Contents

1 Main Page ........................................ 1

2 Test List ........................................ 3

3 File Index ....................................... 13
   3.1 File List ..................................... 13

4 File Documentation ................................ 15
   4.1 mainpagetests/01_mainpage.dox File Reference ........................................ 15
   4.2 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
       CMAtributes.cpp File Reference ........................................ 15
       4.2.1 Function Documentation ........................................ 15
       4.2.1.1 CMAtributes_AttributeGetSynchronous ........................................ 15
       4.2.1.2 CMAtributes_AttributeGetAsynchronous ........................................ 16
       4.2.1.3 CMAtributes_AttributeSetSynchronous ........................................ 16
       4.2.1.4 CMAtributes_AttributeSetAsynchronous ........................................ 16
       4.2.1.5 CMAtributes_AttributeSubscription ........................................ 16
       4.2.1.6 main ........................................ 16
       4.2.2 Variable Documentation ................................. 16
       4.2.2.1 serviceId ........................................ 16
       4.2.2.2 clientId ........................................ 16
       4.2.2.3 domain ........................................ 17
       4.2.2.4 testAddress ........................................ 17
       4.2.2.5 tasync ........................................ 17
   4.3 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
       CMAtributeSubscription.cpp File Reference ........................................ 17
       4.3.1 Typedef Documentation ........................................ 18
       4.3.1.1 ProxyPtr ........................................ 18
       4.3.2 Function Documentation ........................................ 18
       4.3.2.1 testSubscription ........................................ 18
       4.3.2.2 CMAtributeSubscription_SubscriptionStandard ........................................ 18
       4.3.2.3 CMAtributeSubscription_SubscriptionOnAvailable ........................................ 18
       4.3.2.4 CMAtributeSubscription_SubscriptionMultithreading ........................................ 18
| 4.3.2.5 | CMAtributeSubscription_SubscriptionUnsubscribeFromCallback | 18 |
| 4.3.2.6 | CMAtributeSubscription_SubscribeAndUnsubscribeTwoCallbacksCoexistent | 19 |
| 4.3.2.7 | CMAtributeSubscription_SubscribeAndUnsubscribeImmediatelyUnsubscribing | 19 |
| 4.3.2.8 | CMAtributeSubscription_SubscribeAndUnsubscribeSequentially | 19 |
| 4.3.2.9 | CMAtributeSubscription_SubscribeAndUnsubscribeUnsubscribe | 20 |
| 4.3.2.10 | CMAtributeSubscription_SubscribeServiceNotAvailable | 20 |
| 4.3.2.11 | CMAtributeSubscription_SubscribeUnregisterSetValueRegisterService | 20 |
| 4.3.2.12 | CMAtributeSubscription_SubscribeUnregisterNoValueSetRegisterService | 21 |
| 4.3.2.13 | CMAtributeSubscription_SubscribeSecondProxyLater | 21 |
| 4.3.2.14 | CMAtributeSubscription_SubscribeThreeCallbacksServiceNotAvailable | 21 |
| 4.3.2.15 | CMAtributeSubscription_SubscribeThreeCallbacksServiceAvailable | 21 |
| 4.3.2.16 | main | 22 |
| 4.3.3 | Variable Documentation | 22 |
| 4.3.3.1 | daemonId | 22 |
| 4.3.3.2 | clientId | 22 |
| 4.3.3.3 | serviceld | 22 |
| 4.3.3.4 | domain | 22 |
| 4.3.3.5 | testAddress | 22 |
| 4.3.3.6 | daemonAddress | 22 |
| 4.3.3.7 | wt | 22 |
| 4.3.3.8 | mut | 22 |
| 4.3.3.9 | data_queue | 22 |
| 4.3.3.10 | data_cond | 22 |
| 4.4 | /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.coreverification/src/-CMMethodCalls.cpp File Reference | 22 |
| 4.4.1 | Function Documentation | 22 |
| 4.4.1.1 | CMMethodCalls_SynchronousMethodCall | 22 |
| 4.4.1.2 | CMMethodCalls_AsynchronousMethodCall | 23 |
| 4.4.1.3 | main | 23 |
| 4.4.2 | Variable Documentation | 23 |
| 4.4.2.1 | serviceld | 23 |
| 4.4.2.2 | clientId | 23 |
| 4.4.2.3 | domain | 23 |
| 4.4.2.4 | testAddress | 23 |
| 4.4.2.5 | tasync | 23 |
| 4.5 | /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.coreverification/src/-DTAdvanced.cpp File Reference | 23 |
| 4.5.1 | Function Documentation | 23 |
| 4.5.1.1 | DTAdvanced_SendAndReceive | 23 |
| 4.5.1.2 | DTAdvanced_AttributeSet | 23 |
4.5.1.3 DTAdvanced_BroadcastReceive ........................................... 24
4.5.1.4 main ................................................................. 24
4.5.2 Variable Documentation ..................................................... 24
  4.5.2.1 domain ........................................................... 24
  4.5.2.2 testAddress ....................................................... 24
  4.5.2.3 connectionIdService .............................................. 24
  4.5.2.4 connectionIdClient .............................................. 24
4.6 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
  DTCombined.cpp File Reference .............................................. 24
  4.6.1 Function Documentation ............................................... 24
    4.6.1.1 DTCombined_SendAndReceive .................................. 24
    4.6.1.2 main ........................................................... 25
  4.6.2 Variable Documentation ............................................... 25
    4.6.2.1 domain ........................................................ 25
    4.6.2.2 testAddress ................................................... 25
    4.6.2.3 connectionIdService ......................................... 25
    4.6.2.4 connectionIdClient ......................................... 25
4.7 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
  DTDerived.cpp File Reference ............................................... 25
  4.7.1 Function Documentation ............................................... 25
    4.7.1.1 DTDerived_SendAndReceive .................................. 25
    4.7.1.2 DTDerived_AttributeSet ...................................... 25
    4.7.1.3 DTDerived_BroadcastReceive ................................ 25
    4.7.1.4 main ........................................................... 26
  4.7.2 Variable Documentation ............................................... 26
    4.7.2.1 domain ........................................................ 26
    4.7.2.2 testAddress ................................................... 26
    4.7.2.3 connectionId_client ......................................... 26
    4.7.2.4 connectionId_service ........................................ 26
4.8 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
  DTPrimitive.cpp File Reference ............................................. 26
  4.8.1 Function Documentation ............................................... 26
    4.8.1.1 DTPrimitive_SendAndReceive ................................ 26
    4.8.1.2 DTPrimitive_AttributeSet .................................... 26
    4.8.1.3 DTPrimitive_BroadcastReceive ................................ 27
    4.8.1.4 main ........................................................... 27
  4.8.2 Variable Documentation ............................................... 27
    4.8.2.1 domain ........................................................ 27
    4.8.2.2 testAddress ................................................... 27
    4.8.2.3 connectionIdService ......................................... 27
    4.8.2.4 connectionIdClient ......................................... 27
4.9 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src-/PFComplex.cpp File Reference

