingResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasConstrastResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetOneShotModeImagePathRequest">
  <wsdl:part element="rti:camGetOneShotModeImagePath" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanControlTemperatureRequest">
  <wsdl:part element="rti:camCanControlTemperature" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetImageRequest">
  <wsdl:part element="rti:scamGetImage" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROINumXResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDeviceResponse">
  <wsdl:part element="rti:devGetDeviceResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsSlewingRequest">
  <wsdl:part element="rti:mntIsSlewing" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewObjectRequest">
  <wsdl:part element="rti:mntSlewObject" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwSelectFilterKindRequest">
  <wsdl:part element="rti:fwSelectFilterKind" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devDisconnectResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetMinStepRequest">
  <wsdl:part element="rti:focGetMinStep" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCanSetAltitudeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wspGetMeasureUnitRequest">
  <wsdl:part element="rti:wspGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devDisconnectRequest">
  <wsdl:part element="rti:devDisconnect" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetXsizeResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetAutoGainResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDistanceMeasureUnitResponse">
  <wsdl:part element="rti:strGetDistanceMeasureUnitResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="focGetMinStepResponse">
```

```
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rttGetCurrentPositionRequest">
  <wsdl:part element="rti:rttGetCurrentPosition" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focHaltRequest">
  <wsdl:part element="rti:focHalt" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetTrackingRateRequest">
  <wsdl:part element="rti:mntSetTrackingRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetParkResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetAutoGainResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetXsizeRequest">
  <wsdl:part element="rti:camGetXsize" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetAcquisitionModeRequest">
  <wsdl:part element="rti:camGetAcquisitionMode" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetLastExposureDurationResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldGetMeasureUnitRequest">
  <wsdl:part element="rti:cldGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camIsCoolerOnResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetCoolerOnRequest">
  <wsdl:part element="rti:camSetCoolerOn" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntPulseGuideRequest">
  <wsdl:part element="rti:mntPulseGuide" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetSlewRateRequest">
  <wsdl:part element="rti:mntSetSlewRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domMoveAzimuthRequest">
  <wsdl:part element="rti:domMoveAzimuth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndSetMeasureStatesRequest">
  <wsdl:part element="rti:rndSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasConstrastRequest">
  <wsdl:part element="rti:camHasConstrast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetGammaResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamHasContrastResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempGetMeasureStatesRequest">
  <wsdl:part element="rti:tempGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focIsTempCompAvailableResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="execStopOpRequest">
  <wsdl:part element="rti:execStopOp" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetContrastRequest">
  <wsdl:part element="rti:scamGetContrast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewCoordinatesResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetOrientationMeasureStatesResponse">
  <wsdl:part element="rti:strGetOrientationMeasureStatesResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="rndGetMeasureUnitResponse">
  <wsdl:part element="rti:rndGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsPointingAtCoordinatesRequest">
  <wsdl:part element="rti:mntIsPointingAtCoordinates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasGainResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCloseResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwIsAtHomeRequest">
  <wsdl:part element="rti:fwIsAtHome" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBitDepthResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetExposureTimeRequest">
  <wsdl:part element="rti:scamSetExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetMaxIncrementResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetAbosoluteDegreesRequest">
  <wsdl:part element="rti:strGetAbosoluteDegrees" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageDataTypeResponse">
  <wsdl:part element="rti:camGetImageDataTypeResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwSetOffsetResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focIsTempCompAvailableRequest">
  <wsdl:part element="rti:focIsTempCompAvailable" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domParkResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camIsPulseGuidingResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToAltAzRequest">
  <wsdl:part element="rti:mntSlewToAltAz" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focMoveResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wspSetMeasureStatesRequest">
  <wsdl:part element="rti:wspSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strSetDistanceMeasureStatesResponse">
```

```
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetContinueModeQualityResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntMoveEastRequest">
  <wsdl:part element="rti:mntMoveEast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROIStartYRequest">
  <wsdl:part element="rti:camGetROIStartY" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROINumXResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domSetParkResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camAbortExposureRequest">
  <wsdl:part element="rti:camAbortExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execGetInfoResponse">
  <wsdl:part element="rti:execGetInfoResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetTiltMaxRequest">
  <wsdl:part element="rti:scamGetTiltMax" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetMaxBinXRequest">
  <wsdl:part element="rti:camGetMaxBinX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetUtcClockRequest">
  <wsdl:part element="rti:mntSetUtcClock" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="camCanStopExposureRequest">
  <wsdl:part element="rti:camCanStopExposure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntParkResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetTiltMinRequest">
  <wsdl:part element="rti:scamGetTiltMin" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetMeasureStatesResponse">
  <wsdl:part element="rti:wvnGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBinXRequest">
  <wsdl:part element="rti:camGetBinX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetPanMinResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetLastExposureStartRequest">
  <wsdl:part element="rti:camGetLastExposureStart" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetStepSizeRequest">
  <wsdl:part element="rti:focGetStepSize" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContinueModeQualityRequest">
  <wsdl:part element="rti:camGetContinueModeQuality" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPosAxis2Request">
  <wsdl:part element="rti:mntGetPosAxis2" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROIStartXResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="camGetAutoExposureTimeRequest">
  <wsdl:part element="rti:camGetAutoExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGoHomeRequest">
  <wsdl:part element="rti:mntGoHome" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetMaxAduResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetContinueModeQualityRequest">
  <wsdl:part element="rti:camSetContinueModeQuality" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetGammaRequest">
  <wsdl:part element="rti:camGetGamma" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCanSetParkResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnGetAbosluteDegreesResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetGainResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strIsAvailableOrientationRequest">
  <wsdl:part element="rti:strIsAvailableOrientation" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetConfigurationRequest">
  <wsdl:part element="rti:devGetConfiguration" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domGoHomeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntStopSlewRequest">
  <wsdl:part element="rti:mntStopSlew" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="dwrGetChannelsNumberResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStopContinueModeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetTrackingRequest">
  <wsdl:part element="rti:mntCanSetTracking" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetPanRotationRequest">
  <wsdl:part element="rti:scamSetPanRotation" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetPanMaxRequest">
  <wsdl:part element="rti:scamGetPanMax" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetZoomRequest">
  <wsdl:part element="rti:scamSetZoom" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetGainResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetPointingModelRequest">
  <wsdl:part element="rti:mntGetPointingModel" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devUpdateDevicePropertyAsyncRequest">
  <wsdl:part element="rti:devUpdateDevicePropertyAsync" name="parameters">
```

```
  </wsdl:part>
</wsdl:message>
<wsdl:message name="focIsAbsoluteResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanPulseGuideResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetFilterListResponse">
  <wsdl:part element="rti:fwGetFilterListResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDevicesResponse">
  <wsdl:part element="rti:devGetDevicesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGoHomeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetAutoExposureTimeRequest">
  <wsdl:part element="rti:camSetAutoExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetOneShotModeQualityResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetTrackingResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camSetROIStartYResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSetTrackingResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="mntGetSlewRateRequest">
  <wsdl:part element="rti:mntGetSlewRate" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCameraTypeRequest">
  <wsdl:part element="rti:camGetCameraType" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="tempSetMeasureStatesResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barSetMeasureStatesRequest">
  <wsdl:part element="rti:barSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetPositionNumberResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsGetMeasureUnitRequest">
  <wsdl:part element="rti:rhsGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStopContinueModeRequest">
  <wsdl:part element="rti:camStopContinueMode" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="execStartOpResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndIsRainingResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetPanRotationRequest">
  <wsdl:part element="rti:scamGetPanRotation" name="parameters"></wsdl:part>
</wsdl:message>
```

```
<wsdl:message name="wspGetMeasureRequest">
  <wsdl:part element="rti:wspGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToCoordinatesRequest">
  <wsdl:part element="rti:mntSlewToCoordinates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndGetMeasureUnitRequest">
  <wsdl:part element="rti:rndGetMeasureUnit" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamIsZoomSupportedRequest">
  <wsdl:part element="rti:scamIsZoomSupported" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndGetMeasureRequest">
  <wsdl:part element="rti:rndGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwSelectFilterKindResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fhtGetMeasureStatesRequest">
  <wsdl:part element="rti:fhtGetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetZoomResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetContrastResponse">
  <wsdl:part element="rti:longResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rndGetMeasureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="rhsGetMeasureUnitResponse">
  <wsdl:part element="rti:rhsGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHeatSinkTemperatureResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetLastErrorResponse">
  <wsdl:part element="rti:stringResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsParkedRequest">
  <wsdl:part element="rti:mntIsParked" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wspGetMeasureUnitResponse">
  <wsdl:part element="rti:wspGetMeasureUnitResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetFocuserRequest">
  <wsdl:part element="rti:camGetFocuser" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetZoomResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamIsPTSupportedRequest">
  <wsdl:part element="rti:scamIsPTSupported" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetContrastRequest">
  <wsdl:part element="rti:scamSetContrast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetLastExposureStartResponse">
  <wsdl:part element="rti:camGetLastExposureStartResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTrackingRequest">
  <wsdl:part element="rti:mntGetTracking" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamStopExposureResponse">
```

```
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetOneShotModeImageFormatsRequest">
  <wsdl:part element="rti:camGetOneShotModeImageFormats" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="wspGetMeasureStatesResponse">
  <wsdl:part element="rti:wspGetMeasureStatesResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamSetContrastResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetExposureTimeResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetZoomMaxResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSlewObjectRequest">
  <wsdl:part element="rti:mntCanSlewObject" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="fwGetFilterSizeResponse">
  <wsdl:part element="rti:fwGetFilterSizeResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDegreesRequest">
  <wsdl:part element="rti:strGetDegrees" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasSubframeRequest">
  <wsdl:part element="rti:camHasSubframe" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetOneShotModeQualityResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCanSetParkRequest">
  <wsdl:part element="rti:domCanSetPark" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetParkResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetBinXResponse">
  <wsdl:part element="rti:intResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camCanAbortExposureResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focGetPositionRequest">
  <wsdl:part element="rti:focGetPosition" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="barGetMeasureRequest">
  <wsdl:part element="rti:barGetMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="cldSetMeasureStatesRequest">
  <wsdl:part element="rti:cldSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetDistanceMeasureRequest">
  <wsdl:part element="rti:strGetDistanceMeasure" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetObjectExposureTimeRequest">
  <wsdl:part element="rti:camGetObjectExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntAxisRateResponse">
  <wsdl:part element="rti:mntAxisRateResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanSetGuideRatesResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="mntGetALTParkPosResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domCanSetAzimuthRequest">
  <wsdl:part element="rti:domCanSetAzimuth" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camStartContinueModeRequest">
  <wsdl:part element="rti:camStartContinueMode" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanMoveAzisResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetExposureTimeRequest">
  <wsdl:part element="rti:scamGetExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devGetDevicePropertyResponse">
  <wsdl:part element="rti:devGetDevicePropertyResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="mntIsAtHomeRequest">
  <wsdl:part element="rti:mntIsAtHome" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContinueModeImagePathRequest">
  <wsdl:part element="rti:camGetContinueModeImagePath" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetContrastRequest">
  <wsdl:part element="rti:camGetContrast" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="wvnSetMeasureStatesRequest">
  <wsdl:part element="rti:wvnSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="strGetOrientationMeasureStatesRequest">
  <wsdl:part element="rti:strGetOrientationMeasureStates" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="logoutRequest">
  <wsdl:part element="rti:logout" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetDeclinationRateRightAscensionResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="loginCertRequest">
  <wsdl:part element="rti:loginCert" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasExposureTimeRequest">
  <wsdl:part element="rti:camHasExposureTime" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camHasSubframeResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTargetRightAscensionRequest">
  <wsdl:part element="rti:mntGetTargetRightAscension" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamGetVideoStreamingURLRequest">
  <wsdl:part element="rti:scamGetVideoStreamingURL" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="scamHasBrightnessResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetTargetRightAscensionResponse">
  <wsdl:part element="rti:doubleResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="domMoveAltitudeResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="fhtSetMeasureStatesRequest">
  <wsdl:part element="rti:fhtSetMeasureStates" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntCanMoveAzisRequest">
  <wsdl:part element="rti:mntCanMoveAzis" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntSlewToCoordinatesAsyncRequest">
  <wsdl:part element="rti:mntSlewToCoordinatesAsync" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetImageResponse">
  <wsdl:part element="rti:camGetImageResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetCameraTypeResponse">
  <wsdl:part element="rti:camGetCameraTypeResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="devIsBlockedResponse">
  <wsdl:part element="rti:booleanResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="mntGetAZParkPosRequest">
  <wsdl:part element="rti:mntGetAZParkPos" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="camGetROIStartXRequest">
  <wsdl:part element="rti:camGetROIStartX" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="focSetTempCompResponse">
  <wsdl:part element="rti:voidResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:portType name="gloria_rti">
  <wsdl:operation name="loginCert">
    <wsdl:documentation>
      Authenticates the user using X509 private key of Gloria System. @param user User
      login. @param time Client datetime. @param sign Sign using private key of Gloria
      System. The sign content is: SIGN[user + time]. @param sessionType TOKEN |
      COOKIE @return Session token if a cookie is not used. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:loginCertRequest"></wsdl:input>
    <wsdl:output message="rti:loginCertResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="login">
    <wsdl:documentation>
      Authenticates the user in the system. @param user User login. @param pw
      Password. @param sessionType TOKEN | COOKIE @return Session token if a cookie is
      not used. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:loginRequest"></wsdl:input>
    <wsdl:output message="rti:loginResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="logout">
    <wsdl:documentation>
      Closes the user session. @param session Session token if the used login method
      is based on session token type. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:logoutRequest"></wsdl:input>
    <wsdl:output message="rti:logoutResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="devGetDevice">
    <wsdl:documentation>
      Recover the information of one device. @param session Session token if the used
      login method is based on session token type. @param deviceId Device Identifier.
      @return Current device data. @throws RtiError In error case.
```

