

Public Support - Support Request #14314

Question about the IProperties Interface

2021-05-17 07:00 - hidden

Status: Closed	
Priority: Normal	
Category:	
Customer: AUDI	Product Issue Numbers:
Department:	Affected Products: ADTF 3.12.2
Requester's Priority: Normal	Platform: Windows 10 64bit
Support Level: 2nd Level	Topic: ADTF::SDK
Resolution: Solved Issue	FAQ Links:

Description

Supportanfrage

Hallo,

da wir ADTF3 Properties mit einem Generator generieren benutzen wir das IProperties Interface. Etwas schade finde ich noch, dass Min, Max, DisplayName, Description, IsChangeable, etc. noch nicht wirklich funktionieren. Aber das ist ja scheinbar im ADTF-IProperties-Interface nicht verfügbar. Wird das in Zukunft möglich sein, so wie in ADTF2?

Mit freundlichen Grüßen

i. A. Jonas Tratz
ASAP Engineering GmbH

Lösung:

Min/Max, DisplayName und Value Lists werden bereits unterstützt. Über das `property_variable<>` template auch sehr komfortabel mit `property_variable<>::SetValidRange()`, `property_variable<>::SetValueList()` und `property_variable<>::SetDisplayName()`.

Ansonsten musst du das selber über die entsprechenden Sub-Properties abbilden:

```
template <typename T>
tVoid set_valid_range(IProperties& oSubProperties, const T& xMin, const T& xMax)
{
    set_property<T>(oSubProperties,
        detail::tReservedProperties::strRangeMin,
        xMin);
    set_property<T>(oSubProperties,
        detail::tReservedProperties::strRangeMax,
        xMax);
}
```

Noch ein Nachtrag bzgl. Property Variables:

- https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html

Ich möchte im Filter eben nicht das `RegisterPropertyVariable()` benutzen, sondern "direkt" auf den Properties arbeiten.

z.B. in meinem Filter Konstruktor:

...

```

cPropertiesTestFilter::cPropertiesTestFilter(): m_pProps(NULL)
{
    object_ptr<IProperties> pProperties;
    THROW_IF_FAILED(GetProperties(pProperties));
    m_pProps = new properties::cGeneratedProperties(*pProperties);
}
...

```

benutzte ich meine Klasse die Properties anhand eines Input Files generiert:

```

...
class cGeneratedProperties
{
public:
    /// The IProperties on which the properties will be set on and read from.
    adtf::base::IProperties& m_oProperties;

public:
    cGeneratedProperties(adtf::base::IProperties& oProperties) : m_oProperties(oProperties)
    {
        Setup(m_oProperties);
    }

    virtual ~cGeneratedProperties() {};

    /**
     * Stores the IProperties as member var for later usage and sets the properties on this.
     */
    tVoid Setup(adtf::base::IProperties& oProperties)
    {
        m_oProperties = oProperties;

        m_oProperties.SetProperty(adtf::base::property<tInt>("propertyName", 1));
    }
}

```

Hier hätte ich gerne eine Zeile wie in ADTF2 um mit NSSUBPROP_DISPLAYNAME den Display name zu setzen:

```

//Damals in ADTF2:
m_pConfig->SetProperty(adtf::base::property<tChar*> (adtf_util::cString::Format("%s%s",
"propertyName", NSSUBPROP_DISPLAYNAME).GetPtr(), "Input::GridMap::Layer"));
//ADTF3 Wunschverhalten:
m_oProperties.SetProperty(adtf::base::property<tChar*> (adtf_util::cString::Format("%s%s",
"propertyName", NSSUBPROP_DISPLAYNAME).GetPtr(), "Input::GridMap::Layer"));

```

Zum lesen benutzte ich folgendes, welches einwandfrei funktioniert.

```

...
tVoid GetProperties()
{
    m_oProperties.GetProperty("propertyName", m_oIntProperty_tmp);
    nProp1 = m_oIntProperty_tmp.GetValueT();
}
...

tResult cPropertiesTestFilter::Init(tInitStage eStage)
{
    if (eStage == StageNormal)
    {
        m_pProps->GetProperties();
        LOG_INFO("Prop 1 = %u\n", m_pProps->nProp1);
    }
}

```

Die Frage wäre also, kann ich, wenn ich direkt auf den Properties arbeite, dort Sachen wie Min, Max, DisplayName, Description, IsChangeable, etc. setzen?