4.9.1 Function Documentation

4.9.1.1 PFComplex_Ping_Pong_Complex_Synchronous

4.9.1.2 PFComplex_Ping_Pong_Complex_Asynchronous

4.9.1.3 main

4.9.2 Variable Documentation

4.9.2.1 usecPerSecond

4.9.2.2 serviceld

4.9.2.3 clientId

4.9.2.4 domain

4.9.2.5 testAddress

4.9.2.6 tasync

4.9.2.7 maxArraySize

4.9.2.8 loopCountPerPayload

4.10 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src-/PFPrimitive.cpp File Reference

4.10.1 Function Documentation

4.10.1.1 PFPrimitive_Ping_Pong_Primitive_Synchronous

4.10.1.2 PFPrimitive_Ping_Pong_Primitive_Asynchronous

4.10.1.3 main

4.10.2 Variable Documentation

4.10.2.1 serviceld

4.10.2.2 clientId

4.10.2.3 domain

4.10.2.4 testAddress

4.10.2.5 tasync

4.10.2.6 usecPerSecond

4.10.2.7 maxPrimitiveArraySize

4.10.2.8 loopCountPerPayload

4.11 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src-/RTBuildProxiesAndStubs.cpp File Reference

4.11.1 Function Documentation

4.11.1.1 RTBuildProxiesAndStubs_LoadedRuntimeCanBuildProxiesAndStubs

4.11.1.2 RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes

4.11.1.3 main

4.11.2 Variable Documentation

4.11.2.1 domain

4.11.2.2 testAddress

4.11.2.3 applicationNameService

4.11.2.4 applicationNameClient
4.12 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
  RTLoadingRuntime.cpp File Reference .................................................. 30
  4.12.1 Function Documentation ............................................................ 31
    4.12.1.1 RTLoadingRuntime_LoadsDefaultRuntime .................................. 31
    4.12.1.2 main ................................................................................. 31
  4.13 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
  StabilitySP.cpp File Reference .............................................................. 31
  4.13.1 Function Documentation ............................................................... 31
    4.13.1.1 StabilitySP_RepeatedRegistrations ........................................ 31
    4.13.1.2 StabilitySP_MultipleMethodCalls .......................................... 32
    4.13.1.3 StabilitySP_MultipleAttributeSets ........................................ 32
    4.13.1.4 StabilitySP_MultipleAttributeGets ........................................ 32
    4.13.1.5 StabilitySP_MultipleAttributeGetAsyncs .................................. 32
    4.13.1.6 StabilitySP_MultipleAttributeSetAsyncs .................................. 33
    4.13.1.7 StabilitySP_MultipleAttributeSubscriptions ............................. 33
    4.13.1.8 main ................................................................................. 33
    4.13.2 Variable Documentation ................................................................ 33
      4.13.2.1 serviceld ........................................................................ 33
      4.13.2.2 clientId .......................................................................... 33
      4.13.2.3 domain ........................................................................... 33
      4.13.2.4 testAddress ...................................................................... 33
      4.13.2.5 COMMONAPI_CONFIG_SUFFIX ........................................... 33
      4.13.2.6 MAXSERVERCOUNT ............................................................ 33
      4.13.2.7 MAXTHREADCOUNT .......................................................... 33
      4.13.2.8 MAXMETHODCALLS ........................................................... 33
      4.13.2.9 MAXREGLOOPS ................................................................. 33
      4.13.2.10 MAXREGCOUNT ............................................................... 33
      4.13.2.11 MESSAGESIZE ................................................................. 33
      4.13.2.12 MAXSUBSCRIPTIONSETS ................................................ 33
  4.14 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
  THMainLoopIndependence.cpp File Reference ........................................... 34
  4.14.1 Function Documentation ............................................................... 34
    4.14.1.1 THMainLoopIndependence_ProxyReceivesAnswerOnlyIfStubMainLoopRuns . 34
    4.14.1.2 THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers ........... 34
    4.14.1.3 main ................................................................................. 34
    4.14.2 Variable Documentation ............................................................. 34
      4.14.2.1 domain ........................................................................... 34
      4.14.2.2 instance6 ......................................................................... 34
      4.14.2.3 instance7 ......................................................................... 34
      4.14.2.4 instance8 ......................................................................... 34
      4.14.2.5 mainloopName1 ................................................................. 35
4.14.2.6 mainloopName2 ................................................................. 35
4.14.2.7 thirdPartyServiceId ....................................................... 35

4.15 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
THMainLoopIntegration.cpp File Reference ......................................... 35
4.15.1 Function Documentation ..................................................... 35
  4.15.1.1 THMainLoopIntegration_VerifyCommunicationWithMainLoop .......... 35
  4.15.1.2 THMainLoopIntegration_VerifyTransportReading ....................... 35
  4.15.1.3 THMainLoopIntegration_VerifySyncCallMessageHandlingOrder ........ 36
  4.15.1.4 main ............................................................................. 36
4.15.2 Variable Documentation ...................................................... 36
  4.15.2.1 domain ......................................................................... 36
  4.15.2.2 instance ........................................................................ 36
  4.15.2.3 connection_client .......................................................... 36
  4.15.2.4 connection_service ........................................................ 36

4.16 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-
THMainLoopTwoThreads.cpp File Reference ........................................... 36
4.16.1 Function Documentation ..................................................... 36
  4.16.1.1 THMainLoopTwoThreads_ProxyGetsAvailableStatus .................. 36
  4.16.1.2 THMainLoopTwoThreads_ProxyGetsFunctionResponse ............... 36
  4.16.1.3 main ............................................................................. 37
4.16.2 Variable Documentation ...................................................... 37
  4.16.2.1 domain ......................................................................... 37
  4.16.2.2 instance ........................................................................ 37

Index ................................................................................................. 38
Chapter 1

Main Page

Copyright (C) 2015 BMW AG

• This file is part of GENIVI project IPC CommonAPI C++.