```
  </wsdl:documentation>
  <wsdl:input message="rti:devGetDeviceRequest"></wsdl:input>
  <wsdl:output message="rti:devGetDeviceResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="devGetDevices">
  <wsdl:documentation>
    Recover the information of all devices. @param session Session token if the used
    login method is based on session token type. @return Current devices list.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:devGetDevicesRequest"></wsdl:input>
  <wsdl:output message="rti:devGetDevicesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="devGetDeviceProperties">
  <wsdl:documentation>
    Recover the properties of a device. @param session Session token if the used
    login method is based on session token type. @param deviceId Device Identifier.
    @return Properties List. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:devGetDevicePropertiesRequest"></wsdl:input>
  <wsdl:output message="rti:devGetDevicePropertiesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="devGetDeviceProperty">
  <wsdl:documentation>
    Recover a property of a device. @param session Session token if the used login
    method is based on session token type. @param deviceId Device Identifier. @param
    name Property name @return Properties List. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:devGetDevicePropertyRequest"></wsdl:input>
  <wsdl:output message="rti:devGetDevicePropertyResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="devUpdateDeviceProperty">
  <wsdl:documentation>
    Updates a property into a Device configuration. If the property is a readonly
    property or the value cannot be (internally) transformed into the proper type,
    an exception will be throw. @param session Session token if the used login
    method is based on session token type. @param deviceId Device Identifier. @param
    name Property name. @param value Values of the property. @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:devUpdateDevicePropertyRequest"></wsdl:input>
  <wsdl:output message="rti:devUpdateDevicePropertyResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="devUpdateDevicePropertyAsync">
  <wsdl:documentation>
    Updates a property into a Device configuration. If the property is a readonly
    property or the value cannot be (internally) transformed into the proper type,
    an exception will be throw. Asynchronous call. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    Identifier. @param name Property name. @param value Values of the property.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:devUpdateDevicePropertyAsyncRequest"></wsdl:input>
  <wsdl:output message="rti:devUpdateDevicePropertyAsyncResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="devIsConnected">
  <wsdl:documentation>
    Checks the link between the driver and the device. If True, the link is enable.
    @param session Session token if the used login method is based on session token
```

```
      type. @param deviceId Device Identifier. @return boolean. @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:devIsConnectedRequest"></wsdl:input>
    <wsdl:output message="rti:devIsConnectedResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="devConnect">
    <wsdl:documentation>
      Creates the link between the driver and the device. @param session Session token
      if the used login method is based on session token type. @param deviceId Device
      Identifier. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:devConnectRequest"></wsdl:input>
    <wsdl:output message="rti:devConnectResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="devDisconnect">
    <wsdl:documentation>
      Controls the link between the driver and the device. Set True to enable the
      link. Set False to disable the link. @param session Session token if the used
      login method is based on session token type. @param deviceId Device Identifier.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:devDisconnectRequest"></wsdl:input>
    <wsdl:output message="rti:devDisconnectResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="devIsBlocked">
    <wsdl:documentation>
      Returns true if the device is blocked because of its blocked state or some
      dependency to other device. @param session Session token if the used login
      method is based on session token type. @param deviceId Device Identifier.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:devIsBlockedRequest"></wsdl:input>
    <wsdl:output message="rti:devIsBlockedResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="devGetConfiguration">
    <wsdl:documentation>
      Returns a String containing a full configuration information. For logging
      purpose. @param session Session token if the used login method is based on
      session token type. @param deviceId Device Id. @return Configuration in a
      string. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:devGetConfigurationRequest"></wsdl:input>
    <wsdl:output message="rti:devGetConfigurationResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camImageReady">
    <wsdl:documentation>
      If True, there is an image from the camera available. If False, no image is
      available . @param session Session token if the used login method is based on
      session token type. @param deviceId Device identifier. @return Boolean value.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camImageReadyRequest"></wsdl:input>
    <wsdl:output message="rti:camImageReadyResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetCameraType">
    <wsdl:documentation>
      Returns the camera type. @param session Session token if the used login method
      is based on session token type. @return CameraType. @throws RtiError In error
```

```
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetCameraTypeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetCameraTypeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetXsize">
  <wsdl:documentation>
    Width of the camera sensor in unbinned pixels. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return width. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetXsizeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetXsizeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetYSize">
  <wsdl:documentation>
    Height of the camera sensor in unbinned pixels. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifiers. @return height. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetYSizeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetYSizeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camCanAbortExposure">
  <wsdl:documentation>
    Returns True if the camera can abort exposures. False if not. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Identifer identifier. @return Boolean value.
  </wsdl:documentation>
  <wsdl:input message="rti:camCanAbortExposureRequest"></wsdl:input>
  <wsdl:output message="rti:camCanAbortExposureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camCanAsymetricBin">
  <wsdl:documentation>
    Returns true, the camera can have different binning on the X and Y axes, as
    determined by BinX and BinY. If False, the binning must be equal on the X and Y
    axes. @param session Session token if the used login method is based on session
    token type. @param deviceId Device identifier. @return Boolean value. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camCanAsymetricBinRequest"></wsdl:input>
  <wsdl:output message="rti:camCanAsymetricBinResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camCanGetCoolerPower">
  <wsdl:documentation>
    If True, the camera can return the cooler power level. If False, this
    information is not available. The cooler power level is normally regulated
    internally to the camera, based on the temperature setpoint. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Boolean value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:camCanGetCoolerPowerRequest"></wsdl:input>
  <wsdl:output message="rti:camCanGetCoolerPowerResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camCanSetCCDTemperature">
  <wsdl:documentation>
    If True, the camera's cooler setpoint can be adjusted. If False, the camera
    either uses open-loop cooling or does not have the ability to adjust temperature
```

```
      from software, and setting the TemperatureSetpoint property has no effect.
      @param session Session token if the used login method is based on session token
      type. @param deviceId Device identifier. @return Boolean value. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camCanSetCCDTemperatureRequest"></wsdl:input>
    <wsdl:output message="rti:camCanSetCCDTemperatureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camCanControlTemperature">
    <wsdl:documentation>
      If True, the CCD Temperature and the Heat Sink temperature can be read. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device Identifier. @return Boolean value. @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camCanControlTemperatureRequest"></wsdl:input>
    <wsdl:output message="rti:camCanControlTemperatureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camCanStopExposure">
    <wsdl:documentation>
      Some cameras support StopExposure, which allows the exposure to be terminated
      before the exposure timer completes, but will still read out the image. Returns
      True if StopExposure is available, False if not. @param session Session token if
      the used login method is based on session token type. @param deviceId Device
      identifier. @return Boolean value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camCanStopExposureRequest"></wsdl:input>
    <wsdl:output message="rti:camCanStopExposureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetCoolerPower">
    <wsdl:documentation>
      Returns the present cooler power level, in percent. Returns zero if CoolerOn is
      False. @param session Session token if the used login method is based on session
      token type. @param deviceId Device identifier. @return Measure. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetCoolerPowerRequest"></wsdl:input>
    <wsdl:output message="rti:camGetCoolerPowerResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetElectronsPerADU">
    <wsdl:documentation>
      Returns the gain of the camera in photoelectrons per A/D unit. (Some cameras
      have multiple gain modes; these should be selected via the SetupDialog and thus
      are static during a session) @param session Session token if the used login
      method is based on session token type. @param deviceId Device identifier.
      @return Value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetElectronsPerADURequest"></wsdl:input>
    <wsdl:output message="rti:camGetElectronsPerADUResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetFullWellCapacity">
    <wsdl:documentation>
      Reports the full well capacity of the camera in electrons, at the current camera
      settings [FullWellCapacity]. @param session Session token if the used login
      method is based on session token type. @param deviceId Device identifier.
      @return Value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetFullWellCapacityRequest"></wsdl:input>
    <wsdl:output message="rti:camGetFullWellCapacityResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
    </wsdl:operation>
    <wsdl:operation name="camHasShutter">
      <wsdl:documentation>
        If True, the camera has a mechanical shutter. If False, the camera does not have
        a shutter. If there is no shutter, the StartExposure command will ignore the
        Light parameter. @param session Session token if the used login method is based
        on session token type. @param deviceId Device identifier. @return Value. @throws
        RtiError In error case.
      </wsdl:documentation>
      <wsdl:input message="rti:camHasShutterRequest"></wsdl:input>
      <wsdl:output message="rti:camHasShutterResponse"></wsdl:output>
      <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="camHasBrightness">
      <wsdl:documentation>
        If True, the camera's brightness can be adjusted. If False, the camera can not
        adjust the brightness. @param session Session token if the used login method is
        based on session token type. @param deviceId Device identifier. @return Boolean
        value. @throws RtiError In error case.
      </wsdl:documentation>
      <wsdl:input message="rti:camHasBrightnessRequest"></wsdl:input>
      <wsdl:output message="rti:camHasBrightnessResponse"></wsdl:output>
      <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="camHasConstrast">
      <wsdl:documentation>
        If True, the camera's contrast can be adjusted. If False, the camera can not
        adjust the contrast. @param session Session token if the used login method is
        based on session token type. @param deviceId Device identifier. @return Boolean
        value. @throws RtiError In error case.
      </wsdl:documentation>
      <wsdl:input message="rti:camHasConstrastRequest"></wsdl:input>
      <wsdl:output message="rti:camHasConstrastResponse"></wsdl:output>
      <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="camHasGain">
      <wsdl:documentation>
        If True, the camera's gain can be adjusted. If False, the camera can not adjust
        the gain. @param session Session token if the used login method is based on
        session token type. @param deviceId Device identifier. @return Boolean value.
        @throws RtiError In error case.
      </wsdl:documentation>
      <wsdl:input message="rti:camHasGainRequest"></wsdl:input>
      <wsdl:output message="rti:camHasGainResponse"></wsdl:output>
      <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="camHasGamma">
      <wsdl:documentation>
        If True, the camera’s gamma can be adjusted. If False, the camera can not adjust
        the gamma. @param session Session token if the used login method is based on
        session token type. @param deviceId Device identifier. @return Boolean value.
        @throws RtiError In error case.
      </wsdl:documentation>
      <wsdl:input message="rti:camHasGammaRequest"></wsdl:input>
      <wsdl:output message="rti:camHasGammaResponse"></wsdl:output>
      <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="camHasSubframe">
      <wsdl:documentation>
        If True,the camera supports subframes. @param session Session token if the used
        login method is based on session token type. @param deviceId Device identifier.
        @return Boolean value. @throws RtiError In error case.
      </wsdl:documentation>
      <wsdl:input message="rti:camHasSubframeRequest"></wsdl:input>
      <wsdl:output message="rti:camHasSubframeResponse"></wsdl:output>
```

```
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camHasExposureTime">
  <wsdl:documentation>
    If True, the camera's exposure time can be changed. If False, the camera can not
    change the exposure time. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier. @return
    Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camHasExposureTimeRequest"></wsdl:input>
  <wsdl:output message="rti:camHasExposureTimeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camHeatSinkTemperature">
  <wsdl:documentation>
    Returns the current heat sink temperature (called "ambient temperature" by some
    manufacturers) in degrees Celsius. Only valid if CanControlTemperature is True .
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Measure. @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camHeatSinkTemperatureRequest"></wsdl:input>
  <wsdl:output message="rti:camHeatSinkTemperatureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camIsPulseGuiding">
  <wsdl:documentation>
    If True, pulse guiding is in progress. Required if the PulseGuide() method
    (which is non-blocking) is implemented. See the PulseGuide. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Boolean value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:camIsPulseGuidingRequest"></wsdl:input>
  <wsdl:output message="rti:camIsPulseGuidingResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetLastError">
  <wsdl:documentation>
    Reports the last error condition reported by the camera hardware or
    communications link. The string may contain a text message or simply an error
    code. The error value is cleared the next time any method is called. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @return String. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetLastErrorRequest"></wsdl:input>
  <wsdl:output message="rti:camGetLastErrorResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetLastExposureDuration">
  <wsdl:documentation>
    Reports the actual exposure duration in seconds (i.e. shutter open time). This
    may differ from the exposure time requested due to shutter latency, camera
    timing precision, etc. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @return Seconds.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetLastExposureDurationRequest"></wsdl:input>
  <wsdl:output message="rti:camGetLastExposureDurationResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetLastExposureStart">
  <wsdl:documentation>
    Reports the actual exposure start. It could be formatted in the FITS-standard
```

```
    CCYY-MM-DDThh:mm:ss[.sss...] format. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return timestamp. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetLastExposureStartRequest"></wsdl:input>
  <wsdl:output message="rti:camGetLastExposureStartResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetMaxAdu">
  <wsdl:documentation>
    Reports the maximum ADU value the camera can produce. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @return maximum ADU. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetMaxAduRequest"></wsdl:input>
  <wsdl:output message="rti:camGetMaxAduResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetMaxBinX">
  <wsdl:documentation>
    If AsymmetricBinning = False, returns the maximum allowed binning factor. If
    AsymmetricBinning = True, returns the maximum allowed binning factor for the X
    axis. @param session Session token if the used login method is based on session
    token type. @param deviceId Device identifier. @return The maximum allowed
    binning factor for the X axis. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetMaxBinXRequest"></wsdl:input>
  <wsdl:output message="rti:camGetMaxBinXResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetMaxBinY">
  <wsdl:documentation>
    If AsymmetricBinning = False, equals MaxBinX. If AsymmetricBinning = True,
    returns the maximum allowed binning factor for the Y axis. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return The maximum allowed binning factor for the Y
    axis. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetMaxBinYRequest"></wsdl:input>
  <wsdl:output message="rti:camGetMaxBinYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetPixelSizeX">
  <wsdl:documentation>
    Returns the width of the sensor chip pixels in microns, as provided by the
    camera driver. @param session Session token if the used login method is based on
    session token type. @param deviceId Device identifier. @return Width. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetPixelSizeXRequest"></wsdl:input>
  <wsdl:output message="rti:camGetPixelSizeXResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetPixelSizeY">
  <wsdl:documentation>
    Returns the heigth of the sensor chip pixels in microns, as provided by the
    camera driver. @param session Session token if the used login method is based on
    session token type. @param deviceId Device identifier. @return Heigth. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetPixelSizeYRequest"></wsdl:input>
  <wsdl:output message="rti:camGetPixelSizeYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetAcquisitionMode">
```

```
    <wsdl:documentation>
      If it is Continuous mode, the camera will take images according to the
      ExposureTime and FPS values established. If it is OneShot mode, only one image
      will be taken according to the ExposureTime. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return Continuous, OneShot. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetAcquisitionModeRequest"></wsdl:input>
    <wsdl:output message="rti:camGetAcquisitionModeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetFPS">
    <wsdl:documentation>
      Frames Per Second. It only takes effect in the “Continuous” Acquisition Mode.
      @param deviceId Device identifier. @return Frames Per Second. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetFPSRequest"></wsdl:input>
    <wsdl:output message="rti:camGetFPSResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetDigitilizingMode">
    <wsdl:documentation>
      If it is Automatic mode, the camera will take images according to the
      Brightness, Contrast, Gamma and Gain values calculated automatically by the
      camera itself. If it is Manual mode, the camera will take images according to
      the Brightness, Contrast, Gamma and Gain values established by the variables.
      @param session Session token if the used login method is based on session token
      type. @param deviceId Device identifier. @return Automatic, Manual. @throws
      RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetDigitilizingModeRequest"></wsdl:input>
    <wsdl:output message="rti:camGetDigitilizingModeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetBinX">
    <wsdl:documentation>
      Gets the binning factor for the X axis. Also returns the current value. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device identifier. @return The binning factor for the X axis.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetBinXRequest"></wsdl:input>
    <wsdl:output message="rti:camGetBinXResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camSetBinX">
    <wsdl:documentation>
      Sets the binning factor for the X axis. Also returns the current value. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device identifier. @param value The binning factor for the X
      axis. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camSetBinXRequest"></wsdl:input>
    <wsdl:output message="rti:camSetBinXResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetBinY">
    <wsdl:documentation>
      Gets the binning factor for the Y axis. Also returns the current value. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device identifier. @return The binning factor for the Y axis.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetBinYRequest"></wsdl:input>
```

