

REDHAWK JSON-XML Adapter

+
Stephanie

The background features a light blue-grey gradient with several thin, wavy, dashed lines in a light brown color. There are two white circles: one in the top-left corner and one in the bottom-right corner. A small purple plus sign is located above the word 'Background'.

Background

JSON-XML Conversion



REDHAWK VSCode extension was using TypeScript to convert component and waveform files



Desire for a more standalone tool that also utilizes current REDHAWK parsers (Python)



Plans to make the adapter extensible for frameworks other than REDHAWK

Initial Goals

Create tool with command-line arguments for converting JSON and XML files



Integrate tool into REDHAWK VSCode extension

The background features a light blue-grey gradient with several thin, wavy, dashed lines in a light brown color. There are two white circles: one in the top-left corner and one in the bottom-right corner. A small purple plus sign is located above the text.

Project In Use

Component in JSON⁺

```
{
  "name": "my_component",
  "metadata": {
    "id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0",
    "description": "Test description!",
    "implementations": [
      "cpp"
    ]
  },
  "properties": [
    {
      "name": "my_simple",
      "value": {
        "type": "string",
        "default": "Hello"
      },
      "metadata": {
        "mode": "readonly",
        "kind": [
          "test"
        ],
        "action": "eq"
      }
    },
    {
      "name": "my_simple_sequence",
      "value": {
        "type": "float",
        "default": [
          1,
          2,
          3
        ],
        "is_sequence": true
      }
    }
  ]
}
```

```
{
  "metadata": {
    "mode": "writeonly",
    "kind": [
      "event"
    ],
    "action": "gt"
  },
  {
    "name": "my_struct",
    "value": {
      "type": [
        {
          "name": "my_struct::simple",
          "value": {
            "type": "string",
            "default": "Howdy"
          },
          "metadata": {
            "kind": [
              "message"
            ],
            "action": "lt"
          }
        }
      ]
    },
    "metadata": {
      "mode": "readwrite",
      "kind": "configure"
    }
  }
},
]
```

```
Add In Port | Add Out Port | Add Bidirectional Port
"ports": [
  {
    "name": "dataFloat",
    "interface": "IDL:BULKIO/dataFloat:1.0",
    "direction": "in"
  },
  {
    "name": "dataShort",
    "interface": "IDL:BULKIO/dataShort:1.0",
    "direction": "out"
  }
]
}
```

Component JSON editing in VSCode extension

my_component Component

Id: DCE:3b523608-43f7-4334-89...
Description: Test description!
Implementat...: `cpp` / `python`

Properties

Category	Simple	Category	Simple Sequence	Category	Struct
Name	my_simple	Name	my_simple_sequence	Name	my_struct
Value	Hello	Value	1,2,3	Value	Type
Type	string	Type	float	Type	string
Mode	readonly	Mode	writeonly	Kind	message
Kind	test	Kind	event	Action	It
Action	eq	Action	gt	Complex	Enumerations
Complex	Enumerations	Complex	Enumerations	Mode	readwrite
				Kind	allocation
				Action	external
				Complex	

Updated SPD and PRF files

```
my_component > my_component.spd.xml
```

```
1 <softpkg id="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0" name="my_component">
2   <description>Test description!</description>
3   <propertyfile type="PRF">
4     <localfile name="my_component.prf.xml"/>
5   </propertyfile>
6   <descriptor>
7     <localfile name="my_component.scd.xml"/>
8   </descriptor>
9   <implementation id="cpp">
10    <code type="Executable">
11      <localfile name="cpp/my_component"/>
12      <entrypoint>cpp/my_component</entrypoint>
13    </code>
14    <programminglanguage name="C++"/>
15  </implementation>
16 </softpkg>
```

```
my_component > my_component.prf.xml
```

```
1 <properties>
2   <simple id="my_simple" type="string" mode="readonly">
3     <value>Hello</value>
4     <kind kindtype="test"/>
5     <action type="eq"/>
6   </simple>
7   <simplesequence id="my_simple_sequence" type="float" mode="writeonly">
8     <values>
9       <value>1</value>
10      <value>2</value>
11      <value>3</value>
12    </values>
13    <kind kindtype="event"/>
14    <action type="gt"/>
15  </simplesequence>
16  <struct id="my_struct">
17    <simple id="my_struct::simple" type="string">
18      <value>Howdy</value>
19      <kind kindtype="message"/>
20      <action type="lt"/>
21    </simple>
22    <configurationkind kindtype="allocation"/>
23  </struct>
24 </properties>
```


Component SCD editing in VSCode extension

Ports

Name	<input type="text" value="dataFloat"/>		Name	<input type="text" value="dataShort"/>		<input type="button" value="New"/>
Interface	<input type="text" value="IDL:BULKIO/dataFloat:1.0"/>		Interface	<input type="text" value="IDL:BULKIO/dataShort:1.0"/>		
Direction	<input type="text" value="in"/>		Direction	<input type="text" value="out"/>		