Contributions are licensed to the GENIVI Alliance under one or more Contribution License Agreements.

This document

This document provides a list of tests which are implemented in the project +org.genivi.commonapi.core-\verification+ which is part of CommonAPI-Tools. These tests are middleware independent and can be used to verify the correct implementation of middleware specific bindings.

About IPC CommonAPI C++

IPC CommonAPI C++ is a C++ based abstraction API for communication stacks, which enables applications to use different communication middleware - so called language bindings - as backend without any changes to the application code.

More information

can be found at the project homepage
Please see the project download section for available language bindings.
Chapter 2

Test List

Global `CMAtributes_AttributeGetAsynchronous()`
Test asynchronous getValue API function for attributes with combinations of additional properties readonly and 
noSubscriptions (testAttribute, testA readonly, testB noSubscriptions, testC readonly noSubscriptions).

- Set attribute to certain value on stub side.
- Call getValue.
- Check if returned call status is `CommonAPI::CallStatus::SUCCESS`.
- Check if value of is equal to expected value.

Global `CMAtributes_AttributeGetSynchronous()`
Test synchronous getValue API function for attributes with combinations of additional properties readonly and 
noSubscriptions (testAttribute, testA readonly, testB noSubscriptions, testC readonly noSubscriptions).

- Set attribute to certain value on stub side.
- Call getValue.
- Check if returned call status is `CommonAPI::CallStatus::SUCCESS`.
- Check if value of is equal to expected value.

Global `CMAtributes_AttributeSetAsynchronous()`
Test asynchronous setValue API function for attributes with combinations of additional properties readonly and 
noSubscriptions (testAttribute, testB noSubscriptions).

- Set attribute to certain value on proxy side.
- Check if returned call status is `CommonAPI::CallStatus::SUCCESS`.
- Check if returned value of setValue is equal to expected value.

Global `CMAtributes_AttributeSetSynchronous()`
Test synchronous setValue API function for attributes with combinations of additional properties readonly and 
noSubscriptions (testAttribute, testB noSubscriptions)

- Set attribute to certain value on proxy side.
- Check if returned call status is `CommonAPI::CallStatus::SUCCESS`.
- Check if returned value of setValue is equal to expected value.

Global `CMAtributes_AttributeSubscription()`
Test subscription API function for attributes

Global `CMAtributeSubscription_SubscribeAndUnsubscribeImmediatelyUnsubscribing()`
Test of subscribing and immediately unsubscribing a callback

- subscribe first callback
- subscribe second callback
• unsubscribe second callback
• change value
• check that only first callback was executed

Global `CMAttributeSubscription_SubscribeAndUnsubscribeSequentially()`
Test of subscribing and immediately sequentially

• subscribe first callback
• subscribe second callback
• change value
• check that both callbacks were executed by changing the value
• unsubscribe first callback
• change value
• check that only second callback was executed
• unsubscribe second callback
• change value
• check that both callbacks were not executed by changing the value

Global `CMAttributeSubscription_SubscribeAndUnsubscribeTwoCallbacksCoexistent()`
Test of subscribe and unsubscribe with two coexistent callbacks

• subscribe both callbacks
• change value
• check that both callbacks were executed by changing the value
• unsubscribe both callbacks
• change value
• check that both callbacks were not executed by changing the value

Global `CMAttributeSubscription_SubscribeAndUnsubscribeUnsubscribe()`
Test of behaviour in case unsubscribe is called two times

• set default value
• register service
• subscribe for the attribute
• current value must be communicated to the proxy
• value of attribute is changed
• changed value must be communicated to the proxy
• proxy unsubscribes for the attribute
• value of attribute is changed
• changed value must not be communicated to the proxy
• proxy unsubscribes again for the attribute
• value of attribute is changed
• changed value must not be communicated to the proxy
• unregister service

Global `CMAttributeSubscription_SubscribeSecondProxyLater()`
Test of subscribing a second proxy a little bit later

• proxy subscribes for an attribute of the service
• register service
• initial value must be communicated to the proxy
• create a second proxy
• second proxy subscribes for the same attribute of the service
• current attribute value must be communicated to the proxy
• value of attribute is changed
• changed value must be communicated to both proxies
• unregister service

Global CMAtributeSubscription_SubscribeServiceNotAvailable ()
Test of subscribing in case that service is not available

• set default value
• subscribe for the attribute
• no value is communicated to the proxy
• register service
• current value must be communicated to the proxy
• value of attribute is changed
• changed value must be communicated to the proxy
• unregister service

Global CMAtributeSubscription_SubscribeThreeCallbacksServiceAvailable ()
Test of subscribing three callbacks after registering the service

• register service
• proxy subscribes three callbacks for an attribute of the service
• initial value must be communicated to every callback

Global CMAtributeSubscription_SubscribeThreeCallbacksServiceNotAvailable ()
Test of subscribing three callbacks before registering the service

• proxy subscribes three callbacks for an attribute of the service
• register service
• initial value must be communicated to every callback

Global CMAtributeSubscription_SubscribeUnregisterNoValueSetRegisterService ()
Test of unregister a service in case a proxy is subscribed for an attribute of this service. During the unregistered
time of the service the value of the attribute is not changed.

• register service
• proxy subscribes for an attribute of the service
• value of attribute is set
• changed value must be communicated to the proxy
• unregister service
• register service
• current attribute value must be communicated to the proxy
• value of attribute is changed
• changed value must be communicated to the proxy
• unregister service

Global CMAtributeSubscription_SubscribeUnregisterSetValueRegisterService ()
Test of unregister a service in case a proxy is subscribed for an attribute of this service. During the unregistered
time of the service the value of the attribute is changed.