```
  <wsdl:output message="rti:camGetBinYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetBinY">
  <wsdl:documentation>
    Sets the binning factor for the Y axis. Also returns the current value. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @param value The binning factor for the Y
    axis. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetBinYRequest"></wsdl:input>
  <wsdl:output message="rti:camSetBinYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camIsCoolerOn">
  <wsdl:documentation>
    Returns true if the camera cooler is on. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camIsCoolerOnRequest"></wsdl:input>
  <wsdl:output message="rti:camIsCoolerOnResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetCoolerOn">
  <wsdl:documentation>
    Turns on and off the camera cooler. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @param value new value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetCoolerOnRequest"></wsdl:input>
  <wsdl:output message="rti:camSetCoolerOnResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetROINumX">
  <wsdl:documentation>
    Returns the subframe width. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetROINumXRequest"></wsdl:input>
  <wsdl:output message="rti:camGetROINumXResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetROINumX">
  <wsdl:documentation>
    Sets the subframe width. If binning is active, value is in binned pixels. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @param value new value. @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetROINumXRequest"></wsdl:input>
  <wsdl:output message="rti:camSetROINumXResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetROINumY">
  <wsdl:documentation>
    Sets the subframe height. If binning is active, value is in binned pixels.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @param value New value. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetROINumYRequest"></wsdl:input>
  <wsdl:output message="rti:camSetROINumYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="camGetROINumY">
  <wsdl:documentation>
    Returns the subframe height. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return subframe height. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetROINumYRequest"></wsdl:input>
  <wsdl:output message="rti:camGetROINumYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetROIStartX">
  <wsdl:documentation>
    Sets the subframe start position for the X axis (0 based). If binning is active,
    value is in binned pixels. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier. @param
    ROIStartX new value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetROIStartXRequest"></wsdl:input>
  <wsdl:output message="rti:camSetROIStartXResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetROIStartX">
  <wsdl:documentation>
    Returns the subframe start position for the X axis (0 based). @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Subframe start position for the X axis.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetROIStartXRequest"></wsdl:input>
  <wsdl:output message="rti:camGetROIStartXResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetROIStartY">
  <wsdl:documentation>
    Sets the subframe start position for the Y axis (0 based). If binning is active,
    value is in binned pixels. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier. @param value
    The subframe start position for the Y axis. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetROIStartYRequest"></wsdl:input>
  <wsdl:output message="rti:camSetROIStartYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetROIStartY">
  <wsdl:documentation>
    Returns The subframe start position for the Y axis (0 based). @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return The subframe start position for the Y axis.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetROIStartYRequest"></wsdl:input>
  <wsdl:output message="rti:camGetROIStartYResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetBrightness">
  <wsdl:documentation>
    Sets the value for image brightness. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @param value brightness. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetBrightnessRequest"></wsdl:input>
  <wsdl:output message="rti:camSetBrightnessResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
```

```
<wsdl:operation name="camGetBrightness">
  <wsdl:documentation>
    Returns the value of image brightness. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return image brightness. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetBrightnessRequest"></wsdl:input>
  <wsdl:output message="rti:camGetBrightnessResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetContrast">
  <wsdl:documentation>
    Sets the value for image contrast. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @param value New value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetContrastRequest"></wsdl:input>
  <wsdl:output message="rti:camSetContrastResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetContrast">
  <wsdl:documentation>
    Returns the image contrast. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return Current value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetContrastRequest"></wsdl:input>
  <wsdl:output message="rti:camGetContrastResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetGain">
  <wsdl:documentation>
    Sets the value for image gain.. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    value New value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetGainRequest"></wsdl:input>
  <wsdl:output message="rti:camSetGainResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetGain">
  <wsdl:documentation>
    Returns the image gain. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @return Current
    value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetGainRequest"></wsdl:input>
  <wsdl:output message="rti:camGetGainResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetGamma">
  <wsdl:documentation>
    Sets the value for image gamma correction. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @param value New value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetGammaRequest"></wsdl:input>
  <wsdl:output message="rti:camSetGammaResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetGamma">
  <wsdl:documentation>
    Returns the image gamma correction. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return Current value. @throws RtiError In error case.
```

```
  </wsdl:documentation>
  <wsdl:input message="rti:camGetGammaRequest"></wsdl:input>
  <wsdl:output message="rti:camGetGammaResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetExposureTime">
  <wsdl:documentation>
    Sets the value in seconds for the image exposure time. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @param value New value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetExposureTimeRequest"></wsdl:input>
  <wsdl:output message="rti:camSetExposureTimeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetExposureTime">
  <wsdl:documentation>
    Returns the value in seconds for the image exposure time. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @return Current value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetExposureTimeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetExposureTimeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetObjectExposureTime">
  <wsdl:documentation>
    Returns the exposure time interval for the object and filter selected. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @param filterType filter @param object
    @return Current value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetObjectExposureTimeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetObjectExposureTimeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetCCDTemperature">
  <wsdl:documentation>
    Sets the CCD cooler setpoint. Only valid if CanControlTemperature is True.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @param value New value. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetCCDTemperatureRequest"></wsdl:input>
  <wsdl:output message="rti:camSetCCDTemperatureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetCCDTemperature">
  <wsdl:documentation>
    Returns the CCD cooler setpoint. Only valid if CanControlTemperature is True.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Current value. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetCCDTemperatureRequest"></wsdl:input>
  <wsdl:output message="rti:camGetCCDTemperatureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetCCDCurrentTemperature">
  <wsdl:documentation>
    Returns the current CCD temperature in degrees Kelvin. Only valid if
    CanControlTemperature is True. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return Current value. @throws RtiError In error case.
  </wsdl:documentation>
```

```
  <wsdl:input message="rti:camGetCCDCurrentTemperatureRequest"></wsdl:input>
  <wsdl:output message="rti:camGetCCDCurrentTemperatureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camAbortExposure">
  <wsdl:documentation>
    Aborts the current exposure, if any, and returns the camera to READY state.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camAbortExposureRequest"></wsdl:input>
  <wsdl:output message="rti:camAbortExposureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camPulseGuide">
  <wsdl:documentation>
    This method may return immediately after the move has started, in which case
    back-to-back dual axis pulse-guiding can be supported. Use the IsPulseGuiding
    property to detect when all moves have completed. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @param direction direction in which the guide-rate motion is to be
    made. The values for GuideDirections are: 0=guideNorth, 1=guideSouth,
    2=guideEast, 3=guideWest. @param duration Duration of the guide-rate motion
    (milliseconds). @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camPulseGuideRequest"></wsdl:input>
  <wsdl:output message="rti:camPulseGuideResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camStartExposure">
  <wsdl:documentation>
    Starts an exposure. Use ImageReady to check when the exposure is complete, and
    also use the AcquisitionMode and DigitizingMode values. This method uses
    ExposureTime variable. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @param light
    Frame type (true: light frame, false: darkframe). @return String Image uid
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camStartExposureRequest"></wsdl:input>
  <wsdl:output message="rti:camStartExposureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camStopExposure">
  <wsdl:documentation>
    Stops the current exposure, if any. If an exposure is in progress, the readout
    process is initiated. Ignored if readout is already in process. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camStopExposureRequest"></wsdl:input>
  <wsdl:output message="rti:camStopExposureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetImage">
  <wsdl:documentation>
    Returns an image from the last exposure. The image could be an array of short,
    long, double (imageContentType attribute shows the type). If the application
    cannot handle multispectral images, it should use just the first plane. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @return Image data. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetImageRequest"></wsdl:input>
  <wsdl:output message="rti:camGetImageResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="camGetOneShotModeImageFormats">
  <wsdl:documentation>
    Returns the list of image formats available in this camera for OneShot Mode.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Image Format list. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetOneShotModeImageFormatsRequest"></wsdl:input>
  <wsdl:output message="rti:camGetOneShotModeImageFormatsResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetContinueModeImageFormats">
  <wsdl:documentation>
    Returns the list of image formats available in this camera for Continue Mode.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Image Format list. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetContinueModeImageFormatsRequest"></wsdl:input>
  <wsdl:output message="rti:camGetContinueModeImageFormatsResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetImageDataType">
  <wsdl:documentation>
    Gets the image data type (Short, Long, Double) @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return Image Content type. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetImageDataTypeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetImageDataTypeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetBitDepth">
  <wsdl:documentation>
    Gets the current CCD Bit Depth. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return Current CCD Bit Depth @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetBitDepthRequest"></wsdl:input>
  <wsdl:output message="rti:camGetBitDepthResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetBitDepth">
  <wsdl:documentation>
    Sets the CCD Bit Depth. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @param bits Bit
    Depth. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetBitDepthRequest"></wsdl:input>
  <wsdl:output message="rti:camSetBitDepthResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetContinueModeQuality">
  <wsdl:documentation>
    Gets the quality in continue mode @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier @return
    int Quality value @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetContinueModeQualityRequest"></wsdl:input>
  <wsdl:output message="rti:camGetContinueModeQualityResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetContinueModeQuality">
  <wsdl:documentation>
```

```
    Sets the quality in continue mode @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetContinueModeQualityRequest"></wsdl:input>
  <wsdl:output message="rti:camSetContinueModeQualityResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetOneShotModeQuality">
  <wsdl:documentation>
    Gets the quality in oneShot mode @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier @return
    int Quality value @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetOneShotModeQualityRequest"></wsdl:input>
  <wsdl:output message="rti:camGetOneShotModeQualityResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetOneShotModeQuality">
  <wsdl:documentation>
    Sets the quality in oneShot mode @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetOneShotModeQualityRequest"></wsdl:input>
  <wsdl:output message="rti:camSetOneShotModeQualityResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetContinueModeImagePath">
  <wsdl:documentation>
    Gets the image path where images taken in continue mode are stored. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier @return String The image path @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetContinueModeImagePathRequest"></wsdl:input>
  <wsdl:output message="rti:camGetContinueModeImagePathResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetOneShotModeImagePath">
  <wsdl:documentation>
    Gets the image path where images taken in oneShot mode are stored. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier @return String The image path @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetOneShotModeImagePathRequest"></wsdl:input>
  <wsdl:output message="rti:camGetOneShotModeImagePathResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetAutoGain">
  <wsdl:documentation>
    Gets is gain parameter is auto-adjusted @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier
    @return True if gain parameter is auto-adjusted @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetAutoGainRequest"></wsdl:input>
  <wsdl:output message="rti:camGetAutoGainResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetAutoGain">
  <wsdl:documentation>
    Sets the gain parameter as auto-adjusted or not. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier @param value True if gain parameter is auto-adjusted. @throws
```

```
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetAutoGainRequest"></wsdl:input>
  <wsdl:output message="rti:camSetAutoGainResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetAutoExposureTime">
  <wsdl:documentation>
    Gets is exposure time parameter is auto-adjusted @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier @return True if gain parameter is auto-adjusted @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetAutoExposureTimeRequest"></wsdl:input>
  <wsdl:output message="rti:camGetAutoExposureTimeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camSetAutoExposureTime">
  <wsdl:documentation>
    Sets the exposure time parameter as auto-adjusted or not. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier @param value True if gain parameter is auto-adjusted. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camSetAutoExposureTimeRequest"></wsdl:input>
  <wsdl:output message="rti:camSetAutoExposureTimeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camStartContinueMode">
  <wsdl:documentation>
    Starts the continue mode. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier @return String
    Image uid @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camStartContinueModeRequest"></wsdl:input>
  <wsdl:output message="rti:camStartContinueModeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camStopContinueMode">
  <wsdl:documentation>
    Stops the continue mode. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:camStopContinueModeRequest"></wsdl:input>
  <wsdl:output message="rti:camStopContinueModeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetImageURL">
  <wsdl:documentation>
    Returns the URL where the image is stored. The image is in the default format.
    @deprecated camGetImageURLProperFormat method is the right option. This method
    will be removed in future versions. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier
    @param uid Image identificator @return String URL @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:camGetImageURLRequest"></wsdl:input>
  <wsdl:output message="rti:camGetImageURLResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="camGetImageURLProperFormat">
  <wsdl:documentation>
    Returns the URL where the image is stored @param session Session token if the
    used login method is based on session token type. @param deviceId Device
```

```
      identifier @param uid Image identificator @param uid Image Format @return String
      URL @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetImageURLProperFormatRequest"></wsdl:input>
    <wsdl:output message="rti:camGetImageURLProperFormatResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetFilterWheels">
    <wsdl:documentation>
      Returns the filter wheels identifiers associated to the camera. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier @return Identifier list @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetFilterWheelsRequest"></wsdl:input>
    <wsdl:output message="rti:camGetFilterWheelsResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="camGetFocuser">
    <wsdl:documentation>
      Returns the focuser identifier associated to the camera. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier @return Identifier @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:camGetFocuserRequest"></wsdl:input>
    <wsdl:output message="rti:camGetFocuserResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="domGetNumberElement">
    <wsdl:documentation>
      Returns the number (N) of element of the dome. @param session Session token if
      the used login method is based on session token type. @param deviceId Device
      identifier. @return Number of elements. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:domGetNumberElementRequest"></wsdl:input>
    <wsdl:output message="rti:domGetNumberElementResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="domCanSetAltitude">
    <wsdl:documentation>
      Returns true if it is capable of setting dome altitude. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return boolean value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:domCanSetAltitudeRequest"></wsdl:input>
    <wsdl:output message="rti:domCanSetAltitudeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="domCanSetAzimuth">
    <wsdl:documentation>
      Returns true if it is capable of setting dome azimuth. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return Boolean value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:domCanSetAzimuthRequest"></wsdl:input>
    <wsdl:output message="rti:domCanSetAzimuthResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="domCanSetPark">
    <wsdl:documentation>
      Returns true if it is capable of setting dome park position. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return Boolean value. @throws RtiError In error
      case.
    </wsdl:documentation>
```

