

# Serialization

---

- [Introduction](#)
- [Examples of serialization](#)
  - [Empty model](#)
  - [Model with single item](#)
  - [PropertyItem with data](#)
  - [Parent with single child](#)

## Introduction

The `SessionModel` can be serialised to XML file, and then restored from it, using following code:

```
SessionModel model;
XmlDocument document({&model});
document.save("filename.xml");

model.clear(); // clear the model or modify it in any way

document.load("filename.xml");

// at this point, the model will be exactly as at the moment of saving
}
```

Multiple models can be saved in single XML file, if needed:

```
SessionModel model;
PulseScheduleModel pulse_schedule_model;
ComponentModel component_model;

XmlDocument document({&model, &pulse_schedule_module, &component_model});
document.save("filename.xml");
}
```

## Examples of serialization

### Empty model

C++

```
TestModel model;
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Document>
  <Model type="TestModel"/>
</Document>
```

## Model with single item

C++

```
TestModel model;
model.InsertItem<PropertyItem>();
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Document>
  <Model type="TestModel">
    <Item type="PropertyItem">
      <ItemData>
        <Variant role="0" type="string">{5c383869-750e-4b7d-af18-fbca95494254}</Variant>
        <Variant role="2" type="string">PropertyItem</Variant>
      </ItemData>
      <TaggedItems defaultTag=""/>
    </Item>
  </Model>
</Document>
```

## PropertyItem with data

C++

```
PropertyItem item;
item.setData(42, DataRole::kData);
item.setData("width", DataRole::kDisplay);
```

XML

```
<Item type="Property">
  <ItemData>
    <Variant role="0" type="string">{8f923bfc-94b3-456e-b222-0c81f19b8f5f}</Variant>
    <Variant role="1" type="int">42</Variant>
  </ItemData>
</Item>
```

```
<Variant role="2" type="string">width</Variant>
</ItemData>
<TaggedItems defaultTag=""/>
</Item>
```

## Parent with single child

### C++

```
SessionItem parent;
parent.setDisplayName("parent_name");
parent.registerTag(TagInfo::CreateUniversalTag("defaultTag"), /*set_as_default*/
true);

auto child = parent.insertItem(std::make_unique<PropertyItem>(),
TagIndex::append());
child->setDisplayName("child_name");
```

### XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Item type="SessionItem">
  <ItemData>
    <Variant role="0" type="string">{ca0fc80b-1246-4a69-8896-c6df46e3aa99}</Variant>
  </ItemData>
  <Variant role="2" type="string">parent_name</Variant>
</ItemData>
<TaggedItems defaultTag="defaultTag">
  <ItemContainer>
    <TagInfo max="-1" min="0" name="defaultTag"/>
    <Item type="Property">
      <ItemData>
        <Variant role="0" type="string">{9459511d-1096-4fa1-a3e8-eb1d7386694d}</Variant>
      </ItemData>
      <Variant role="2" type="string">child_name</Variant>
    </ItemData>
    <TaggedItems defaultTag=""/>
  </Item>
</ItemContainer>
</TaggedItems>
</Item>
```