Public Support - Support Request #11007

No UI Widget when overriding Init of cQtUIFilter

2020-04-08 18:15 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	VW	**Product Issue Numbers:**	
Department:	EEIS	**Affected Products:**	ADTF 3.6.3
Requester's Priority:	Normal	**Platform:**	Windows 10 64bit
Support Level:	2nd Level	**Topic:**	ADTF::UI
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage

ich bin auf ein Problem beim hinzufügen einer GUi für einen Filter gestoßen. Wenn ich die Funktion "virtual tResult Init(tInitStage eStage) override;" in meinem Filter verwende, wird mir die GUI nicht länger angezeigt. Muss ich evtl. die Funktion CreateView() manuell in der Init aufrufen, oder kann man das nicht kombinieren?

Falls es nicht geht, gibt es eine andere Möglichkeit die Properties eines Filters über eine GUI zu verändern, auch wenn die GUI in einem externen Filter ist?

Lösung

sieh dir mal unser Bsp. Demo Qt Video Display Filter, hier wird auch die ::Ini@t üebrschrieben.
Wichtig dabei:
Du musst zu Beginn deiner Funktion die ::Init@ der Basisklasse cQtUIFilter aufrufen, siehe Implementierung im example:

```
RETURN_IF_FAILED(cQtUIFilter::Init(eStage));
```

Ansonsten hängt sich das Window nicht ins XSystem, das übernimmt die Basisklasse für dich.

Den Basis ::Init call musst du ja bei einen normalen cFilter Ableitung auch machen.
Steht auch so in der Doku -> cFilterBase::Init

Noch ein Hinweis zu deinem Use Case:
Im ADTF 3 legst du Property Variablen an und registrierst diese.
d.h. die haben immer den aktuellen Wert aus der Laufzeit.
Mit der 3.7.0 wird es auch ein Callback dafür geben.

History

#1 - 2020-04-09 10:22 - hidden

- Project changed from Public Support to 20

- Status changed from New to In Progress

- Topic set to ADTF::FilterSDK

#4 - 2020-04-09 10:54 - hidden

- Status changed from In Progress to Customer Feedback Required

- Topic changed from ADTF::FilterSDK to ADTF::UI

Hallo Jessica,

sieh dir mal unser Bsp. Demo Qt Video Display Filter, hier wird auch die ::Ini@t üebrschrieben.
Wichtig dabei:
Du musst zu Beginn deiner Funktion die ::Init@ der Basisklasse cQtUIFilter aufrufen, siehe Implementierung im example:

```
RETURN_IF_FAILED(cQtUIFilter::Init(eStage));
```

Ansonsten hängt sich das Window nicht ins XSystem, das übernimmt die Basisklasse für dich.

Den Basis ::Init call musst du ja bei einen normalen cFilter Ableitung auch machen.
Steht auch so in der Doku -> cFilterBase::Init

#5 - 2020-04-09 10:56 - hidden

Hallo, danke für den Hinweis, ich gucke es mir gleich an und gebe schnellstmöglich eine Rückmeldung ob es so funktioniert.

#6 - 2020-04-09 11:32 - hidden

Ich habe in einem Minimalbeispiel mal versucht, die Init-Funktion zum laufen zu bringen:

```
tResult training::cTrainingFilter::Init(tInitStage eStage)
{
    RETURN_IF_FAILED(cFilter::Init(eStage));
    return tResult();
}
```

Aber die Gui wird trotzdem nicht angezeit. Unten der Code von meinem Minimalbeispiel (Filter). Die Gui wird angezeigt sobald ich die Init() Funktion auskommentiere.