```
  <wsdl:input message="rti:domCanSetParkRequest"></wsdl:input>
  <wsdl:output message="rti:domCanSetParkResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domIsAtHome">
  <wsdl:documentation>
    Returns true if the dome is in its home position. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domIsAtHomeRequest"></wsdl:input>
  <wsdl:output message="rti:domIsAtHomeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domIsAtPark">
  <wsdl:documentation>
    Returns true if the dome is in its park position. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domIsAtParkRequest"></wsdl:input>
  <wsdl:output message="rti:domIsAtParkResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domGetAltitude">
  <wsdl:documentation>
    Returns the altitude position of the dome. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Altitude. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domGetAltitudeRequest"></wsdl:input>
  <wsdl:output message="rti:domGetAltitudeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domGetAzimuth">
  <wsdl:documentation>
    Returns the azimuth position of the dome. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Azimuth. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domGetAzimuthRequest"></wsdl:input>
  <wsdl:output message="rti:domGetAzimuthResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domOpen">
  <wsdl:documentation>
    This method will open the n_element of the dome. For this, the state must be
    CLOSE. @param session Session token if the used login method is based on session
    token type. @param deviceId Device identifier. @param element The element
    (order) to open. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domOpenRequest"></wsdl:input>
  <wsdl:output message="rti:domOpenResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domClose">
  <wsdl:documentation>
    This method will close the n_element of the dome. For this, the state must be
    OPEN. @param session Session token if the used login method is based on session
    token type. @param deviceId Device identifier. @param element The element
    (order) to close. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domCloseRequest"></wsdl:input>
  <wsdl:output message="rti:domCloseResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="domGoHome">
  <wsdl:documentation>
    This method will move the dome to the home position. If an error happens (i.e.
    electrical, motor...), this method will be used to move the dome in a known
    position. @param session Session token if the used login method is based on
    session token type. @param deviceId Device identifier. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:domGoHomeRequest"></wsdl:input>
  <wsdl:output message="rti:domGoHomeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domSetPark">
  <wsdl:documentation>
    This method will establish the dome park position. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @param altitude Altitude value. @param azimuth Azimuth value.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domSetParkRequest"></wsdl:input>
  <wsdl:output message="rti:domSetParkResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domPark">
  <wsdl:documentation>
    Moves dome to park position. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domParkRequest"></wsdl:input>
  <wsdl:output message="rti:domParkResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domMoveAzimuth">
  <wsdl:documentation>
    This method will move the dome to the indicated azimuth. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @param azimuth Azimuth value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domMoveAzimuthRequest"></wsdl:input>
  <wsdl:output message="rti:domMoveAzimuthResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domMoveAltitude">
  <wsdl:documentation>
    This method will move the dome to the indicated altitude. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @param altitude Altitude value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:domMoveAltitudeRequest"></wsdl:input>
  <wsdl:output message="rti:domMoveAltitudeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domSetTracking">
  <wsdl:documentation>
    Access method to the state (on/off) of the mount tracking drive.. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return value If true the tracking is active @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domSetTrackingRequest"></wsdl:input>
  <wsdl:output message="rti:domSetTrackingResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="domGetTracking">
  <wsdl:documentation>
    Access method to the state (on/off) of the mount tracking drive.. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @param value True to activate the tracking @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domGetTrackingRequest"></wsdl:input>
  <wsdl:output message="rti:domGetTrackingResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="domSlewObject">
  <wsdl:documentation>
    Slew to an object @param session Session token if the used login method is based
    on session token type. @param deviceId Device identifier. @param object @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:domSlewObjectRequest"></wsdl:input>
  <wsdl:output message="rti:domSlewObjectResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetUtcClock">
  <wsdl:documentation>
    Returns the UTC date/time of the telescope's internal clock. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Date Value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetUtcClockRequest"></wsdl:input>
  <wsdl:output message="rti:mntGetUtcClockResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntSetUtcClock">
  <wsdl:documentation>
    Sets the UTC date/time of the telescope's internal clock. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @param date New date. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntSetUtcClockRequest"></wsdl:input>
  <wsdl:output message="rti:mntSetUtcClockResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetSiderealDate">
  <wsdl:documentation>
    The local apparent sidereal time from the telescope's internal clock (hours,
    sidereal). Local Apparent Sidereal Time is the sidereal time used for pointing
    telescopes, and thus must be calculated from the Greenwich Mean Sidereal time,
    longitude, nutation in longitude and true ecliptic obliquity. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Double Value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetSiderealDateRequest"></wsdl:input>
  <wsdl:output message="rti:mntGetSiderealDateResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntIsAtHome">
  <wsdl:documentation>
    Returns true if the mount is stopped in the home position. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Boolean Value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntIsAtHomeRequest"></wsdl:input>
  <wsdl:output message="rti:mntIsAtHomeResponse"></wsdl:output>
```

```
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntIsParked">
    <wsdl:documentation>
      Returns true if the mount is stopped in the parked position. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return Boolean Value. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntIsParkedRequest"></wsdl:input>
    <wsdl:output message="rti:mntIsParkedResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetTargetRightAscension">
    <wsdl:documentation>
      The right ascension for the target of an equatorial slew operation. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device identifier. @return Double target right ascension.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetTargetRightAscensionRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetTargetRightAscensionResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetTargetDeclination">
    <wsdl:documentation>
      The declination for the target of an equatorial slew operation. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return Double target declination. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetTargetDeclinationRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetTargetDeclinationResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetTrackingDeclinationRate">
    <wsdl:documentation>
      Returns the Declination tracking rate. @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @return Double declination tracking rate. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetTrackingDeclinationRateRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetTrackingDeclinationRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetTrackingAscensionRate">
    <wsdl:documentation>
      Returns True if the camera can abort exposures. False if not. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return double Ascension rate. @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetTrackingAscensionRateRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetTrackingAscensionRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetTrackingRate">
    <wsdl:documentation>
      Returns the current tracking rate of the mount. @param session Session token if
      the used login method is based on session token type. @param deviceId Device
      identifier. @return int The tracking rate. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetTrackingRateRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetTrackingRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="mntSetTrackingRate">
  <wsdl:documentation>
    Sets the current tracking rate of the mount. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @param rateThe tracking rate. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntSetTrackingRateRequest"></wsdl:input>
  <wsdl:output message="rti:mntSetTrackingRateResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetTracking">
  <wsdl:documentation>
    Access method to the state (on/off) of the telescope's sidereal tracking drive.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Boolean value. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetTrackingRequest"></wsdl:input>
  <wsdl:output message="rti:mntGetTrackingResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntSetTracking">
  <wsdl:documentation>
    Access method to the state (on/off) of the telescope's sidereal tracking drive.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @param value New Value. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntSetTrackingRequest"></wsdl:input>
  <wsdl:output message="rti:mntSetTrackingResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetGuideRateDeclination">
  <wsdl:documentation>
    Returns the current Declination movement rate offset for telescope guiding
    (degrees/sec). @param session Session token if the used login method is based on
    session token type. @param deviceId Device identifier. @return the declination
    movement rate offset. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetGuideRateDeclinationRequest"></wsdl:input>
  <wsdl:output message="rti:mntGetGuideRateDeclinationResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetDeclinationRateRightAscension">
  <wsdl:documentation>
    Returns the current Right Ascension movement rate offset for telescope guiding
    (degrees/sec). @param session Session token if the used login method is based on
    session token type. @param deviceId Device identifier. @return The right
    ascension movement rate offset. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetDeclinationRateRightAscensionRequest">
  </wsdl:input>
  <wsdl:output message="rti:mntGetDeclinationRateRightAscensionResponse">
  </wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntIsSlewing">
  <wsdl:documentation>
    Returns True if the camera can abort exposures. False if not. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Boolean value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntIsSlewingRequest"></wsdl:input>
```

```
  <wsdl:output message="rti:mntIsSlewingResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetPosAxis2">
  <wsdl:documentation>
    Returns the Azimuth current position. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return AZ Position. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetPosAxis2Request"></wsdl:input>
  <wsdl:output message="rti:mntGetPosAxis2Response"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetPosAxis3">
  <wsdl:documentation>
    Returns Image rotator/de-rotator current position. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @return ROT Position. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetPosAxis3Request"></wsdl:input>
  <wsdl:output message="rti:mntGetPosAxis3Response"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetPosAxis1">
  <wsdl:documentation>
    Returns the Altitude current position. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return ALT Position. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntGetPosAxis1Request"></wsdl:input>
  <wsdl:output message="rti:mntGetPosAxis1Response"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntCanPulseGuide">
  <wsdl:documentation>
    Returns true if this property can be set. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntCanPulseGuideRequest"></wsdl:input>
  <wsdl:output message="rti:mntCanPulseGuideResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntCanSetGuideRates">
  <wsdl:documentation>
    True if the guide rate properties used for PulseGuide method can be adjusted.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Boolean value. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntCanSetGuideRatesRequest"></wsdl:input>
  <wsdl:output message="rti:mntCanSetGuideRatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntCanSetPark">
  <wsdl:documentation>
    Returns true if this property can be set. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntCanSetParkRequest"></wsdl:input>
  <wsdl:output message="rti:mntCanSetParkResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntSetPark">
```

```
    <wsdl:documentation>
      Set the parked position in equatorial coordinates @param session Session token
      if the used login method is based on session token type. @param deviceId Device
      identifier. @param ascension Angle. @param declination Angle. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntSetParkRequest"></wsdl:input>
    <wsdl:output message="rti:mntSetParkResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetALTParkPos">
    <wsdl:documentation>
      Return the parked position ALT coordinate. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return Parked RA Position. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetALTParkPosRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetALTParkPosResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetAZParkPos">
    <wsdl:documentation>
      Return the parked position AZ coordinate. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return Parked DEC Position. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetAZParkPosRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetAZParkPosResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSetTracking">
    <wsdl:documentation>
      Returns true if the sidereal drive can be change to on/off value. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return Boolean value. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSetTrackingRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSetTrackingResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSetTrackingRate">
    <wsdl:documentation>
      Returns true if this property can be set. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return Boolean value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSetTrackingRateRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSetTrackingRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewCoordinates">
    <wsdl:documentation>
      Returns true if this telescope is capable of programmed slewing (synchronous) to
      equatorial. @param session Session token if the used login method is based on
      session token type. @param deviceId Device identifier. @return Boolean value.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSlewCoordinatesRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSlewCoordinatesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewCoordinatesAsync">
    <wsdl:documentation>
      Returns true if the mount is capable of programmed slewing (asynchronous) to
```

```
      equatorial. @param session Session token if the used login method is based on
      session token type. @param deviceId Device identifier. @return Boolean value.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSlewCoordinatesAsyncRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSlewCoordinatesAsyncResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewObject">
    <wsdl:documentation>
      Return true if the mount is capable of slewing to an object. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return Boolean value. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSlewObjectRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSlewObjectResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewAltAz">
    <wsdl:documentation>
      Returns true if the mount is capable of programmed synchronous slewing to local
      horizontal coordinates. @param session Session token if the used login method is
      based on session token type. @param deviceId Device identifier. @return Boolean
      value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSlewAltAzRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSlewAltAzResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewAzAsync">
    <wsdl:documentation>
      Returns true if the mount is capable of programmed asynchronous slewing to local
      horizontal coordinates. @param session Session token if the used login method is
      based on session token type. @param deviceId Device identifier. @return Boolean
      value. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanSlewAzAsyncRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanSlewAzAsyncResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntCanMoveAzis">
    <wsdl:documentation>
      Returns True if the mount can be controlled about the specified axis via. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device identifier. @return Boolean value. @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntCanMoveAzisRequest"></wsdl:input>
    <wsdl:output message="rti:mntCanMoveAzisResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntAxisRate">
    <wsdl:documentation>
      Returns a collection of Rate objects describing the supported rates of motion.
      @param session Session token if the used login method is based on session token
      type. @param deviceId Device identifier. @return List of AxisRateType. @throws
      RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntAxisRateRequest"></wsdl:input>
    <wsdl:output message="rti:mntAxisRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntTrackingRates">
    <wsdl:documentation>
```

```
      Returns a collection of Rate values of the TrackingRate property. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return List of TrackingRateType. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntTrackingRatesRequest"></wsdl:input>
    <wsdl:output message="rti:mntTrackingRatesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGoHome">
    <wsdl:documentation>
      Moves the mount to the home position (synchronous). @param session Session token
      if the used login method is based on session token type. @param deviceId Device
      identifier. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGoHomeRequest"></wsdl:input>
    <wsdl:output message="rti:mntGoHomeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntPark">
    <wsdl:documentation>
      Moves the mount to the parked position and fix the state to PARKED. @param
      session Session token if the used login method is based on session token type.
      @param deviceId Device identifier. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntParkRequest"></wsdl:input>
    <wsdl:output message="rti:mntParkResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntUnpark">
    <wsdl:documentation>
      Moves the mount to a ready position. Before call Unpark(), the mount state must
      have fixed to READY. @param session Session token if the used login method is
      based on session token type. @param deviceId Device identifier. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntUnparkRequest"></wsdl:input>
    <wsdl:output message="rti:mntUnparkResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToAltAz">
    <wsdl:documentation>
      This synchronous method moves the mount to the given local horizontal
      coordinates. This Method must be implemented if CanSlewAltAz returns True.
      Raises an error if the slew fails. @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @param azimuth Azimuth value. @param altitude Altitude value. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntSlewToAltAzRequest"></wsdl:input>
    <wsdl:output message="rti:mntSlewToAltAzResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToAltAzAsync">
    <wsdl:documentation>
      Asynchronous SlewToAltAz method. @param session Session token if the used login
      method is based on session token type. @param deviceId Device identifier. @param
      azimuth Azimuth value. @param altitude Altitude value. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntSlewToAltAzAsyncRequest"></wsdl:input>
    <wsdl:output message="rti:mntSlewToAltAzAsyncResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToCoordinates">
```

```
  <wsdl:documentation>
    This synchronous method moves the mount to the given ecuatorial coordinates.
    This Method must be implemented if CanSlewCoordinates returns True. Raises an
    error if the slew fails. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier. @param
    ascension Angle. @param declination Angle. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntSlewToCoordinatesRequest"></wsdl:input>
  <wsdl:output message="rti:mntSlewToCoordinatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntSlewToCoordinatesAsync">
  <wsdl:documentation>
    Asynchronous SlewToCoordinates method. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @param ascension Angle. @param declination Angle. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntSlewToCoordinatesAsyncRequest"></wsdl:input>
  <wsdl:output message="rti:mntSlewToCoordinatesAsyncResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntMoveAxis">
  <wsdl:documentation>
    The mount will start moving at the specified rate about the specified axis and
    continue indefinitely following the rate of motion. This must be implemented if
    the CanMoveAxis property returns True for the given axis. The movement rate must
    be within the allowed values (AxisRate method). @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @param axisType 0-Primary axis (e.g., Right Ascension or Azimuth),
    1-Secondary axis (e.g., Declination or Altitude), 2-Tertiary axis (e.g. imager
    rotator/de-rotator). @param rate The rate of motion (deg/sec, + = clockwise)
    about the specified axis. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntMoveAxisRequest"></wsdl:input>
  <wsdl:output message="rti:mntMoveAxisResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntPulseGuide">
  <wsdl:documentation>
    Moves the scope in the given direction for the given interval or time at the
    rate given by the corresponding guide rate property. @param session Session
    token if the used login method is based on session token type. @param deviceId
    Device identifier. @param guideDirection North-0, South-1, East-2, West-3 @param
    duration Duration of the movement. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntPulseGuideRequest"></wsdl:input>
  <wsdl:output message="rti:mntPulseGuideResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntStopSlew">
  <wsdl:documentation>
    To stop the movement in the axis parameter. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @param axisType 0-Primary axis (e.g., Right Ascension or Azimuth),
    1-Secondary axis (e.g., Declination or Altitude), 2-Tertiary axis (e.g. imager
    rotator/de-rotator). @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:mntStopSlewRequest"></wsdl:input>
  <wsdl:output message="rti:mntStopSlewResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntStopSlewAxis">
  <wsdl:documentation>
    To stop all movements. @param session Session token if the used login method is
```

based on session token type. @param deviceId Device identifier. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntStopSlewAxisRequest"></wsdl:input>
<wsdl:output message="rti:mntStopSlewAxisResponse"></wsdl:output>
<wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntGetPointingModel">
<wsdl:documentation>
Returns the pointing model of a mount. @param session Session token if the used login method is based on session token type. @param deviceId Device identifier. @return The pointing model. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntGetPointingModelRequest"></wsdl:input>
<wsdl:output message="rti:mntGetPointingModelResponse"></wsdl:output>
<wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntMoveNorth">
<wsdl:documentation>
The mount moves North during a period of time @param session Session token if the used login method is based on session token type. @param deviceId Device identifier. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntMoveNorthRequest"></wsdl:input>
<wsdl:output message="rti:mntMoveNorthResponse"></wsdl:output>
<wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntMoveSouth">
<wsdl:documentation>
The mount moves South during a period of time @param session Session token if the used login method is based on session token type. @param deviceId Device identifier. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntMoveSouthRequest"></wsdl:input>
<wsdl:output message="rti:mntMoveSouthResponse"></wsdl:output>
<wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntMoveEast">
<wsdl:documentation>
The mount moves East during a period of time @param session Session token if the used login method is based on session token type. @param deviceId Device identifier. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntMoveEastRequest"></wsdl:input>
<wsdl:output message="rti:mntMoveEastResponse"></wsdl:output>
<wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntMoveWest">
<wsdl:documentation>
The mount moves West during a period of time @param session Session token if the used login method is based on session token type. @param deviceId Device identifier. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntMoveWestRequest"></wsdl:input>
<wsdl:output message="rti:mntMoveWestResponse"></wsdl:output>
<wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="mntSetSlewRate">
<wsdl:documentation>
Sets the slew rate. * SLEW_RATE (faster) * MOVE_RATE (second faster) * GUIDE_RATE (slowest) * CENTER_RATE(second lowest) @param session Session token if the used login method is based on session token type. @param deviceId Device identifier. @param rate String. @throws RtiError In error case.
</wsdl:documentation>
<wsdl:input message="rti:mntSetSlewRateRequest"></wsdl:input>