• register service
• proxy subscribes for an attribute of the service
• value of attribute is set
- changed value must be communicated to the proxy
- unregister service
- value of attribute is changed
- changed value must not be communicated to the proxy
- register service
- current attribute value must be communicated to the proxy
- value of attribute is changed
- changed value must be communicated to the proxy
- unregister service

Global CMAtributeSubscription_SubscriptionMultithreading()
Subscription test with several threads.
- Start several threads.
- The threads subscribe for the availability status.
- The available-callback subscribes for TestAttribute if service is available for proxy and unsubscribes if service is not available for proxy.
  - Change attribute in service by set method; the new attribute value should be received by all the threads.
  - The new value is written into a queue.
  - Check if the values of each thread are written into the queue.

Global CMAtributeSubscription_SubscriptionOnAvailable()
Subscription test with subscription on available-event.
- Subscribe for available-event.
- Available-callback subscribes for TestPredefinedTypeAttribute if service is available for proxy and unsubscribes if service is not available for proxy.
  - Change attribute in service by set method; the new attribute value should be received by the proxy because the service is not registered.
  - Register service and change value again; the value should now be received.
  - Unregister and change value again.

Global CMAtributeSubscription_SubscriptionStandard()
Subscription standard test.
- Register service and check if proxy is available.
- Proxy subscribes for TestAttribute (uint8_t).
  - Change attribute in service several times by set method.
  - Callback function in proxy writes the received values in a queue.
  - Check if values in the queue are the same as the values that were set in the service.
  - Unregister test service.

Global CMAtributeSubscription_SubscriptionUnsubscribeFromCallback()
Subscription test: unsubscribe from the subscription callback.
- Register service and check if proxy is available.
- Proxy subscribes for TestAttribute (uint8_t).
  - Change attribute in service by set method.
  - Check if callback function in proxy received the right value.
  - Change value to the magic value 99: this triggers the callback to unsubscribe.
  - Change value again; the callback should now be called anymore.
  - Unregister the test service.
Global `CMMethodCalls_AsynchronousMethodCall()`
Call test method asynchronous and check call status.
- Test stub sets in-value of test method equal out-value of test method.
- Make asynchronous call of test method.
- Do checks of call status (CommonAPI::CallStatus::SUCCESS) and returned value in callback function.

Global `CMMethodCalls_SynchronousMethodCall()`
Call test method synchronous and check call status.
- Test stub sets in-value of test method equal out-value of test method.
- Make synchronous call of test method.
- Check if returned call status is CommonAPI::CallStatus::SUCCESS.
- Check if out value of test method is equal to in value.

Global `DTAdvanced_AttributeSet()`
Test attribute functions with advanced types
- Call set function of attributes with advanced types
- Call get function and check if the return value is the same

Global `DTAdvanced_BroadcastReceive()`
Test broadcast with advanced types
- Subscribe to broadcast which contains advanced types
- Call function to cause the stub to fire broadcast event with the same content
- Check if the values in the callback function are as expected

Global `DTCombined_SendAndReceive()`
Test function call with combined type
- The combined type is one structure with combinations of advanced and primitive types
- Function call of a function that has for each advanced type one argument (test values) and one return value
- The stub copies the test values to the return values
- On client side the test values are compared with the return values

Global `DTDerived_AttributeSet()`
Test attribute functions with derived types
- Call set function of attributes with derived types
- Call get function and check if the return value is the same

Global `DTDerived_BroadcastReceive()`
Test broadcast with derived types
- Subscribe to broadcast which contains derived types
- Call function to cause the stub to fire broadcast event with the same content
- Check if the values in the callback function are as expected

Global `DTPrimitive_AttributeSet()`
Test attribute functions with primitive types
- Call set function of attributes with primitive types
- Call get function and check if the return value is the same

Global `DTPrimitive_BroadcastReceive()`
Test broadcast with primitive types
- Subscribe to broadcast which contains primitive types
• Call function to cause the stub to fire broadcast event with the same content
• Check if the values in the callback function are as expected

Global DTPrimitive_SendAndReceive ()
Test function call with primitive types
  • Primitive types are: uint8_t, int8_t, uint16_t, int16_t, uint32_t, int32_t, uint64_t, int64_t, bool, float, double,
    std::string, ByteBuffer
  • Function call of a function that has for each primitive type one argument (test values) and one return value
  • The stub copies the test values to the return values
  • On client side the test values are compared with the return values

Global PFComplex_Ping_Pong_Complex_Asynchronous ()
Test asynchronous ping pong function call
  • complex array is array of a struct containing an union and another struct with primitive datatypes
    – The stub just set (copies) the in array to the out array
    – Only the CallStatus will be used to verify the async call has succeeded
    – Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
    – Doing loopCountPerPaylod loops to calc the mean time

Global PFComplex_Ping_Pong_Complex_Synchronous ()
Test synchronous ping pong function call
  • complex array is array of a struct containing an union and another struct with primitive datatypes
    – The stub just set the in array to the out array
    – CallStatus and array content will be used to verify the sync call has succeeded
    – Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
    – Doing primitiveLoopSize loops to build the mean time

Global PFPrimitive_Ping_Pong_Primitive_Asynchronous ()
Test asynchronous ping pong function call
  • primitive array is array of UInt_8
    – The stub just set (copies) the in array to the out array
    – Only the CallStatus will be used to verify the async call has succeeded
    – Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
    – Doing primitiveLoopSize loops to build the mean time

Global PFPrimitive_Ping_Pong_Primitive_Synchronous ()
Test synchronous ping pong function call
  • primitive array is array of UInt_8
    – The stub just set the in array to the out array
    – CallStatus and array content will be used to verify the sync call has succeeded
    – Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
    – Doing primitiveLoopSize loops to build the mean time

Global RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes ()
Loads Runtime, creates proxy and stub/service two times.
  • Calls CommonAPI::Runtime::get() and checks if return value is true
  • Create stub and register service
  • Create proxy
  • Do some synchronous calls
  • Unregister the service.
  • Create stub and register service
• Create proxy
• Checks whether proxy is available
• Unregister the service

Global RTBuildProxiesAndStubs_LoadedRuntimeCanBuildProxiesAndStubs ()
Loads Runtime, creates proxy and stub/service.
• Calls CommonAPI::Runtime::get() and checks if return value is true.
• Checks if test proxy with domain and test instance can be created.
• Checks if test stub can be created.
• Register the test service.
• Unregister the test service.