my_component > my_component.scd.xml

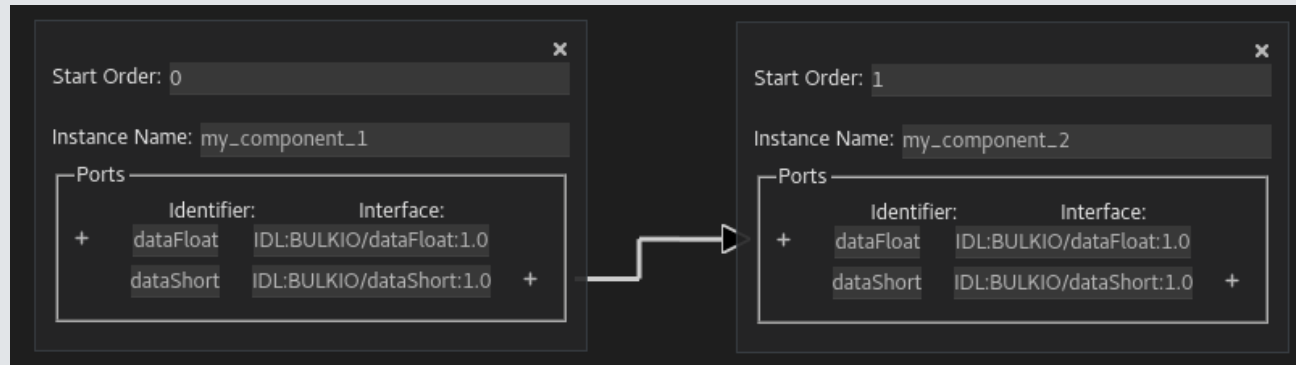
```
1 <softwarecomponent>
2   <componenttype>resource</componenttype>
3   <componentfeatures>
4     <supportsinterface repid="IDL:CF/PortAccessor:1.0" supportsname="PortAccessor"/>
5     <supportsinterface repid="IDL:CF/PropertySet:1.0" supportsname="PropertySet"/>
6     <supportsinterface repid="IDL:CF/ControllableInterface:1.0" supportsname="ControllableInterface"/>
7     <supportsinterface repid="IDL:RH/BaseComponent:1.0" supportsname="BaseComponent"/>
8     <supportsinterface repid="IDL:BULKIO/updateSRI:1.0" supportsname="updateSRI"/>
9     <supportsinterface repid="IDL:RH/Logging:1.0" supportsname="Logging"/>
10    <supportsinterface repid="IDL:BULKIO/dataShort:1.0" supportsname="dataShort"/>
11    <supportsinterface repid="IDL:RH/PropertyEmitter:1.0" supportsname="PropertyEmitter"/>
12    <supportsinterface repid="IDL:RH/PortAccessor:1.0" supportsname="PortAccessor"/>
13    <supportsinterface repid="IDL:BULKIO/ProvidesPortStatisticsProvider:1.0" supportsname="ProvidesPortStatisticsProvider"/>
14    <supportsinterface repid="IDL:BULKIO/dataFloat:1.0" supportsname="dataFloat"/>
15    <supportsinterface repid="IDL:CF/ComponentIdentifier:1.0" supportsname="ComponentIdentifier"/>
16    <supportsinterface repid="IDL:CF/LifeCycle:1.0" supportsname="LifeCycle"/>
17    <supportsinterface repid="IDL:CF/TestableInterface:1.0" supportsname="TestableInterface"/>
18    <ports>
19      <provides repid="IDL:BULKIO/dataFloat:1.0" providesname="dataFloat"/>
20      <uses repid="IDL:BULKIO/dataShort:1.0" usesname="dataShort"/>
21    </ports>
22  </componentfeatures>
```

Waveform in JSON

```
"name": "my_waveform",
"metadata": {
  "assembly_controller": "my_component_1",
  "version": "1.0.0",
  "description": ""
},
"component_files": [
  {
    "id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0",
    "file": "/components/my_component/my_component.spd.xml"
  }
],
"connections": [
  {
    "id": "connection_1",
    "from": {
      "instance_ref_id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_1",
      "port_ref_id": "dataShort"
    },
    "to": {
      "instance_ref_id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_2",
      "port_ref_id": "dataFloat"
    }
  }
],
```

```
"component_instances": [
  {
    "file_ref_id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0",
    "instance_id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_1",
    "instance_name": "my_component_1",
    "start_order": 0
  },
  {
    "file_ref_id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0",
    "instance_id": "DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_2",
    "instance_name": "my_component_2",
    "start_order": 1
  }
]
```

Waveform editing in VSCode extension



```
my_waveform > my_waveform.sad.xml
1 <softwareassembly name="my_waveform">
2   <componentfiles>
3     <componentfile id="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0" type="SPD">
4       <localfile name="/components/my_component/my_component.spd.xml"/>
5     </componentfile>
6   </componentfiles>
7   <partitioning>
8     <componentplacement>
9       <componentfilerref refid="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0"/>
10      <componentinstantiation id="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_1"/>
11    </componentplacement>
12    <componentplacement>
13      <componentfilerref refid="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0"/>
14      <componentinstantiation id="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_2"/>
15    </componentplacement>
16  </partitioning>
17  <assemblycontroller>
18    <componentinstantiationref refid="my_component_1"/>
19  </assemblycontroller>
20  <connections>
21    <connectinterface id="connection_1">
22      <usesport>
23        <identifier>dataShort</identifier>
24        <componentinstantiationref refid="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_1"/>
25      </usesport>
26      <providesport>
27        <identifier>dataFloat</identifier>
28        <componentinstantiationref refid="DCE:3b523608-43f7-4334-8967-d8d244e5a0f0_2"/>
29      </providesport>
30    </connectinterface>
31  </connections>
32 </softwareassembly>
```



Challenges

Challenges

01

Understanding and using REDHAWK parsers

02

Working with a schema that is currently being updated

03

Integrating the tool into the REDHAWK VSCode extension



What I Learned

New Experiences

1

Working with JSON
and XML files

2

Following Gitlab
processes and
CI/CD

- Pylint and unit testing

3

Working on a
virtual machine

4

Integrating a new
tool into an existing
one

- Comprehending
TypeScript



Status

Progress

- + Standalone script created for converting from JSON to XML and back through the command line
- + Adapter integrated and usable in the VSCode extension

Future Work

- + Test VSCode extension integration
- + Make adapter extensible
- + Account for all elements in adapter