```
    <wsdl:output message="rti:mntSetSlewRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntGetSlewRate">
    <wsdl:documentation>
      Gets the slew rate. * SLEW_RATE (faster) * MOVE_RATE (second faster) *
      GUIDE_RATE (slowest) * CENTER_RATE(second lowest) @param session Session token
      if the used login method is based on session token type. @param deviceId Device
      identifier. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntGetSlewRateRequest"></wsdl:input>
    <wsdl:output message="rti:mntGetSlewRateResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntSlewObject">
    <wsdl:documentation>
      Slew to an object @param session Session token if the used login method is based
      on session token type. @param deviceId Device identifier. @param object Object
      to slew to. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntSlewObjectRequest"></wsdl:input>
    <wsdl:output message="rti:mntSlewObjectResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntIsPointingAtObject">
    <wsdl:documentation>
      Check if the mount is pointing at Object. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @param object Object to slew to. @param raError Allowed error in RA
      coordinate (Hours) @param decError Allowed error in DEC coordinate (Degrees)
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntIsPointingAtObjectRequest"></wsdl:input>
    <wsdl:output message="rti:mntIsPointingAtObjectResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="mntIsPointingAtCoordinates">
    <wsdl:documentation>
      Check if the mount is pointing at proper coordinates. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @param ra Proper RA coordinate (Hours) @param dec Proper DEC
      coordinate (Degrees) @param raError Allowed error in RA coordinate (Hours)
      @param decError Allowed error in DEC coordinate (Degrees) @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:mntIsPointingAtCoordinatesRequest"></wsdl:input>
    <wsdl:output message="rti:mntIsPointingAtCoordinatesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="focGetCamera">
    <wsdl:documentation>
      Retrieves the camera identifier where the focuser is connected. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return String @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:focGetCameraRequest"></wsdl:input>
    <wsdl:output message="rti:focGetCameraResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="focIsAbsolute">
    <wsdl:documentation>
      True if the focuser is capable of absolute position. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return boolean @throws RtiError In error case.
    </wsdl:documentation>
```

```
  <wsdl:input message="rti:focIsAbsoluteRequest"></wsdl:input>
  <wsdl:output message="rti:focIsAbsoluteResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focGetStepSize">
  <wsdl:documentation>
    Step size (microns) for the focuser. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return size. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focGetStepSizeRequest"></wsdl:input>
  <wsdl:output message="rti:focGetStepSizeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focGetMaxIncrement">
  <wsdl:documentation>
    Maximum increment size in one move operation allowed by the focuser. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @return Increment. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:focGetMaxIncrementRequest"></wsdl:input>
  <wsdl:output message="rti:focGetMaxIncrementResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focGetMaxStep">
  <wsdl:documentation>
    Maximum step position permitted. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return Step. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focGetMaxStepRequest"></wsdl:input>
  <wsdl:output message="rti:focGetMaxStepResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focGetMinStep">
  <wsdl:documentation>
    Minimum step position permitted. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return Step. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focGetMinStepRequest"></wsdl:input>
  <wsdl:output message="rti:focGetMinStepResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focGetPosition">
  <wsdl:documentation>
    Current focuser position, in steps. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return position. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focGetPositionRequest"></wsdl:input>
  <wsdl:output message="rti:focGetPositionResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focIsTempCompAvailable">
  <wsdl:documentation>
    Its value is True if focuser has temperature compensation available. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @return boolean. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:focIsTempCompAvailableRequest"></wsdl:input>
  <wsdl:output message="rti:focIsTempCompAvailableResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="focGetTemperature">
  <wsdl:documentation>
    Current ambient temperature as measured by the focuser. Raises an exception if
    ambient temperature is not available. Commonly available on focusers with a
    built-in temperature compensation mode. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return Temperature. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focGetTemperatureRequest"></wsdl:input>
  <wsdl:output message="rti:focGetTemperatureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focSetTempComp">
  <wsdl:documentation>
    If the TempCompAvailable property is True, then setting TempComp to True puts
    the focuser into temperature tracking mode. Set to False to turn off temperature
    tracking. While focuser is in temperature tracking mode, Move commands will be
    rejected. An exception will be raised if focIsTempCompAvailable is False and an
    attempt is made to set TempComp to true. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @param trackingMode tracking mode. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focSetTempCompRequest"></wsdl:input>
  <wsdl:output message="rti:focSetTempCompResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focHalt">
  <wsdl:documentation>
    It stops any previous focuser motion. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focHaltRequest"></wsdl:input>
  <wsdl:output message="rti:focHaltResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="focMove">
  <wsdl:documentation>
    Moves the focuser by the specified amount or to the specified position depending
    on the value of the Absolute property. If the Absolute property is True, then
    this is an absolute positioning focuser. The Move command tells the focuser to
    move to an exact step position, and the Position parameter of the Move method is
    an integer between 0 and MaxStep (0 greater or equal to position less or equal
    to MaxStep). If the Absolute property is False, then this is a relative
    positioning focuser. The Move command tells the focuser to move in a relative
    direction, and the Position parameter of the Move method (in this case, step
    distance) is an integer between minus MaxIncrement and plus MaxIncrement (-
    MaxIncrement greater or equal to position less or equal to MaxIncrement) @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @param position The target position of the
    movement. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:focMoveRequest"></wsdl:input>
  <wsdl:output message="rti:focMoveResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="dwrGetChannelsNumber">
  <wsdl:documentation>
    Returns the channels number. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return Channel number. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:dwrGetChannelsNumberRequest"></wsdl:input>
  <wsdl:output message="rti:dwrGetChannelsNumberResponse"></wsdl:output>
```

```
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="dwrGetChannelType">
  <wsdl:documentation>
    Returns the type of a channel (duty cycle, temperature controlled). @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @param channel channel id (order). @return
    Type. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:dwrGetChannelTypeRequest"></wsdl:input>
  <wsdl:output message="rti:dwrGetChannelTypeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="dwrSetTemperatureThreshold">
  <wsdl:documentation>
    Set the temperature threshold for the selected channel. If this channel is not
    temperature controlled an exception will be raised. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @param channel Channel id (order). @param temperature threshold.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:dwrSetTemperatureThresholdRequest"></wsdl:input>
  <wsdl:output message="rti:dwrSetTemperatureThresholdResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="dwrSetCycleThreshold">
  <wsdl:documentation>
    Set the duty cycle for the channel introduced as parameter. If this channel is
    temperature controlled an exception will be raised. @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @param channel Channel id (order). @param cycle duty cycle. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:dwrSetCycleThresholdRequest"></wsdl:input>
  <wsdl:output message="rti:dwrSetCycleThresholdResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fhtGetMeasureUnit">
  <wsdl:documentation>
    Return the measure unit of the sensor. A/V if a photodiode is used; Ohms is a
    photoresistor is used... @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier. @return
    MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fhtGetMeasureUnitRequest"></wsdl:input>
  <wsdl:output message="rti:fhtGetMeasureUnitResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fhtGetMeasure">
  <wsdl:documentation>
    Returns the measure of the sensor. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fhtGetMeasureRequest"></wsdl:input>
  <wsdl:output message="rti:fhtGetMeasureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fhtSetMeasureStates">
  <wsdl:documentation>
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
```

```
  <wsdl:input message="rti:fhtSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:fhtSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fhtGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fhtGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:fhtGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rndGetMeasureUnit">
  <wsdl:documentation>
    Returns the measure units of the sensor. Cycles if a capacitor is used; Inches
    is a hygroscopic disc is used. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:rndGetMeasureUnitRequest"></wsdl:input>
  <wsdl:output message="rti:rndGetMeasureUnitResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rndGetMeasure">
  <wsdl:documentation>
    Returns the measure of the sensor. Cycles if a capacitor is used; Inches is a
    hygroscopic disc is used. @param session Session token if the used login method
    is based on session token type. @param deviceId Device identifier. @return
    MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:rndGetMeasureRequest"></wsdl:input>
  <wsdl:output message="rti:rndGetMeasureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rndSetMeasureStates">
  <wsdl:documentation>
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:rndSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:rndSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rndGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:rndGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:rndGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rndIsRaining">
  <wsdl:documentation>
    Returns true if it is raining. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:rndIsRainingRequest"></wsdl:input>
  <wsdl:output message="rti:rndIsRainingResponse"></wsdl:output>
```

```
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="rhsGetMeasureUnit">
    <wsdl:documentation>
      Returns the measure units of the sensor (%). @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return MeasureUnit. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:rhsGetMeasureUnitRequest"></wsdl:input>
    <wsdl:output message="rti:rhsGetMeasureUnitResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="rhsGetMeasure">
    <wsdl:documentation>
      Returns the measure of the sensor. @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @return MeasureUnit. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:rhsGetMeasureRequest"></wsdl:input>
    <wsdl:output message="rti:rhsGetMeasureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="rhsSetMeasureStates">
    <wsdl:documentation>
      Sets the measure interval states. @param session Session token if the used login
      method is based on session token type. @param deviceId Device identifier. @param
      states State (measure interval and alarm activation). @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:rhsSetMeasureStatesRequest"></wsdl:input>
    <wsdl:output message="rti:rhsSetMeasureStatesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="rhsGetMeasureStates">
    <wsdl:documentation>
      Returns the list of the measure interval states. @param session Session token if
      the used login method is based on session token type. @param deviceId Device
      identifier. @return List of states information. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:rhsGetMeasureStatesRequest"></wsdl:input>
    <wsdl:output message="rti:rhsGetMeasureStatesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="barGetMeasureUnit">
    <wsdl:documentation>
      Returns the measure units of the sensor (Pascal). @param session Session token
      if the used login method is based on session token type. @param deviceId Device
      identifier. @return MeasureUnit. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:barGetMeasureUnitRequest"></wsdl:input>
    <wsdl:output message="rti:barGetMeasureUnitResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="barGetMeasure">
    <wsdl:documentation>
      Returns the measure of the sensor. @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @return MeasureUnit. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:barGetMeasureRequest"></wsdl:input>
    <wsdl:output message="rti:barGetMeasureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="barSetMeasureStates">
    <wsdl:documentation>
```

```
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:barSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:barSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="barGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:barGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:barGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="tempGetMeasureUnit">
  <wsdl:documentation>
    Returns the measure units of the sensor (Kelvin, Celsius). @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:tempGetMeasureUnitRequest"></wsdl:input>
  <wsdl:output message="rti:tempGetMeasureUnitResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="tempGetMeasure">
  <wsdl:documentation>
    Returns the measure of the sensor. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:tempGetMeasureRequest"></wsdl:input>
  <wsdl:output message="rti:tempGetMeasureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="tempSetMeasureStates">
  <wsdl:documentation>
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:tempSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:tempSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="tempGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:tempGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:tempGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="strIsAvailableOrientation">
  <wsdl:documentation>
    Returns true if the the device can provide the storm orientation. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return Boolean value. @throws RtiError In error
```

```
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:strIsAvailableOrientationRequest"></wsdl:input>
    <wsdl:output message="rti:strIsAvailableOrientationResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="strGetDistanceMeasureUnit">
    <wsdl:documentation>
      Returns the measure unit of the sensor (distance: MILES, KM). @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return MeasureUnit. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:strGetDistanceMeasureUnitRequest"></wsdl:input>
    <wsdl:output message="rti:strGetDistanceMeasureUnitResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="strGetDistanceMeasure">
    <wsdl:documentation>
      Returns the measure of the sensor distance:(MILES, KM). @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return Measure. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:strGetDistanceMeasureRequest"></wsdl:input>
    <wsdl:output message="rti:strGetDistanceMeasureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="strGetDegrees">
    <wsdl:documentation>
      Returns the relative storm orientation. @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @return degrees. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:strGetDegreesRequest"></wsdl:input>
    <wsdl:output message="rti:strGetDegreesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="strGetAbosoluteDegrees">
    <wsdl:documentation>
      Returns the absolute storm orientation (Configuration). @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return degrees. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:strGetAbosoluteDegreesRequest"></wsdl:input>
    <wsdl:output message="rti:strGetAbosoluteDegreesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="strSetDistanceMeasureStates">
    <wsdl:documentation>
      Sets the distance measure interval states. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @param states State (measure interval and alarm activation). @throws
      RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:strSetDistanceMeasureStatesRequest"></wsdl:input>
    <wsdl:output message="rti:strSetDistanceMeasureStatesResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="strGetDistanceMeasureStates">
    <wsdl:documentation>
      Returns the list of the distance measure interval states. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return List of states information. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:strGetDistanceMeasureStatesRequest"></wsdl:input>
```