Global RTLoadingRuntime_LoadsDefaultRuntime ()
Loads Default Runtime.
• Calls CommonAPI::Runtime::get().
• Success if return value is true.

Global StabilitySP_MultipleAttributeGetAsyncs ()
Create a number of services and proxies and get attributes through them.
• Register MAXSERVERCOUNT addresses as services
  – Set the attribute for service, at the stub side.
• Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then gets attributes MAXMETHODCALLS times for each asynchronously
• Each attribute is MESSAGESIZE bytes long.
• Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the callbacks are not called correct number of times

Global StabilitySP_MultipleAttributeGets ()
Create a number of services and proxies and get attributes through them.
• Register MAXSERVERCOUNT addresses as services
  – Set the attribute for service, at the stub side.
• Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then gets attributes MAXMETHODCALLS times for each.
• Each attribute is MESSAGESIZE bytes long.
• Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the returned attribute from the server is not correct

Global StabilitySP_MultipleAttributeSetAsyncs ()
Create a number of services and proxies and set attributes through them.
• Register MAXSERVERCOUNT addresses as services
  – Set the attribute for service, at the stub side.
• Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times for each asynchronously
• Each attribute is MESSAGESIZE bytes long.
• Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the callbacks are not called correct number of times

Global StabilitySP_MultipleAttributeSets ()
Create a number of services and proxies and set attributes through them.
• Register MAXSERVERCOUNT addresses as services
Test List

- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times to each.
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the return attribute from the server is not correct

Global **StabilitySP_MultipleAttributeSubscriptions ()**
Create a number of services and proxies and set attributes through them.

- Register MAXSERVERCOUNT addresses as services
  - Set the attribute for service, at the stub side.
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times for each asynchronously
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the callbacks are not called correct number of times

Global **StabilitySP_MultipleMethodCalls ()**
Create a number of services and proxies and send messages through them.

- Register MAXSERVERCOUNT addresses as services
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sends MAXMETHODCALLS messages to each.
- Each message is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the return message from the server is not correct

Global **StabilitySP_RepeatedRegistrations ()**
Register and unregister services in a loop.

- do MAXREGLOOPS times:
  - register MAXREGCOUNT addresses as services
  - unregister the addresses that were just registered
  - check the return code of each register/unregister call
  - test fails if any of the return codes are false

Global **THMainLoopIndependence_ProxyReceivesAnswerOnlyIfStubMainLoopRuns ()**
Proxy Receives Answer Only If Stub MainLoop Runs.

- start proxy in thread 1 and call testPredefinedTypeMethod
- proxy should not receive answer, if the stub mainloop does not run
- run mainloop of stub
- now the stub mainloop also runs, so the proxy should receive the answer

Global **THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers ()**
Proxy Receives Just His Own Answers.

- start 2 proxies in own threads
- call test method in each proxy
- now each proxy should have received the answer to his own request

Global **THMainLoopIntegration_VerifyCommunicationWithMainLoop ()**
Verifies Transport Reading When Dispatching Watches.

- get proxy with available flag = true
- generate big test data
• send asynchronous test message
• dispatch dispatchSource: the message must not be arrived
• dispatch watches (reads transport).
• dispatch dispatchSources again: now the message must be arrived.

Global THMainLoopIntegration_VerifySyncCallMessageHandlingOrder ()
Verifies Synchronous Call Message Handling Order.
• get proxy with available flag = true
• subscribe for broadcast event
• generate 5 test broadcasts
• 5 broadcasts should arrive in the right order

Global THMainLoopIntegration_VerifyTransportReading ()
Verifies Transport Reading When Dispatching Watches.
• get proxy with available flag = true
• generate big test data
• send asynchronous test message
• dispatch dispatchSource: the message must not be arrived
• dispatch watches (reads transport).
• dispatch dispatchSources again: now the message must be arrived.

Global THMainLoopTwoThreads_ProxyGetsAvailableStatus ()
Proxy Receives Available when MainLoop Dispatched sourced out to other thread.

Global THMainLoopTwoThreads_ProxyGetsFunctionResponse ()
Proxy gets function response when MainLoop Dispatched sourced out to other thread.
Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/CM-Attributes.cpp ................................................................. 15
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/CM-AttributeSubscription.cpp ................................................................. 17
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/CM-MethodCalls.cpp ................................................................. 22
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/DT-Advanced.cpp ................................................................. 23
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/DT-Combined.cpp ................................................................. 24
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/DT-Derived.cpp ................................................................. 25
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/DT-Primitive.cpp ................................................................. 26
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/PF-Complex.cpp ................................................................. 27
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/PF-Primitive.cpp ................................................................. 28
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/RT-BuildProxiesAndStubs.cpp ................................................................. 29
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/RT-LoadingRuntime.cpp ................................................................. 30
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/Stability-SP.cpp ................................................................. 31
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/TH-MainLoopIndependence.cpp ................................................................. 34
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/TH-MainLoopIntegration.cpp ................................................................. 35
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/TH-MainLoopTwoThreads.cpp ................................................................. 36
Chapter 4

File Documentation

4.1 mainpagetests/01_mainpage.dox File Reference

4.2 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-
core.verification/src/CMAtributes.cpp File Reference

Functions

- void CMAtributes_AttributeGetSynchronous ()
- void CMAtributes_AttributeGetAsynchronous ()
- void CMAtributes_AttributeSetSynchronous ()
- void CMAtributes_AttributeSetAsynchronous ()
- void CMAtributes_AttributeSubscription ()
- int main (int argc, char **argv)

Variables

- const std::string servicedId = "service-sample"
- const std::string clientId = "client-sample"
- const std::string domain = "local"
- const std::string testAddress = "commonapi.communication.TestInterface"
- const int tasync = 100000

4.2.1 Function Documentation

4.2.1.1 void CMAtributes_AttributeGetSynchronous ( )

**Test** Test synchronous getValue API function for attributes with combinations of additional properties readonly and
noSubscriptions (testAttribute, testA readonly, testB noSubscriptions, testC readonly noSubscriptions).