Oder brauche ich dafür immer eine property_variable?

wie oben in meinem Code Snippet beschrieben musst du die passenden SubProperties setzen:

```
object_ptr<IProperties> pProperties;  
THROW_IF_FAILED(GetProperties(pProperties));  
set_property_by_path<tInt32>(*pProperties, "myprop1", 123);  
set_property_by_path<tInt32>(*pProperties, (std::string("myprop1/") + adtf::base::giant:::  
detail::tReservedProperties::strRangeMin).c_str(), 0);  
set_property_by_path<tInt32>(*pProperties, (std::string("myprop1/") + adtf::base::giant:::  
detail::tReservedProperties::strRangeMax).c_str(), 200);
```

Eine kleine Frage hintendran, ich habe versucht "custom subproperties" zu erstellen, um quasi "changeable" properties abzubilden:

```
adtf::base::set_property_by_path<tBool>(m_oProperties, (std::string("myprop1").append(  
"/#__is_changeable")).c_str(), tTrue);
```

diese werden in ADTF3.10 nicht angezeigt, gibt es dazu eine Möglichkeit, damit man dies sehen kann?

Alles was mit "#__" anfängt wird nicht in die plugin description geschrieben, nimm einfach einen anderen Namen für die Subproperty.

History

#1 - 2021-05-17 12:49 - hidden

- Project changed from Public Support to 4
- Description updated
- Status changed from New to In Progress
- Customer set to AUDI
- Topic set to ADTF::Common
- Affected Products ADTF 3.12.2 added
- Platform Windows 10 64bit added

#3 - 2021-05-18 09:05 - hidden

Hi Jonas,

also Min/Max, DisplayName und Value Lists werden bereits unterstützt. Über das property_variable<> template auch sehr komfortabel mit
property_variable<>::SetValidRange()
property_variable<>::SetValueList()
property_variable<>::SetDisplayName()

Ansonsten musst du das selber über die entsprechenden Sub-Properties abbilden:

```
template <typename T>  
tVoid set_valid_range(IProperties& oSubProperties, const T& xMin, const T& xMax)  
{  
    set_property<T>(oSubProperties,  
        detail::tReservedProperties::strRangeMin,  
        xMin);  
    set_property<T>(oSubProperties,  
        detail::tReservedProperties::strRangeMax,  
        xMax);  
}
```

Grüße,

Martin

#4 - 2021-05-18 11:25 - hidden

- Status changed from In Progress to Customer Feedback Required
- Topic changed from ADTF::Common to ADTF::SDK

Noch ein Nachtrag bzgl. Property Variables:

- https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html

#5 - 2021-05-18 12:09 - hidden

Hallo zusammen,

danke für die schnelle und ausführlichen Antworten.

Ich möchte im Filter eben nicht das RegisterPropertyVariable() benutzen, sondern "direkt" auf den Properties arbeiten.

z.B. in meinen Filter Konstruktork:

```
...
cPropertiesTestFilter::cPropertiesTestFilter(): m_pProps(NULL)
{
    object_ptr<IProperties> pProperties;
    THROW_IF_FAILED(GetProperties(pProperties));
    m_pProps = new properties::cGeneratedProperties(*pProperties);
}
...
```

benutze ich meine Klasse die Properties anhand eines Input Files generiert:

```
...
class cGeneratedProperties
{
public:
    /// The IProperties on which the properties will be set on and read from.
    adtf::base::IProperties& m_oProperties;

public:
    cGeneratedProperties(adtf::base::IProperties& oProperties) : m_oProperties(oProperties)
    {
        Setup(m_oProperties);
    }

    virtual ~cGeneratedProperties() {};

    /**
     * Stores the IProperties as member var for later usage and sets the properties on this.
     */
    tVoid Setup(adtf::base::IProperties& oProperties)
    {
        m_oProperties = oProperties;

        m_oProperties.SetProperty(adtf::base::property<tInt>("propertyName", 1));
    }
};
```