```
  <wsdl:output message="rti:strGetDistanceMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="strSetOrientationMeasureStates">
  <wsdl:documentation>
    Sets the orientation measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @param states State (measure interval and alarm activation). @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:strSetOrientationMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:strSetOrientationMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="strGetOrientationMeasureStates">
  <wsdl:documentation>
    Returns the list of the orientation measure interval states. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return List of states information. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:strGetOrientationMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:strGetOrientationMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wvnGetMeasureUnit">
  <wsdl:documentation>
    Returns the measure units of the sensor (DEGREES). @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wvnGetMeasureUnitRequest"></wsdl:input>
  <wsdl:output message="rti:wvnGetMeasureUnitResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wvnGetMeasure">
  <wsdl:documentation>
    Returns the measure of the sensor. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wvnGetMeasureRequest"></wsdl:input>
  <wsdl:output message="rti:wvnGetMeasureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wvnGetAbosluteDegrees">
  <wsdl:documentation>
    Is the absolute orientation (Device configuration). @param session Session token
    if the used login method is based on session token type. @param deviceId Device
    identifier. @return degrees. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wvnGetAbosluteDegreesRequest"></wsdl:input>
  <wsdl:output message="rti:wvnGetAbosluteDegreesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wvnSetMeasureStates">
  <wsdl:documentation>
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:wvnSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:wvnSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
```

```
</wsdl:operation>
<wsdl:operation name="wvnGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wvnGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:wvnGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wspGetMeasureUnit">
  <wsdl:documentation>
    Returns the measure units of the sensor. Km/h if a anemometer is used. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @return MeasureUnit. @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wspGetMeasureUnitRequest"></wsdl:input>
  <wsdl:output message="rti:wspGetMeasureUnitResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wspGetMeasure">
  <wsdl:documentation>
    Returns the measure of the sensor. Km/h if a anemometer is used. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wspGetMeasureRequest"></wsdl:input>
  <wsdl:output message="rti:wspGetMeasureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wspSetMeasureStates">
  <wsdl:documentation>
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:wspSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:wspSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="wspGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:wspGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:wspGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="cldGetMeasureUnit">
  <wsdl:documentation>
    Returns the measure units of the sensor. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:cldGetMeasureUnitRequest"></wsdl:input>
  <wsdl:output message="rti:cldGetMeasureUnitResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="cldGetMeasure">
  <wsdl:documentation>
```

```
    Returns the measure of the sensor. Km/h if a anemometer is used. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @return MeasureUnit. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:cldGetMeasureRequest"></wsdl:input>
  <wsdl:output message="rti:cldGetMeasureResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="cldSetMeasureStates">
  <wsdl:documentation>
    Sets the measure interval states. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    states State (measure interval and alarm activation). @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:cldSetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:cldSetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="cldGetMeasureStates">
  <wsdl:documentation>
    Returns the list of the measure interval states. @param session Session token if
    the used login method is based on session token type. @param deviceId Device
    identifier. @return List of states information. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:cldGetMeasureStatesRequest"></wsdl:input>
  <wsdl:output message="rti:cldGetMeasureStatesResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fwGetCamera">
  <wsdl:documentation>
    Retrieves the camera identifier where the filter wheel is connected. @param
    session Session token if the used login method is based on session token type.
    @param deviceId Device identifier. @return Camera identifier. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fwGetCameraRequest"></wsdl:input>
  <wsdl:output message="rti:fwGetCameraResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fwGetFilterList">
  <wsdl:documentation>
    Retrieves the list of filter in the wheel. The list position corresponds with
    the wheel position. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @return filter
    list. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fwGetFilterListRequest"></wsdl:input>
  <wsdl:output message="rti:fwGetFilterListResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fwGetPositionNumber">
  <wsdl:documentation>
    Number of filter position in wheel. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return Number of the possible positions. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fwGetPositionNumberRequest"></wsdl:input>
  <wsdl:output message="rti:fwGetPositionNumberResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fwGetSpeedSwitching">
  <wsdl:documentation>
    Speed filter switching (ms). @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
```

```
      @return Speed switching (ms). @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:fwGetSpeedSwitchingRequest"></wsdl:input>
    <wsdl:output message="rti:fwGetSpeedSwitchingResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="fwGetFilterSize">
    <wsdl:documentation>
      Retrieves the filter inches (1.25'', 2''...) @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return inches. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:fwGetFilterSizeRequest"></wsdl:input>
    <wsdl:output message="rti:fwGetFilterSizeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="fwGetFilterKind">
    <wsdl:documentation>
      Retrieves the kind of filter selected @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @return Kind (G, R, Z, clear...). @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:fwGetFilterKindRequest"></wsdl:input>
    <wsdl:output message="rti:fwGetFilterKindResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="fwIsAtHome">
    <wsdl:documentation>
      Returns true if the wheel is stopped in the home position. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @return boolean value. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:fwIsAtHomeRequest"></wsdl:input>
    <wsdl:output message="rti:fwIsAtHomeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="fwSetOffset">
    <wsdl:documentation>
      Focus offset in each wheel position. These values are focuser and filter
      dependent. @param session Session token if the used login method is based on
      session token type. @param deviceId Device identifier. @param positions A list
      of wheel position, one per wheel belonging to the device. @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:fwSetOffsetRequest"></wsdl:input>
    <wsdl:output message="rti:fwSetOffsetResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="fwSelectFilterKind">
    <wsdl:documentation>
      Select the filter, by filter kind, to switch to. @param session Session token if
      the used login method is based on session token type. @param deviceId Device
      identifier. @param kind Wheel Kind. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:fwSelectFilterKindRequest"></wsdl:input>
    <wsdl:output message="rti:fwSelectFilterKindResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="fwSelectFilterPosition">
    <wsdl:documentation>
      Select filter, by wheel position, to switch to. @param session Session token if
      the used login method is based on session token type. @param deviceId Device
      identifier. @param position Position within the wheel (1..N). @throws RtiError
      In error case.
```

```
  </wsdl:documentation>
  <wsdl:input message="rti:fwSelectFilterPositionRequest"></wsdl:input>
  <wsdl:output message="rti:fwSelectFilterPositionResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="fwGoHome">
  <wsdl:documentation>
    Switch the wheel to the home position. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:fwGoHomeRequest"></wsdl:input>
  <wsdl:output message="rti:fwGoHomeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rttGetCurrentPosition">
  <wsdl:documentation>
    Returns the rotator current position. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return Current position. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:rttGetCurrentPositionRequest"></wsdl:input>
  <wsdl:output message="rti:rttGetCurrentPositionResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="rttSetTargetPosition">
  <wsdl:documentation>
    Sets the rotator target position. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier. @param
    position New position @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:rttSetTargetPositionRequest"></wsdl:input>
  <wsdl:output message="rti:rttSetTargetPositionResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamHasBrightness">
  <wsdl:documentation>
    If True, the camera's brightness can be adjusted. If False, the camera can not
    adjust the brightness. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @return Boolean
    value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamHasBrightnessRequest"></wsdl:input>
  <wsdl:output message="rti:scamHasBrightnessResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamHasContrast">
  <wsdl:documentation>
    If True, the camera's contrast can be adjusted. If False, the camera can not
    adjust the contrast. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @return Boolean
    value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamHasContrastRequest"></wsdl:input>
  <wsdl:output message="rti:scamHasContrastResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamAcquisitionMode">
  <wsdl:documentation>
    If it is Continuous mode, the camera will take a video streaming. If it is
    OneShot mode, only one image will be taken. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Continuous, OneShot. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamAcquisitionModeRequest"></wsdl:input>
```

```
  <wsdl:output message="rti:scamAcquisitionModeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetFPS">
  <wsdl:documentation>
    Frames Per Second. It only takes effect in the “Continuous” Acquisition Mode.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier @return FPS. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetFPSRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetFPSResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetDigitizingMode">
  <wsdl:documentation>
    If it is Automatic mode, the camera will take images according to the
    Brightness, Contrast, Gamma and Gain values calculated automatically by the
    camera itself. If it is Manual mode, the camera will take images according to
    the Brightness, Contrast, Gamma and Gain values established by the variables.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Automatic, Manual. @throws
    RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetDigitizingModeRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetDigitizingModeResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamIsPTSupported">
  <wsdl:documentation>
    Returns if the camera supports Pan and Tilt. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamIsPTSupportedRequest"></wsdl:input>
  <wsdl:output message="rti:scamIsPTSupportedResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamIsZoomSupported">
  <wsdl:documentation>
    Returns if the camera supports Zoom. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return Boolean value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamIsZoomSupportedRequest"></wsdl:input>
  <wsdl:output message="rti:scamIsZoomSupportedResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetPanMin">
  <wsdl:documentation>
    Returns the minimum degrees for the Pan movement. This property will be defined
    if the camera supports Pan and Tilt feature. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return The minimum degrees for the Pan movement. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetPanMinRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetPanMinResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetPanMax">
  <wsdl:documentation>
    Returns the maximum degrees for the Pan movement. This property will be defined
    if the camera supports Pan and Tilt feature. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
```

```
      identifier. @return The maximum degrees for the Pan movement. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamGetPanMaxRequest"></wsdl:input>
    <wsdl:output message="rti:scamGetPanMaxResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamGetTiltMin">
    <wsdl:documentation>
      Returns the minimum degrees for the Tilt movement. This property will be defined
      if the camera supports Pan and Tilt feature. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return The minimum degrees for the Tilt movement. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamGetTiltMinRequest"></wsdl:input>
    <wsdl:output message="rti:scamGetTiltMinResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamGetTiltMax">
    <wsdl:documentation>
      Returns the maximum degrees for the Tilt movement. This property will be defined
      if the camera supports Pan and Tilt feature. @param session Session token if the
      used login method is based on session token type. @param deviceId Device
      identifier. @return The maximum degrees for the Tilt movement. @throws RtiError
      In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamGetTiltMaxRequest"></wsdl:input>
    <wsdl:output message="rti:scamGetTiltMaxResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamGetZoomMax">
    <wsdl:documentation>
      Gets the maximum zoom value allowed. This is the maximum value that applications
      can set. Applications should call ZoomSupported(bool) before using this
      property. Minimun zoom value is zero. @param session Session token if the used
      login method is based on session token type. @param deviceId Device identifier.
      @return The maximum zoom value allowed. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamGetZoomMaxRequest"></wsdl:input>
    <wsdl:output message="rti:scamGetZoomMaxResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamGetExposureTime">
    <wsdl:documentation>
      Gets the value in seconds for the image exposure time. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @return The value in seconds for the image exposure time.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamGetExposureTimeRequest"></wsdl:input>
    <wsdl:output message="rti:scamGetExposureTimeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamSetExposureTime">
    <wsdl:documentation>
      Sets the value in seconds for the image exposure time. @param session Session
      token if the used login method is based on session token type. @param deviceId
      Device identifier. @param value The value in seconds for the image exposure
      time. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamSetExposureTimeRequest"></wsdl:input>
    <wsdl:output message="rti:scamSetExposureTimeResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
```

```
<wsdl:operation name="scamIsImageReady">
  <wsdl:documentation>
    If True, there is an image from the camera available. If False, no image is
    available and attempts to use the ImageArray method will produce an exception.
    @param session Session token if the used login method is based on session token
    type. @param deviceId Device identifier. @return Boolean value. @throws RtiError
    In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamIsImageReadyRequest"></wsdl:input>
  <wsdl:output message="rti:scamIsImageReadyResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetVideoStreamingURL">
  <wsdl:documentation>
    URL for streaming. @param session Session token if the used login method is
    based on session token type. @param deviceId Device identifier. @return URL for
    streaming. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetVideoStreamingURLRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetVideoStreamingURLResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetImageURL">
  <wsdl:documentation>
    Returns the last image URL. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier @return
    String URL @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetImageURLRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetImageURLResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamSetBrightness">
  <wsdl:documentation>
    Sets the value for image brightness. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @param value brightness. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamSetBrightnessRequest"></wsdl:input>
  <wsdl:output message="rti:scamSetBrightnessResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetBrightness">
  <wsdl:documentation>
    Returns the value of image brightness. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @return image brightness. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetBrightnessRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetBrightnessResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamSetContrast">
  <wsdl:documentation>
    Sets the value for image contrast. @param session Session token if the used
    login method is based on session token type. @param deviceId Device identifier.
    @param value New value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamSetContrastRequest"></wsdl:input>
  <wsdl:output message="rti:scamSetContrastResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetContrast">
  <wsdl:documentation>
    Returns the image contrast. @param session Session token if the used login
```

```
    method is based on session token type. @param deviceId Device identifier.
    @return Current value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetContrastRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetContrastResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetPanRotation">
  <wsdl:documentation>
    Returns the current pan rotation in degrees. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return The current pan rotation in degrees. @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetPanRotationRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetPanRotationResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamSetPanRotation">
  <wsdl:documentation>
    The pan will be moved until the desired position in degrees. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @param value New value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamSetPanRotationRequest"></wsdl:input>
  <wsdl:output message="rti:scamSetPanRotationResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetTiltRotation">
  <wsdl:documentation>
    Return the current tilt rotation in degrees. @param session Session token if the
    used login method is based on session token type. @param deviceId Device
    identifier. @return The current tilt rotation in degrees. @throws RtiError In
    error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetTiltRotationRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetTiltRotationResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamSetTiltRotation">
  <wsdl:documentation>
    The tilt will be moved until the desired position in degrees. @param session
    Session token if the used login method is based on session token type. @param
    deviceId Device identifier. @param value New value. @throws RtiError In error
    case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamSetTiltRotationRequest"></wsdl:input>
  <wsdl:output message="rti:scamSetTiltRotationResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamGetZoom">
  <wsdl:documentation>
    Return the current zoom value. @param session Session token if the used login
    method is based on session token type. @param deviceId Device identifier.
    @return The current zoom value. @throws RtiError In error case.
  </wsdl:documentation>
  <wsdl:input message="rti:scamGetZoomRequest"></wsdl:input>
  <wsdl:output message="rti:scamGetZoomResponse"></wsdl:output>
  <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="scamSetZoom">
  <wsdl:documentation>
    The zoom will be moved until the desired position in degrees. @param session
    Session token if the used login method is based on session token type. @param
```

```
      deviceId Device identifier. @param value New value. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamSetZoomRequest"></wsdl:input>
    <wsdl:output message="rti:scamSetZoomResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamAbortExposure">
    <wsdl:documentation>
      Aborts the current exposure, if any, and returns the camera to READY state.
      @param session Session token if the used login method is based on session token
      type. @param deviceId Device identifier. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamAbortExposureRequest"></wsdl:input>
    <wsdl:output message="rti:scamAbortExposureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamStartExposure">
    <wsdl:documentation>
      Starts an exposure. Use ImageReady to check when the exposure is complete, and
      also use the AcquisitionMode and DigitizingMode values. This method uses
      ExposureTime variable. @param session Session token if the used login method is
      based on session token type. @param deviceId Device identifier. @param light
      Frame type (true: light frame, false: darkframe). @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamStartExposureRequest"></wsdl:input>
    <wsdl:output message="rti:scamStartExposureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamStopExposure">
    <wsdl:documentation>
      Stops the current exposure, if any. If an exposure is in progress, the readout
      process is initiated. Ignored if readout is already in process. @param session
      Session token if the used login method is based on session token type. @param
      deviceId Device identifier. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamStopExposureRequest"></wsdl:input>
    <wsdl:output message="rti:scamStopExposureResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="scamGetImage">
    <wsdl:documentation>
      Returns an image from the last exposure. The image could be an array of short,
      long, double (imageContentType attribute shows the type). @param session Session
      token if the used login method is based on session token type. @return Image
      data. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:scamGetImageRequest"></wsdl:input>
    <wsdl:output message="rti:scamGetImageResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="execStartOp">
    <wsdl:documentation>
      Starts the interactive Observing Plan execution. @param session Session token if
      the used login method is based on session token type. @param uuidOp Observing
      Plan identifier. @param user Observing Plan owner. @param seconds Observing Plan
      duration in seconds. @return Observing Plan UUID. @throws RtiError In error
      case.
    </wsdl:documentation>
    <wsdl:input message="rti:execStartOpRequest"></wsdl:input>
    <wsdl:output message="rti:execStartOpResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="execStopOp">
```