- Set attribute to certain value on stub side.
- Call getValue.
- Check if returned call status is CommonAPI::CallStatus::SUCCESS.
- Check if value of is equal to expected value.
4.2.1.2 void CMAttributes_AttributeGetAsynchronous ( )

Test Test asynchronous getValue API function for attributes with combinations of additional properties readonly and noSubscriptions (testAttribute, testA readonly, testB noSubscriptions, testC readonly noSubscriptions).
- Set attribute to certain value on stub side.
- Call getValue.
- Check if returned call status is CommonAPI::CallStatus::SUCCESS.
- Check if value of is equal to expected value.

4.2.1.3 void CMAttributes_AttributeSetSynchronous ( )

Test Test synchronous setValue API function for attributes with combinations of additional properties readonly and noSubscriptions (testAttribute, testB noSubscriptions)
- Set attribute to certain value on proxy side.
- Check if returned call status is CommonAPI::CallStatus::SUCCESS.
- Check if returned value of setValue is equal to expected value.

4.2.1.4 void CMAttributes_AttributeSetAsynchronous ( )

Test Test asynchronous setValue API function for attributes with combinations of additional properties readonly and noSubscriptions (testAttribute, testB noSubscriptions).
- Set attribute to certain value on proxy side.
- Check if returned call status is CommonAPI::CallStatus::SUCCESS.
- Check if returned value of setValue is equal to expected value.

4.2.1.5 void CMAttributes_AttributeSubscription ( )

Test Test subscription API function for attributes
- Subscribe on testAttribute.
- Set attribute to certain value on stub side.
- Do checks of call status (CommonAPI::CallStatus::SUCCESS) and returned value in callback function.
- Checks if returned value of setValue is equal to expected value.
- Set attribute to certain value with synchronous call from proxy.
- Check again.

4.2.1.6 int main ( int argc, char ∗∗ argv )

4.2.2 Variable Documentation

4.2.2.1 const std::string serviceId = "service-sample"

4.2.2.2 const std::string clientId = "client-sample"
4.2.2.3 const std::string domain = "local"

4.2.2.4 const std::string testAddress = "commonapi.communication.TestInterface"

4.2.2.5 const int tasync = 100000

4.3 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/CMAtributeSubscription.cpp File Reference

Typedefs

• typedef std::shared_ptr<br v1_0::commonapi::communication::TestInterfaceProxy<> > ProxyPtr

Functions

• void testSubscription (ProxyPtr pp)
• void CMAtributeSubscription_SubscriptionStandard ()
• void CMAtributeSubscription_SubscriptionOnAvailable ()
• void CMAtributeSubscription_SubscriptionMultithreading ()
• void CMAtributeSubscription_SubscriptionUnsubscribeFromCallback ()
• void CMAtributeSubscription_SubscribeAndUnsubscribeTwoCallbacksCoexistent ()
• void CMAtributeSubscription_SubscribeAndUnsubscribeImmediatelyUnsubscribing ()
• void CMAtributeSubscription_SubscribeAndUnsubscribeSequentially ()
• void CMAtributeSubscription_SubscribeAndUnsubscribeUnsubscribe ()
• void CMAtributeSubscription_SubscribeAndUnsubscribeServiceNotAvailable ()
• void CMAtributeSubscription_SubscribeUnregisterSetValueRegisterService ()
• void CMAtributeSubscription_SubscribeUnregisterNoValueSetRegisterService ()
• void CMAtributeSubscription_SubscribeSecondProxyLater ()
• void CMAtributeSubscription_SubscribeThreeCallbacksServiceNotAvailable ()
• void CMAtributeSubscription_SubscribeThreeCallbacksServiceAvailable ()
• int main (int argc, char ∗∗argv)

Variables

• const std::string daemonId = "service-sample"
• const std::string clientId = "client-sample"
• const std::string serviceId = "test-service"
• const std::string domain = "local"
• const std::string testAddress = "commonapi.communication.TestInterface"
• const std::string daemonAddress = "commonapi.communication.Daemon"
• const unsigned int wt = 100000
• std::mutex mut
• std::deque<uint32_t> data_queue
• std::condition_variable data_cond
4.3.1 Typedef Documentation

4.3.1.1 typedef std::shared_ptr<v1_0::commonapi::communication::TestInterfaceProxy> ProxyPtr

4.3.2 Function Documentation

4.3.2.1 void testSubscription ( ProxyPtr pp )

4.3.2.2 void CMAttributeSubscription_SubscriptionStandard ( )

**Test** Subscription standard test.

- Register service and check if proxy is available.
- Proxy subscribes for TestAttribute (uint8_t).
  - Change attribute in service several times by set method.
  - Callback function in proxy writes the received values in a queue.
  - Check if values in the queue are the same as the values that were set in the service.
  - Unregister test service.

4.3.2.3 void CMAttributeSubscription_SubscriptionOnAvailable ( )

**Test** Subscription test with subscription on available-event.

- Subscribe for available-event.
- Available-callback subscribes for TestPredefinedTypeAttribute if service is available for proxy and unsubscribes if service is not available for proxy.
  - Change attribute in service by set method; the new attribute value should be received by the proxy because the service is not registered.
  - Register service and change value again; the value should now be received.
  - Unregister and change value again.

4.3.2.4 void CMAttributeSubscription_SubscriptionMultithreading ( )

**Test** Subscription test with several threads.

- Start several threads.
- The threads subscribe for the availability status.
- The available-callback subscribes for TestAttribute if service is available for proxy and unsubscribes if service is not available for proxy.
  - Change attribute in service by set method; the new attribute value should be received by all the threads.
  - The new value is written into a queue.
  - Check if the values of each thread are written into the queue.

