

单体测试总结

+ mock的方法返回的是一个指针

```
{
    VR_LOG("VR_SDSStateLocal Enter StateName:%s\n", GetStateName().GetString());
    if (this == GetStateMachine()->GetRootState()) {
        CancelRequest();
        Clear();
        VR_LOG(" To stop Handler-Timer on stateName [%s] !\n", GetStateName().GetString());
        GetStateMachine()->StopHandlerTimer();
        return ;
    }
    SetReqID(VR_INVALID_REQUET_ID);
    VR_SDSEvent rcEvent(VRSDSCONFIG_NULL_DEF, VRSDSCONFIG_NULL_DEF,
VR_SDS_EVENT_TYPE_SDSINSIDE);
    if (m_cActionVector.empty()) {
        OnEvent(rcEvent);
        return;
    }
    for (WORD i = 0; i < m_cActionVector.size(); ++i) {
        if (NULL != m_cActionVector[i]) {
            if (m_cActionVector[i]->DoAction(this, rcEvent)) {
                break;
            }
        }
    }
}
}
```

mock 方法

```
TEST_F(VR_SDSStateLocal_Test, Enter_RootState)
{
    m_spStateLocal->SetReqID(1);
    VR_SDSStateLocal* pStateLocal = m_spStateLocal.get();
    m_spMockSDSStateMachine->SetStateManager(m_spMockSDSStateManager);
    EXPECT_CALL(*m_spMockSDSStateManager,
GetRootState()).WillRepeatedly(ReturnPointee(&pStateLocal));
    EXPECT_CALL(*m_spMockSDSStateMachine, StopHandlerTimer()).Times(1);
    pStateLocal->Enter();
}
```

+ mock的方法返回的是一个引用

```
virtual VR_SessionInfo &GetSession();
```

mock 方法

```
cSessionInfo.m_cId = "1";
EXPECT_CALL(*m_spMockSDSStateMachine,
GetSession()).WillOnce(testing::ReturnRef(cSessionInfo));
```

+ 纯虚函数才可以mock

+ gmock方法

MOCK_METHOD#1(#2, #3(#4)) #1表示你要mock的方法共有几个参数 #2是你想要mock的方法名称 #3表示这个方法的返回值类型 #4是这个方法具体的参数

ON_CALL(#1, #2(#3)).WillByDefault(Return(#4)); #1表示mock对象 #2表示想定义的那个方法名称。 #3表示定义方法的参数，只有当传入的参数=#3的值时，才会生效，也可用_，表示匹配任何参数输入都生效 #4表示调用要返回值。

除了ON_CALL以外，还有EXPECT_CALL(#1, #2(#3))，参数含义同ON_CALL，还可以有额外的功能，比如：
EXPECT_CALL(#1, #2(#3)).Times(5).WillOnce(Return(100)).WillOnce(Return(150)).WillRepeatedly(Return(200)); 表示第一次返回100，第二次调用返回150，后面全部返回200，以此类推。

ON_CALL和EXCEPT_CALL相当于设置桩函数的属性设置，因此应当在桩函数调用前进行设置。除了TEST宏函数定义和main函数之外，在例子中可以看到我还定义了SubServiceCall，由于这个函数会向Fbase发送服务调用请求，但我们用MockF2Pack打的包其实是空无一物，调用服务显然会使测试过程不可控，因此我们把这个函数也进行打桩。

+ 析构函数为纯虚

在某些类里声明纯虚析构函数很方便。纯虚函数将产生抽象类——不能实例化的类（即不能创建此类型的对象）。有些时候，你想使一个类成为抽象类，但刚好又没有任何纯虚函数。怎么办？因为抽象类是准备被用作基类的，基类必须要有一个虚析构函数，纯虚函数会产生抽象类，所以方法很简单：在想要成为抽象类的类里声明一个纯虚析构函数。

例子

```
{
typedef VR_SessionTimer VR_HandlerTimer;
public:
    CStateMachine();
    CStateMachine(boost::shared_ptr<VR_SDSMultiLanguage> spcMultiLanguage);
    virtual ~CStateMachine() = 0;
}
```

mock方法

```
{
public:
```

```

    My_StateMachine(){}
    ~My_StateMachine(){}
};
class StateMachine_Test : public testing::Test
{
public:
    static void SetUpTestCase()
    {
    }
    static void TearDownTestCase()
    {
    }
};
TEST_F(StateMachine_Test, ClearSession)
{
    My_StateMachine cStateMachine;
    cStateMachine.ClearSession();
}
TEST_F(StateMachine_Test, GetSession)
{
    My_StateMachine cStateMachine;
    cStateMachine.GetSession();
}

```

+ExceptCall time(0)

例子

```

void VR_RequestZoomMapView::OnDelete()
{
    VR_LOG("id=[%d],m_pcListener=[%p],m_bCancel=[%d]",
        this->GetRequestID(), m_pcListener, m_bCancel);
    if ((! m_bCancel) && (NULL != m_pcListener))
    {
        m_pcListener->OnZoomMapResult(this->GetRequestID(), m_iResult, m_zoomMapResult,
            m_zoomMapDirection);
    }
}

```

mock 方法

```

TEST_F(VR_RequestZoomMapView_test, OnDelete_CancelTure)
{
    m_spZoomMapView->m_bCancel = true;
    EXPECT_CALL(*m_spMockVRLListenerBase, OnZoomMapResult(testing::__, testing::__,
        testing::__, _)).Times(0);
    m_spZoomMapView->OnDelete();
}

TEST_F(VR_RequestZoomMapView_test, OnDelete_NullListener)

```

```
{
    m_spZoomMapView->m_pcListener = NULL;
    EXPECT_CALL(*m_spMockVRListenerBase, OnZoomMapResult(testing::__, testing::__,
testing::__, _)).Times(0);
    m_spZoomMapView->OnDelete();
}

TEST_F(VR_RequestZoomMapView_test, OnDelete_CancelFalseListener)
{
    m_spZoomMapView->m_bCancel = false;
    m_spZoomMapView->m_pcListener = m_spMockVRListenerBase.get();
    EXPECT_CALL(*m_spMockVRListenerBase, OnZoomMapResult(testing::__, testing::__,
testing::__, _)).Times(1);
    m_spZoomMapView->OnDelete();
}
```