```
    <wsdl:documentation>
      Finishes the interactive Observing Plan execution. @param session Session token
      if the used login method is based on session token type. @throws RtiError In
      error case.
    </wsdl:documentation>
    <wsdl:input message="rti:execStopOpRequest"></wsdl:input>
    <wsdl:output message="rti:execStopOpResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="execGetInfo">
    <wsdl:documentation>
      Recovers the executor information. @param session Session token if the used
      login method is based on session token type. @return Executor information.
      @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:execGetInfoRequest"></wsdl:input>
    <wsdl:output message="rti:execGetInfoResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="execGetLastOpInfo">
    <wsdl:documentation>
      Recovers the last observing plan information. @param session Session token if
      the used login method is based on session token type. @return Executor
      information. @throws RtiError In error case.
    </wsdl:documentation>
    <wsdl:input message="rti:execGetLastOpInfoRequest"></wsdl:input>
    <wsdl:output message="rti:execGetLastOpInfoResponse"></wsdl:output>
    <wsdl:fault message="rti:rtiError" name="error"></wsdl:fault>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="gloria_rtiSOAP" type="rti:gloria_rti">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="loginCert">
    <soap:operation soapAction="http://gloria.eu/rti/loginCert"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="login">
    <soap:operation soapAction="http://gloria.eu/rti/login"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="logout">
    <soap:operation soapAction="http://gloria.eu/rti/logout"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devGetDevice">
    <soap:operation soapAction="http://gloria.eu/rti/devGetDevice"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devGetDevices">
    <soap:operation soapAction="http://gloria.eu/rti/devGetDevices"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devGetDeviceProperties">
    <soap:operation soapAction="http://gloria.eu/rti/devGetDeviceProperties"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devGetDeviceProperty">
    <soap:operation soapAction="http://gloria.eu/rti/devGetDeviceProperty"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devUpdateDeviceProperty">
    <soap:operation soapAction="http://gloria.eu/rti/devUpdateDeviceProperty"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devUpdateDevicePropertyAsync">
    <soap:operation soapAction="http://gloria.eu/rti/devUpdateDeviceProperty"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devIsConnected">
    <soap:operation soapAction="http://gloria.eu/rti/devIsConnected"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="devConnect">
    <soap:operation soapAction="http://gloria.eu/rti/devConnect"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="devDisconnect">
      <soap:operation soapAction="http://gloria.eu/rti/devDisconnect"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="devIsBlocked">
      <soap:operation soapAction="http://gloria.eu/rti/devIsBlocked"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="devGetConfiguration">
      <soap:operation soapAction="http://gloria.eu/rti/devGetConfiguration"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="camImageReady">
      <soap:operation soapAction="http://gloria.eu/rti/camImageReady"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="camGetCameraType">
      <soap:operation soapAction="http://gloria.eu/rti/camGetCameraType"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="camGetXsize">
      <soap:operation soapAction="http://gloria.eu/rti/camGetXsize"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="camGetYSize">
      <soap:operation soapAction="http://gloria.eu/rti/camGetYSize"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="camCanAbortExposure">
```

```
    <soap:operation soapAction="http://gloria.eu/rti/camCanAbortExposure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camCanAsymetricBin">
    <soap:operation soapAction="http://gloria.eu/rti/camCanAsymetricBin"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camCanGetCoolerPower">
    <soap:operation soapAction="http://gloria.eu/rti/camCanGetCoolerPower"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camCanSetCCDTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/camCanSetCCDTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camCanControlTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/camCanControlTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camCanStopExposure">
    <soap:operation soapAction="http://gloria.eu/rti/camCanStopExposure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetCoolerPower">
    <soap:operation soapAction="http://gloria.eu/rti/camGetCoolerPower"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetElectronsPerADU">
    <soap:operation soapAction="http://gloria.eu/rti/camGetElectronsPerADU"/>
    <wsdl:input>
```

```
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetFullWellCapacity">
    <soap:operation soapAction="http://gloria.eu/rti/camGetFullWellCapacity"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasShutter">
    <soap:operation soapAction="http://gloria.eu/rti/camHasShutter"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasBrightness">
    <soap:operation soapAction="http://gloria.eu/rti/camHasBrightness"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasConstrast">
    <soap:operation soapAction="http://gloria.eu/rti/camHasConstrast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasGain">
    <soap:operation soapAction="http://gloria.eu/rti/camHasGain"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasGamma">
    <soap:operation soapAction="http://gloria.eu/rti/camHasGamma"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasSubframe">
    <soap:operation soapAction="http://gloria.eu/rti/camHasSubframe"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
```

```
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHasExposureTime">
    <soap:operation soapAction="http://gloria.eu/rti/camHasExposureTime"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camHeatSinkTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/camHeatSinkTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camIsPulseGuiding">
    <soap:operation soapAction="http://gloria.eu/rti/camIsPulseGuiding"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetLastError">
    <soap:operation soapAction="http://gloria.eu/rti/camGetLastError"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetLastExposureDuration">
    <soap:operation soapAction="http://gloria.eu/rti/camGetLastExposureDuration"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetLastExposureStart">
    <soap:operation soapAction="http://gloria.eu/rti/camGetLastExposureStart"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetMaxAdu">
    <soap:operation soapAction="http://gloria.eu/rti/camGetMaxAdu"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
```

```
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetMaxBinX">
    <soap:operation soapAction="http://gloria.eu/rti/camGetMaxBinX"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetMaxBinY">
    <soap:operation soapAction="http://gloria.eu/rti/camGetMaxBinY"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetPixelSizeX">
    <soap:operation soapAction="http://gloria.eu/rti/camGetPixelSizeX"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetPixelSizeY">
    <soap:operation soapAction="http://gloria.eu/rti/camGetPixelSizeY"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetAcquisitionMode">
    <soap:operation soapAction="http://gloria.eu/rti/camGetAcquisitionMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetFPS">
    <soap:operation soapAction="http://gloria.eu/rti/camGetFPS"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetDigitilizingMode">
    <soap:operation soapAction="http://gloria.eu/rti/camGetDigitilizingMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```
<wsdl:operation name="camGetBinX">
  <soap:operation soapAction="http://gloria.eu/rti/camGetBinX"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camSetBinX">
  <soap:operation soapAction="http://gloria.eu/rti/camSetBinX"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetBinY">
  <soap:operation soapAction="http://gloria.eu/rti/camGetBinY"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camSetBinY">
  <soap:operation soapAction="http://gloria.eu/rti/camSetBinY"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camIsCoolerOn">
  <soap:operation soapAction="http://gloria.eu/rti/camIsCoolerOn"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camSetCoolerOn">
  <soap:operation soapAction="http://gloria.eu/rti/camSetCoolerOn"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetROINumX">
  <soap:operation soapAction="http://gloria.eu/rti/camGetROINumX"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camSetROINumX">
  <soap:operation soapAction="http://gloria.eu/rti/camSetROINumX"/>
```

```
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetROINumY">
    <soap:operation soapAction="http://gloria.eu/rti/camSetROINumY"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetROINumY">
    <soap:operation soapAction="http://gloria.eu/rti/camGetROINumY"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetROIStartX">
    <soap:operation soapAction="http://gloria.eu/rti/camSetROIStartX"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetROIStartX">
    <soap:operation soapAction="http://gloria.eu/rti/camGetROIStartX"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetROIStartY">
    <soap:operation soapAction="http://gloria.eu/rti/camSetROIStartY"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetROIStartY">
    <soap:operation soapAction="http://gloria.eu/rti/camGetROIStartY"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetBrightness">
    <soap:operation soapAction="http://gloria.eu/rti/camSetBrightness"/>
    <wsdl:input>
      <soap:body use="literal"/>
```

```
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetBrightness">
    <soap:operation soapAction="http://gloria.eu/rti/camGetBrightness"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetContrast">
    <soap:operation soapAction="http://gloria.eu/rti/camSetContrast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetContrast">
    <soap:operation soapAction="http://gloria.eu/rti/camGetContrast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetGain">
    <soap:operation soapAction="http://gloria.eu/rti/camSetGain"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetGain">
    <soap:operation soapAction="http://gloria.eu/rti/camGetGain"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetGamma">
    <soap:operation soapAction="http://gloria.eu/rti/camSetGamma"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetGamma">
    <soap:operation soapAction="http://gloria.eu/rti/camGetGamma"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetExposureTime">
    <soap:operation soapAction="http://gloria.eu/rti/camSetExposureTime"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetExposureTime">
    <soap:operation soapAction="http://gloria.eu/rti/camGetExposureTime"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetObjectExposureTime">
    <soap:operation soapAction="http://gloria.eu/rti/camGetObjectExposureTime"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetCCDTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/camSetCCDTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetCCDTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/camGetCCDTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetCCDCurrentTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/camGetCCDCurrentTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camAbortExposure">
    <soap:operation soapAction="http://gloria.eu/rti/camAbortExposure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
```

```
</wsdl:operation>
<wsdl:operation name="camPulseGuide">
  <soap:operation soapAction="http://gloria.eu/rti/camPulseGuide"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camStartExposure">
  <soap:operation soapAction="http://gloria.eu/rti/camStartExposure"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camStopExposure">
  <soap:operation soapAction="http://gloria.eu/rti/camStopExposure"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetImage">
  <soap:operation soapAction="http://gloria.eu/rti/camGetImage"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetOneShotModeImageFormats">
  <soap:operation soapAction="http://gloria.eu/rti/camGetOneShotModeImageFormats"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetContinueModeImageFormats">
  <soap:operation
  soapAction="http://gloria.eu/rti/camGetContinueModeImageFormats"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetImageDataType">
  <soap:operation soapAction="http://gloria.eu/rti/camGetImageDataType"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
```

```
<wsdl:operation name="camSetBitDepth">
  <soap:operation soapAction="http://gloria.eu/rti/camSetBitDepth"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetBitDepth">
  <soap:operation soapAction="http://gloria.eu/rti/camGetBitDepth"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetContinueModeQuality">
  <soap:operation soapAction="http://gloria.eu/rti/camGetContinueModeQuality"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camSetContinueModeQuality">
  <soap:operation soapAction="http://gloria.eu/rti/camSetContinueModeQuality"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetOneShotModeQuality">
  <soap:operation soapAction="http://gloria.eu/rti/camGetOneShotModeQuality"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camSetOneShotModeQuality">
  <soap:operation soapAction="http://gloria.eu/rti/camSetOneShotModeQuality"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetContinueModeImagePath">
  <soap:operation soapAction="http://gloria.eu/rti/camGetContinueModeImagePath"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="camGetOneShotModeImagePath">
  <soap:operation soapAction="http://gloria.eu/rti/camGetOneShotModeImagePath"/>
```

```
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetAutoGain">
    <soap:operation soapAction="http://gloria.eu/rti/camGetAutoGain"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetAutoExposureTime">
    <soap:operation soapAction="http://gloria.eu/rti/camGetAutoExposureTime"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetAutoGain">
    <soap:operation soapAction="http://gloria.eu/rti/camSetAutoGain"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camSetAutoExposureTime">
    <soap:operation soapAction="http://gloria.eu/rti/camSetAutoExposureTime"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camStartContinueMode">
    <soap:operation soapAction="http://gloria.eu/rti/camStartContinueMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camStopContinueMode">
    <soap:operation soapAction="http://gloria.eu/rti/camStopContinueMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetImageURL">
    <soap:operation soapAction="http://gloria.eu/rti/camGetImageURL"/>
    <wsdl:input>
      <soap:body use="literal"/>
```

```
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetImageURLProperFormat">
    <soap:operation soapAction="http://gloria.eu/rti/camGetImageURLProperFormat"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetFilterWheels">
    <soap:operation soapAction="http://gloria.eu/rti/camGetFilterWheels"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="camGetFocuser">
    <soap:operation soapAction="http://gloria.eu/rti/camGetFocuser"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domGetNumberElement">
    <soap:operation soapAction="http://gloria.eu/rti/domGetNumberElement"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domCanSetAltitude">
    <soap:operation soapAction="http://gloria.eu/rti/domCanSetAltitude"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domCanSetAzimuth">
    <soap:operation soapAction="http://gloria.eu/rti/domCanSetAzimuth"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domCanSetPark">
    <soap:operation soapAction="http://gloria.eu/rti/domCanSetPark"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domIsAtHome">
    <soap:operation soapAction="http://gloria.eu/rti/domIsAtHome"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domIsAtPark">
    <soap:operation soapAction="http://gloria.eu/rti/domIsAtPark"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domGetAltitude">
    <soap:operation soapAction="http://gloria.eu/rti/domGetAltitude"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domGetAzimuth">
    <soap:operation soapAction="http://gloria.eu/rti/domGetAzimuth"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domOpen">
    <soap:operation soapAction="http://gloria.eu/rti/domOpen"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domClose">
    <soap:operation soapAction="http://gloria.eu/rti/domClose"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="domGoHome">
    <soap:operation soapAction="http://gloria.eu/rti/domGoHome"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
```

```
</wsdl:operation>
<wsdl:operation name="domSetPark">
  <soap:operation soapAction="http://gloria.eu/rti/domSetPark"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="domPark">
  <soap:operation soapAction="http://gloria.eu/rti/domPark"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="domMoveAzimuth">
  <soap:operation soapAction="http://gloria.eu/rti/domMoveAzimuth"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="domMoveAltitude">
  <soap:operation soapAction="http://gloria.eu/rti/domMoveAltitude"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="domSetTracking">
  <soap:operation soapAction="http://gloria.eu/rti/domSetTracking"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="domGetTracking">
  <soap:operation soapAction="http://gloria.eu/rti/domGetTracking"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="domSlewObject">
  <soap:operation soapAction="http://gloria.eu/rti/domSlewObject"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntGetUtcClock">
```

```
    <soap:operation soapAction="http://gloria.eu/rti/mntGetUtcClock"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSetUtcClock">
    <soap:operation soapAction="http://gloria.eu/rti/mntSetUtcClock"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetSiderealDate">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetSiderealDate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntIsAtHome">
    <soap:operation soapAction="http://gloria.eu/rti/mntIsAtHome"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntIsParked">
    <soap:operation soapAction="http://gloria.eu/rti/mntIsParked"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetTargetRightAscension">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetTargetRightAscension"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetTargetDeclination">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetTargetDeclination"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetTrackingDeclinationRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetTrackingDeclinationRate"/>
    <wsdl:input>
```