4.3.2.5 void CMAttributeSubscription_SubscriptionUnsubscribeFromCallback ( )

**Test** Subscription test : unsubscribe from the subscription callback.

- Register service and check if proxy is available.
- Proxy subscribes for TestAttribute (uint8_t).
4.3.2.6 void CMAttributeSubscription_SubscribeAndUnsubscribeTwoCallbacksCoexistent ( )

Test Test of subscribe and unsubscribe with two coexistent callbacks

- subscribe both callbacks
- change value
- check that both callbacks were executed by changing the value
- unsubscribe both callbacks
- change value
- check that both callbacks were not executed by changing the value

4.3.2.7 void CMAttributeSubscription_SubscribeAndUnsubscribeImmediatelyUnsubscribing ( )

Test Test of subscribing and immediately unsubscribing a callback

- subscribe first callback
- subscribe second callback
- unsubscribe second callback
- change value
- check that only first callback was executed

4.3.2.8 void CMAttributeSubscription_SubscribeAndUnsubscribeSequentially ( )

Test Test of subscribing and immediately sequentially

- subscribe first callback
- subscribe second callback
- change value
- check that both callbacks were executed by changing the value
- unsubscribe first callback
- change value
- check that only second callback was executed
- unsubscribe second callback
- change value
- check that both callbacks were not executed by changing the value
4.3.2.9  void CMAAttributeSubscription_SubscribeAndUnsubscribeUnsubscribe (  )

**Test**  Test of behaviour in case unsubscribe is called two times

- set default value
- register service
- subscribe for the attribute
- current value must be communicated to the proxy
- value of attribute is changed
- changed value must be communicated to the proxy
- proxy unsubscribes for the attribute
- value of attribute is changed
- changed value must not be communicated to the proxy
- proxy unsubscribes again for the attribute
- value of attribute is changed
- changed value must not be communicated to the proxy
- unregister service

4.3.2.10  void CMAAttributeSubscription_SubscribeServiceNotAvailable (  )

**Test**  Test of subscribing in case that service is not available

- set default value
- subscribe for the attribute
- no value is communicated to the proxy
- register service
- current value must be communicated to the proxy
- value of attribute is changed
- changed value must be communicated to the proxy
- unregister service

4.3.2.11  void CMAAttributeSubscription_SubscribeUnregisterSetValueRegisterService (  )

**Test**  Test of unregister a service in case a proxy is subscribed for an attribute of this service. During the unregistered time of the service the value of the attribute is changed.

- register service
- proxy subscribes for an attribute of the service
- value of attribute is set
- changed value must be communicated to the proxy
- unregister service
- value of attribute is changed
- changed value must not be communicated to the proxy
- register service
- current attribute value must be communicated to the proxy
- value of attribute is changed
- changed value must be communicated to the proxy
- unregister service
4.3.2.12 void CMAttributeSubscription_SubscribeUnregisterNoValueSetRegisterService (
Test Test of unregister a service in case a proxy is subscribed for an attribute of this service. During the unregistered time of the service the value of the attribute is not changed.
• register service
• proxy subscribes for an attribute of the service
• value of attribute is set
• changed value must be communicated to the proxy
• unregister service
• register service
• current attribute value must be communicated to the proxy
• value of attribute is changed
• changed value must be communicated to the proxy
• unregister service

4.3.2.13 void CMAttributeSubscription_SubscribeSecondProxyLater (
Test Test of subscribing a second proxy a little bit later
• proxy subscribes for an attribute of the service
• register service
• initial value must be communicated to the proxy
• create a second proxy
• second proxy subscribes for the same attribute of the service
• current attribute value must be communicated to the proxy
• value of attribute is changed
• changed value must be communicated to both proxies
• unregister service

4.3.2.14 void CMAttributeSubscription_SubscribeThreeCallbacksServiceNotAvailable (
Test Test of subscribing three callbacks before registering the service
• proxy subscribes three callbacks for an attribute of the service
• register service
• initial value must be communicated to every callback

4.3.2.15 void CMAttributeSubscription_SubscribeThreeCallbacksServiceAvailable (
Test Test of subscribing three callbacks after registering the service
• register service
• proxy subscribes three callbacks for an attribute of the service
• initial value must be communicated to every callback
4.3.2.16 int main ( int argc, char ** argv )

4.3.3 Variable Documentation

4.3.3.1 const std::string daemonId = "service-sample"
4.3.3.2 const std::string clientId = "client-sample"
4.3.3.3 const std::string serviceld = "test-service"
4.3.3.4 const std::string domain = "local"
4.3.3.5 const std::string testAddress = "commonapi.communication.TestInterface"
4.3.3.6 const std::string daemonAddress = "commonapi.communication.Daemon"
4.3.3.7 const unsigned int wt = 100000
4.3.3.8 std::mutex mut
4.3.3.9 std::deque< uint32_t > data_queue
4.3.3.10 std::condition_variable data_cond

4.4 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-
core.verification/src/CMMethodCalls.cpp File Reference

Functions

• void CMMethodCalls_SynchronousMethodCall ()
• void CMMethodCalls_AsynchronousMethodCall ()
• int main (int argc, char **argv)

Variables

• const std::string serviceId = "service-sample"
• const std::string clientId = "client-sample"
• const std::string domain = "local"
• const std::string testAddress = "commonapi.communication.TestInterface"
• const int tasync = 100000

4.4.1 Function Documentation

4.4.1.1 void CMMethodCalls_SynchronousMethodCall ( )

Test Call test method synchronous and check call status.

• Test stub sets in-value of test method equal out-value of test method.
• Make synchronous call of test method.
• Check if returned call status is CommonAPI::CallStatus::SUCCESS.
• Check if out value of test method is equal to in value.
4.4.1.2 void CMMethodCalls_AsynchronousMethodCall ( )

Test Call test method asynchronous and check call status.

- Test stub sets in-value of test method equal out-value of test method.
- Make asynchronous call of test method.
- Do checks of call status (CommonAPI::CallStatus::SUCCESS) and returned value in callback function.