filter.h

```
#pragma once

#include "qt_ui_widget.h"
#include <adtf_filtersdk.h>
#include <adtf_ui.h>
//#include <hud_data.h>

using namespace adtf::mediadescription;
using namespace adtf::streaming;
using namespace adtf::ucom;
using namespace adtf::ui;
using namespace adtf::util;
using namespace adtf::base;
using namespace adtf::filter;

namespace training
{
    class cTrainingFilter : public cQtUIFilter
    {
    public:
        ADTF_CLASS_ID_NAME(cTrainingFilter, "mini_ui.cid", "Mini Qt UI Filter");
        ADTF_CLASS_DEPENDENCIES(REQUIRE_INTERFACE(adtf::ui::ant::IQtXSystem));

        cTrainingFilter();
        ~cTrainingFilter() override = default;

        virtual tResult Init(tInitStage eStage) override;
        virtual tResult Start() override;
        virtual tResult Stop() override;
        virtual tResult Process(tNanoSeconds tmTimeOfTrigger, IRunner* pRunner) override;
        virtual tResult ProcessInput(ISampleReader* pReader, const iobject_ptr<const ISample>& pSample)
override;

    protected:
        QWidget* CreateView() override;
        tVoid ReleaseView() override;
        tResult OnIdle() override;

    private:
        cTrainingWidget* m_pWidget = nullptr;
        //ISampleReader* m_pSampleReader = nullptr;
        //md_hud_data::tHUDData m_data{};
    };
}
```

filter.cpp:

```
#include "qt_ui_filter.h"

using namespace training;
//using namespace md_hud_data;

ADTF_PLUGIN("Mini Qt UI", cTrainingFilter)

cTrainingFilter::cTrainingFilter()
{

// TODO create an non-data-triggered input pin using CreateInputPin<cSingleSampleReader>("", description<tHUDD
ata>(), tFalse)
}

tResult training::cTrainingFilter::Init(tInitStage eStage)
{
    RETURN_IF_FAILED(cFilter::Init(eStage));
    RETURN_NOERROR;
    //return tResult();
}

tResult training::cTrainingFilter::Start()
{
    return tResult();
}

tResult training::cTrainingFilter::Stop()
{
    return tResult();
}

tResult training::cTrainingFilter::Process(tNanoSeconds tmTimeOfTrigger, IRunner * pRunner)
{
    return tResult();
}

tResult training::cTrainingFilter::ProcessInput(ISampleReader * pReader, const iobject_ptr<const ISample>&
pSample)
{
    return tResult();
}

QWidget* cTrainingFilter::CreateView()
{
    m_pWidget = new cTrainingWidget(nullptr);
    return m_pWidget;
}

tVoid cTrainingFilter::ReleaseView()
{
    m_pWidget = nullptr;
}

tResult cTrainingFilter::OnIdle()
{
    RETURN_IF_POINTER_NULL(m_pWidget);
    //RETURN_IF_POINTER_NULL(m_pSampleReader);

    RETURN_NOERROR;
}
```

Liebe Grüße,
Jessica

#7 - 2020-04-09 11:39 - hidden

- Status changed from Customer Feedback Required to In Progress

#8 - 2020-04-09 12:06 - hidden

- Status changed from In Progress to Customer Feedback Required

Hi Jessica,

wie gesagt, du musst immer die Funktion der Basisklasse von der du ableitest nochmal zu Beginn aufrufen.

Wenn du also von cQtUIFilter ableitest, dann musst du auch:

```
RETURN_IF_FAILED(cQtUIFilter::Init(eStage));
```

aufrufen.

Siehe auch das verlinkte Example.

Du rufst aber

```
RETURN_IF_FAILED(cFilter::Init(eStage));
```

Das brauchst du natürlich bei einen "normalen" Filter, wenn du von cFilter ableitest.

Noch ein Hinweis zu deinem Use Case:
Im ADTF 3 legst du Property Variablen an und registrierst diese.
d.h. die haben immer den aktuellen Wert aus der Laufzeit.
Mit der 3.7.0 wird es auch ein Callback dafür geben.

Aber ja, ansonsten ist die Init die richtige Methode, wenn du bei der Initialisierung eine Bedatung vorzunehmen hast.

#9 - 2020-04-09 12:26 - hidden

Ah, jetzt habe ich gesehen, was ich falsch gemacht habe... habe den Un terschied beim ersten mal überlesen.
Danke, jetzt funktioniert alles.

#10 - 2020-04-09 13:42 - hidden

- Project changed from 20 to Public Support

- Subject changed from cQtUIFilter Widget wird nicht angezeigt, sobald die Init Funktion verwendet wird to No UI Widget when overriding Init of cQtUIFilter

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Resolution set to Solved Issue

#11 - 2020-04-09 13:42 - hidden

- Private changed from Yes to No

#14 - 2020-07-07 12:49 - hidden

- Status changed from To Be Closed to Closed