```
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetTrackingAscensionRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetTrackingAscensionRate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetTrackingRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetTrackingRate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSetTrackingRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntSetTrackingRate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetTracking">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetTracking"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSetTracking">
    <soap:operation soapAction="http://gloria.eu/rti/mntSetTracking"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetGuideRateDeclination">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetGuideRateDeclination"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetDeclinationRateRightAscension">
    <soap:operation
    soapAction="http://gloria.eu/rti/mntGetDeclinationRateRightAscension"/>
    <wsdl:input>
      <soap:body use="literal"/>
```

```
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntIsSlewing">
    <soap:operation soapAction="http://gloria.eu/rti/mntIsSlewing"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetPosAxis2">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetPosAxis2"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetPosAxis3">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetPosAxis3"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetPosAxis1">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetPosAxis1"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanPulseGuide">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanPulseGuide"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanSetGuideRates">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanSetGuideRates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanSetPark">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanSetPark"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSetPark">
    <soap:operation soapAction="http://gloria.eu/rti/mntSetPark"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetALTParkPos">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetALTParkPos"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetAZParkPos">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetAZParkPos"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanSetTracking">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanSetTracking"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanSetTrackingRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanSetTrackingRate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewCoordinates">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanSlewCoordinates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntCanSlewCoordinatesAsync">
    <soap:operation soapAction="http://gloria.eu/rti/mntCanSlewCoordinatesAsync"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
```

```
</wsdl:operation>
<wsdl:operation name="mntCanSlewObject">
  <soap:operation soapAction="http://gloria.eu/rti/mntCanSlewObject"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntCanSlewAltAz">
  <soap:operation soapAction="http://gloria.eu/rti/mntCanSlewAltAz"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntCanSlewAzAsync">
  <soap:operation soapAction="http://gloria.eu/rti/mntCanSlewAzAsync"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntCanMoveAzis">
  <soap:operation soapAction="http://gloria.eu/rti/mntCanMoveAzis"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntAxisRate">
  <soap:operation soapAction="http://gloria.eu/rti/mntAxisRate"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntTrackingRates">
  <soap:operation soapAction="http://gloria.eu/rti/mntTrackingRates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntGoHome">
  <soap:operation soapAction="http://gloria.eu/rti/mntGoHome"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="mntPark">
```

```
    <soap:operation soapAction="http://gloria.eu/rti/mntPark"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntUnpark">
    <soap:operation soapAction="http://gloria.eu/rti/mntUnpark"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToAltAz">
    <soap:operation soapAction="http://gloria.eu/rti/mntSlewToAltAz"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToAltAzAsync">
    <soap:operation soapAction="http://gloria.eu/rti/mntSlewToAltAzAsync"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToCoordinates">
    <soap:operation soapAction="http://gloria.eu/rti/mntSlewToCoordinates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSlewToCoordinatesAsync">
    <soap:operation soapAction="http://gloria.eu/rti/mntSlewToCoordinatesAsync"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntMoveAxis">
    <soap:operation soapAction="http://gloria.eu/rti/mntMoveAxis"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntPulseGuide">
    <soap:operation soapAction="http://gloria.eu/rti/mntPulseGuide"/>
    <wsdl:input>
```

```
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntStopSlewAxis">
    <soap:operation soapAction="http://gloria.eu/rti/mntStopSlewAxis"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntStopSlew">
    <soap:operation soapAction="http://gloria.eu/rti/mntStopSlew"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetPointingModel">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetPointingModel"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntMoveNorth">
    <soap:operation soapAction="http://gloria.eu/rti/mntMoveNorth"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntMoveSouth">
    <soap:operation soapAction="http://gloria.eu/rti/mntMoveSouth"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntMoveEast">
    <soap:operation soapAction="http://gloria.eu/rti/mntMoveEast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntMoveWest">
    <soap:operation soapAction="http://gloria.eu/rti/mntMoveWest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
```

```
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSetSlewRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntSetSlewRate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntGetSlewRate">
    <soap:operation soapAction="http://gloria.eu/rti/mntGetSlewRate"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntSlewObject">
    <soap:operation soapAction="http://gloria.eu/rti/mntSlewObject"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntIsPointingAtObject">
    <soap:operation soapAction="http://gloria.eu/rti/mntIsPointingAtObject"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="mntIsPointingAtCoordinates">
    <soap:operation soapAction="http://gloria.eu/rti/mntIsPointingAtCoordinates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetCamera">
    <soap:operation soapAction="http://gloria.eu/rti/focGetCamera"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focIsAbsolute">
    <soap:operation soapAction="http://gloria.eu/rti/focIsAbsolute"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
```

```
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetStepSize">
    <soap:operation soapAction="http://gloria.eu/rti/focGetStepSize"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetMaxIncrement">
    <soap:operation soapAction="http://gloria.eu/rti/focGetMaxIncrement"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetMaxStep">
    <soap:operation soapAction="http://gloria.eu/rti/focGetMaxStep"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetMinStep">
    <soap:operation soapAction="http://gloria.eu/rti/focGetMinStep"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetPosition">
    <soap:operation soapAction="http://gloria.eu/rti/focGetPosition"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focIsTempCompAvailable">
    <soap:operation soapAction="http://gloria.eu/rti/focIsTempCompAvailable"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="focGetTemperature">
    <soap:operation soapAction="http://gloria.eu/rti/focGetTemperature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```
<wsdl:operation name="focSetTempComp">
  <soap:operation soapAction="http://gloria.eu/rti/focSetTempComp"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="focHalt">
  <soap:operation soapAction="http://gloria.eu/rti/focHalt"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="focMove">
  <soap:operation soapAction="http://gloria.eu/rti/focMove"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="dwrGetChannelsNumber">
  <soap:operation soapAction="http://gloria.eu/rti/dwrGetChannelsNumber"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="dwrGetChannelType">
  <soap:operation soapAction="http://gloria.eu/rti/dwrGetChannelType"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="dwrSetTemperatureThreshold">
  <soap:operation soapAction="http://gloria.eu/rti/dwrSetTemperatureThreshold"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="dwrSetCycleThreshold">
  <soap:operation soapAction="http://gloria.eu/rti/dwrSetCycleThreshold"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="fhtGetMeasureUnit">
  <soap:operation soapAction="http://gloria.eu/rti/fhtGetMeasureUnit"/>
```

```
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fhtGetMeasure">
    <soap:operation soapAction="http://gloria.eu/rti/fhtGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fhtSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/fhtSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fhtGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/fhtGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rndGetMeasureUnit">
    <soap:operation soapAction="http://gloria.eu/rti/rndGetMeasureUnit"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rndGetMeasure">
    <soap:operation soapAction="http://gloria.eu/rti/rndGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rndSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/rndSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rndGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/rndGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
```

```
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rndIsRaining">
    <soap:operation soapAction="http://gloria.eu/rti/rndIsRaining"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rhsGetMeasureUnit">
    <soap:operation soapAction="http://gloria.eu/rti/rhsGetMeasureUnit"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rhsGetMeasure">
    <soap:operation soapAction="http://gloria.eu/rti/rhsGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rhsSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/rhsSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rhsGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/rhsGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="barGetMeasureUnit">
    <soap:operation soapAction="http://gloria.eu/rti/barGetMeasureUnit"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="barGetMeasure">
    <soap:operation soapAction="http://gloria.eu/rti/barGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="barSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/barSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="barGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/barGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="tempGetMeasureUnit">
    <soap:operation soapAction="http://gloria.eu/rti/tempGetMeasureUnit"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="tempGetMeasure">
    <soap:operation soapAction="http://gloria.eu/rti/tempGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="tempSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/tempSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="tempGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/tempGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="strIsAvailableOrientation">
    <soap:operation soapAction="http://gloria.eu/rti/strIsAvailableOrientation"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
```

```
</wsdl:operation>
<wsdl:operation name="strGetDistanceMeasureUnit">
  <soap:operation soapAction="http://gloria.eu/rti/strGetDistanceMeasureUnit"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="strGetDistanceMeasure">
  <soap:operation soapAction="http://gloria.eu/rti/strGetDistanceMeasure"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="strGetDegrees">
  <soap:operation soapAction="http://gloria.eu/rti/strGetDegrees"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="strGetAbosoluteDegrees">
  <soap:operation soapAction="http://gloria.eu/rti/strGetAbosoluteDegrees"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="strSetDistanceMeasureStates">
  <soap:operation soapAction="http://gloria.eu/rti/strSetDistanceMeasureStates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="strGetDistanceMeasureStates">
  <soap:operation soapAction="http://gloria.eu/rti/strGetDistanceMeasureStates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="strSetOrientationMeasureStates">
  <soap:operation
  soapAction="http://gloria.eu/rti/strSetOrientationMeasureStates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
```

```
<wsdl:operation name="strGetOrientationMeasureStates">
  <soap:operation
  soapAction="http://gloria.eu/rti/strGetOrientationMeasureStates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wvnGetMeasureUnit">
  <soap:operation soapAction="http://gloria.eu/rti/wvnGetMeasureUnit"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wvnGetMeasure">
  <soap:operation soapAction="http://gloria.eu/rti/wvnGetMeasure"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wvnGetAbosluteDegrees">
  <soap:operation soapAction="http://gloria.eu/rti/wvnGetAbosluteDegrees"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wvnSetMeasureStates">
  <soap:operation soapAction="http://gloria.eu/rti/wvnSetMeasureStates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wvnGetMeasureStates">
  <soap:operation soapAction="http://gloria.eu/rti/wvnGetMeasureStates"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wspGetMeasureUnit">
  <soap:operation soapAction="http://gloria.eu/rti/wspGetMeasureUnit"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="wspGetMeasure">
```

```
    <soap:operation soapAction="http://gloria.eu/rti/wspGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="wspSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/wspSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="wspGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/wspGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="cldGetMeasureUnit">
    <soap:operation soapAction="http://gloria.eu/rti/cldGetMeasureUnit"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="cldGetMeasure">
    <soap:operation soapAction="http://gloria.eu/rti/cldGetMeasure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="cldSetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/cldSetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="cldGetMeasureStates">
    <soap:operation soapAction="http://gloria.eu/rti/cldGetMeasureStates"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGetCamera">
    <soap:operation soapAction="http://gloria.eu/rti/fwGetCamera"/>
    <wsdl:input>
```

```
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGetFilterList">
    <soap:operation soapAction="http://gloria.eu/rti/fwGetFilterList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGetPositionNumber">
    <soap:operation soapAction="http://gloria.eu/rti/fwGetPositionNumber"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGetSpeedSwitching">
    <soap:operation soapAction="http://gloria.eu/rti/fwGetSpeedSwitching"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGetFilterSize">
    <soap:operation soapAction="http://gloria.eu/rti/fwGetFilterSize"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGetFilterKind">
    <soap:operation soapAction="http://gloria.eu/rti/fwGetFilterKind"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwIsAtHome">
    <soap:operation soapAction="http://gloria.eu/rti/fwIsAtHome"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwSetOffset">
    <soap:operation soapAction="http://gloria.eu/rti/fwSetOffset"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
```

```
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwSelectFilterKind">
    <soap:operation soapAction="http://gloria.eu/rti/fwSelectFilterKind"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwSelectFilterPosition">
    <soap:operation soapAction="http://gloria.eu/rti/fwSelectFilterPosition"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="fwGoHome">
    <soap:operation soapAction="http://gloria.eu/rti/fwGoHome"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rttGetCurrentPosition">
    <soap:operation soapAction="http://gloria.eu/rti/rttGetCurrentPosition"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="rttSetTargetPosition">
    <soap:operation soapAction="http://gloria.eu/rti/rttSetTargetPosition"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamHasBrightness">
    <soap:operation soapAction="http://gloria.eu/rti/scamHasBrightness"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamHasContrast">
    <soap:operation soapAction="http://gloria.eu/rti/scamHasContrast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
```

```
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamAcquisitionMode">
    <soap:operation soapAction="http://gloria.eu/rti/scamAcquisitionMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetFPS">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetFPS"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetDigitizingMode">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetDigitizingMode"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamIsPTSupported">
    <soap:operation soapAction="http://gloria.eu/rti/scamIsPTSupported "/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamIsZoomSupported">
    <soap:operation soapAction="http://gloria.eu/rti/scamIsZoomSupported"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetPanMin">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetPanMin"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetPanMax">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetPanMax"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```
<wsdl:operation name="scamGetTiltMin">
  <soap:operation soapAction="http://gloria.eu/rti/scamGetTiltMin"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamGetTiltMax">
  <soap:operation soapAction="http://gloria.eu/rti/scamGetTiltMax"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamGetZoomMax">
  <soap:operation soapAction="http://gloria.eu/rti/scamGetZoomMax"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamGetExposureTime">
  <soap:operation soapAction="http://gloria.eu/rti/scamGetExposureTime"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamSetExposureTime">
  <soap:operation soapAction="http://gloria.eu/rti/scamSetExposureTime"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamIsImageReady">
  <soap:operation soapAction="http://gloria.eu/rti/scamIsImageReady"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamGetVideoStreamingURL">
  <soap:operation soapAction="http://gloria.eu/rti/scamGetVideoStreamingURL"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="scamGetImageURL">
  <soap:operation soapAction="http://gloria.eu/rti/scamGetImageURL"/>
```

```
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamSetBrightness">
    <soap:operation soapAction="http://gloria.eu/rti/scamSetBrightness"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetBrightness">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetBrightness"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamSetContrast">
    <soap:operation soapAction="http://gloria.eu/rti/scamSetContrast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetContrast">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetContrast"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetPanRotation">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetPanRotation"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamSetPanRotation">
    <soap:operation soapAction="http://gloria.eu/rti/scamSetPanRotation"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetTiltRotation">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetTiltRotation"/>
    <wsdl:input>
      <soap:body use="literal"/>
```

```
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamSetTiltRotation">
    <soap:operation soapAction="http://gloria.eu/rti/scamSetTiltRotation"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetZoom">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetZoom"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamSetZoom">
    <soap:operation soapAction="http://gloria.eu/rti/scamSetZoom"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamAbortExposure">
    <soap:operation soapAction="http://gloria.eu/rti/scamAbortExposure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamStartExposure">
    <soap:operation soapAction="http://gloria.eu/rti/scamStartExposure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamStopExposure">
    <soap:operation soapAction="http://gloria.eu/rti/scamStopExposure"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="scamGetImage">
    <soap:operation soapAction="http://gloria.eu/rti/scamGetImage"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="execStartOp">
    <soap:operation soapAction="http://gloria.eu/rti/execStartOp"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="execStopOp">
    <soap:operation soapAction="http://gloria.eu/rti/execStopOp"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="execGetInfo">
    <soap:operation soapAction="http://gloria.eu/rti/execGetInfo"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="execGetLastOpInfo">
    <soap:operation soapAction="http://gloria.eu/rti/execGetLastOpInfo"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
 </wsdl:binding>
 <wsdl:service name="gloria_rti">
  <wsdl:documentation/>
  <wsdl:port binding="rti:gloria_rtiSOAP" name="gloria_rtiSOAP">
    <soap:address
    location="https://altamira.asu.cas.cz:8444/RTI/services/gloria_rtiSOAP"/>
  </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```