Hier hätte ich gerne eine Zeile wie in ADTF2 um mit NSSUBPROP_DISPLAYNAME den Display name zu setzen:

```
//Damals in ADTF2:
m_pConfig->SetProperty(adtf::base::property<tChar*> (adtf_util::cString::Format("%s%s", "propertyName",
NSSUBPROP_DISPLAYNAME).GetPtr(), "Input::GridMap::Layer"));
//ADTF3 Wunschverhalten:
m_oProperties.SetProperty(adtf::base::property<tChar*> (adtf_util::cString::Format("%s%s", "propertyName",
NSSUBPROP_DISPLAYNAME).GetPtr(), "Input::GridMap::Layer"));
```

Zum lesen benutze ich folgendes, welches einwandfrei funktioniert.

```
...
tVoid GetProperties()
{
    m_oProperties.GetProperty("propertyName", m_oIntProperty_tmp);
    nProp1 = m_oIntProperty_tmp.GetValueT();
}
...

tResult cPropertiesTestFilter::Init(tInitStage eStage)
{
    if (eStage == StageNormal)
    {
        m_pProps->GetProperties();
        LOG_INFO("Prop 1 = %u\n", m_pProps->nProp1);
    }
}
```

Die Frage wäre also, kann ich, wenn ich direkt auf den Properties arbeite, dort Sachen wie Min, Max, DisplayName, Description, IsChangeable, etc.

setzen?
Oder brauche ich dafür immer eine property_variable?

Viele Grüße
Jonas

#6 - 2021-05-18 14:00 - hidden

Hi Jonas,

wie oben in meinem Code Snippet beschrieben musst du die passenden SubProperties setzen:

```
object_ptr<IProperties> pProperties;  
THROW_IF_FAILED(GetProperties(pProperties));  
set_property_by_path<tInt32>(*pProperties, "myprop1", 123);  
set_property_by_path<tInt32>(*pProperties, (std::string("myprop1/") + adtf::base::giant::detail::  
tReservedProperties::strRangeMin).c_str(), 0);  
set_property_by_path<tInt32>(*pProperties, (std::string("myprop1/") + adtf::base::giant::detail::  
tReservedProperties::strRangeMax).c_str(), 200);
```

Grüße,

Martin

#7 - 2021-05-19 08:31 - hidden

Hi Martin,

perfekt! - Vielen Dank das funktioniert wunderbar.

Viele Grüße
Jonas

#8 - 2021-05-19 08:39 - hidden

- Status changed from Customer Feedback Required to To Be Closed
- Resolution set to Solved Issue

#9 - 2021-05-19 09:33 - hidden

- Subject changed from Frage zu ADTF3-IProperties-Interface to Question about properties interface
- Description updated

#10 - 2021-05-19 09:36 - hidden

- Subject changed from Question about properties interface to Question about the IProperties Interface

#11 - 2021-05-19 09:51 - hidden

Hallo zusammen,

noch eine kleine Frage hintendran, ich habe versucht "custom subproperties" zu erstellen, um quasi "changeable" properties abzubilden:

```
adtf::base::set_property_by_path<tBool>(m_oProperties, (std::string("myprop1").append("/#__is_changeable")).  
c_str(), tTrue);
```

diese werden in ADTF3.10 nicht angezeigt, gibt es dazu eine Möglichkeit, damit man dies sehen kann?

Viele Grüße
Jonas

#12 - 2021-05-19 09:52 - hidden

- Status changed from To Be Closed to In Progress

#14 - 2021-05-19 09:59 - hidden

Hi Jonas,

alles was mit "#__" anfängt wird nicht in die plugin description geschrieben, nimm einfach einen anderen Namen für die Subproperty.

Grüße,

Martin

#15 - 2021-05-20 07:05 - hidden

Hi Martin,

vielen Dank! Ticket kann jetzt wirklich geschlossen werden :)

Viele Grüße
Jonas

#16 - 2021-05-20 07:07 - hidden

- *Project changed from 4 to Public Support*
- *Status changed from In Progress to To Be Closed*
- *Private changed from Yes to No*

#17 - 2021-05-20 07:07 - hidden

- *Status changed from To Be Closed to Closed*

#18 - 2021-05-20 07:11 - hidden

- *Description updated*

Files

image001.png	16.8 KB	2021-05-17	hidden
--------------	---------	------------	--------