4.4.2 Variable Documentation

4.4.2.1 const std::string serviceId = "service-sample"
4.4.2.2 const std::string clientId = "client-sample"
4.4.2.3 const std::string domain = "local"
4.4.2.4 const std::string testAddress = "commonapi.communication.TestInterface"
4.4.2.5 const int tasync = 100000

4.5 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/DTAdvanced.cpp File Reference

Functions

- void DTAdvanced_SendAndReceive ( )
- void DTAdvanced_AttributeSet ( )
- void DTAdvanced_BroadcastReceive ( )
- int main (int argc, char ** argv)

Variables

- const std::string domain = "local"
- const std::string testAddress = "commonapi.datatypes.advanced.TestInterface"
- const std::string connectionIdService = "service-sample"
- const std::string connectionIdClient = "client-sample"

4.5.1 Function Documentation

4.5.1.1 void DTAdvanced_SendAndReceive ( )
4.5.1.2 void DTAdvanced_AttributeSet ( )

Test Test attribute functions with advanced types

- Call set function of attributes with advanced types
- Call get function and check if the return value is the same
4.5.1.3 void DTAdvanced_BroadcastReceive ( )

**Test**  Test broadcast with advanced types

- Subscribe to broadcast which contains advanced types
- Call function to cause the stub to fire broadcast event with the same content
- Check if the values in the callback function are as expected

4.5.1.4 int main ( int argc, char * argv )

4.5.2 Variable Documentation

4.5.2.1 const std::string domain = "local"

4.5.2.2 const std::string testAddress = "commonapi.datatypes.advanced.TestInterface"

4.5.2.3 const std::string connectionIdService = "service-sample"

4.5.2.4 const std::string connectionIdClient = "client-sample"

4.6 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-
core.verification/src/DTCombined.cpp File Reference

**Functions**

- void DTCombined_SendAndReceive ()
- int main (int argc, char *argv)

**Variables**

- const std::string domain = "local"
- const std::string testAddress = "commonapi.datatypes.combined.TestInterface"
- const std::string connectionIdService = "service-sample"
- const std::string connectionIdClient = "client-sample"

4.6.1 Function Documentation

4.6.1.1 void DTCombined_SendAndReceive ( )

**Test**  Test function call with combined type

- The combined type is one structure with combinations of advanced and primitive types
- Function call of a function that has for each advanced type one argument (test values) and one return value
- The stub copies the test values to the return values
- On client side the test values are compared with the return values
4.7.1 Function Documentation

4.7.1.1 void DTDerived_SendAndReceive ( )

4.7.1.2 void DTDerived_AttributeSet ( )

4.7.1.3 void DTDerived_BroadcastReceive ( )

Test Test attribute functions with derived types

• Call set function of attributes with derived types
• Call get function and check if the return value is the same

Test Test broadcast with derived types

• Subscribe to broadcast which contains derived types
• Call function to cause the stub to fire broadcast event with the same content
• Check if the values in the callback function are as expected
4.7.1.4 int main ( int argc, char ** argv )

4.7.2 Variable Documentation

4.7.2.1 const std::string domain = "local"

4.7.2.2 const std::string testAddress = "commonapi.datatypes.derived.TestInterface"

4.7.2.3 const std::string connectionId_client = "client-sample"

4.7.2.4 const std::string connectionId_service = "service-sample"

4.8 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-core.verification/src/DTPrimitive.cpp File Reference

Functions

- void DTPrimitive_SendAndReceive ()
- void DTPrimitive_AttributeSet ()
- void DTPrimitive_BroadcastReceive ()
- int main (int argc, char **argv)

Variables

- const std::string domain = "local"
- const std::string testAddress = "commonapi.datatypes.primitive.TestInterface"
- const std::string connectionIdService = "service-sample"
- const std::string connectionIdClient = "client-sample"

4.8.1 Function Documentation

4.8.1.1 void DTPrimitive_SendAndReceive ( )

**Test** Test function call with primitive types

- Primitive types are: uint8_t, int8_t, uint16_t, int16_t, uint32_t, int32_t, uint64_t, int64_t, bool, float, double, std::string, ByteBuffer
- Function call of a function that has for each primitive type one argument (test values) and one return value
- The stub copies the test values to the return values
- On client side the test values are compared with the return values

4.8.1.2 void DTPrimitive_AttributeSet ( )

**Test** Test attribute functions with primitive types

- Call set function of attributes with primitive types
- Call get function and check if the return value is the same
4.8.1.3 void DTPrimitive_BroadcastReceive ()

**Test** Test broadcast with primitive types

- Subscribe to broadcast which contains primitive types
- Call function to cause the stub to fire broadcast event with the same content
- Check if the values in the callback function are as expected

### 4.8.2 Variable Documentation

#### 4.8.2.1 const std::string domain = "local"

#### 4.8.2.2 const std::string testAddress = "commonapi.datatypes.primitive.TestInterface"

#### 4.8.2.3 const std::string connectionIdService = "service-sample"

#### 4.8.2.4 const std::string connectionIdClient = "client-sample"

#### 4.8.4 int main ( int argc, char ** argv )

### 4.9 Functions

- void PFComplex_Ping_Pong_Complex_Synchronous ()
- void PFComplex_Ping_Pong_Complex_Asynchronous ()
- int main (int argc, char ** argv)

#### Variables

- const int usecPerSecond = 1000000
- const std::string serviceId = "service-sample"
- const std::string clientId = "client-sample"
- const std::string domain = "local"
- const std::string testAddress = "commonapi.performance.complex.TestInterface"
- const int tasync = 100000
- const int maxArraySize = 4096 / 16
- const int loopCountPerPayload = 1000

### 4.9.1 Function Documentation

#### 4.9.1.1 void PFComplex_Ping_Pong_Complex_Synchronous ()

**Test** Test synchronous ping pong function call

- complex array is array of a struct containing an union and another struct with primitive datatypes
  - The stub just set the in array to the out array
  - CallStatus and array content will be used to verify the sync call has succeeded
  - Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
  - Doing primitiveLoopSize loops to build the mean time
4.9.1.2 void PFComplex_Ping_Pong_Complex_Asynchronous ( )

Test  Test asynchronous ping pong function call

- complex array is array of a struct containing an union and another struct with primitive datatypes
  - The stub just set (copies) the in array to the out array
  - Only the CallStatus will be used to verify the async call has succeeded
  - Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
  - Doing loopCountPerPayload loops to calc the mean time

4.9.1.3 int main ( int argc, char * * argv )

4.9.2 Variable Documentation

4.9.2.1 const int usecPerSecond = 1000000
4.9.2.2 const std::string serviceId = "service-sample"
4.9.2.3 const std::string clientId = "client-sample"
4.9.2.4 const std::string domain = "local"
4.9.2.5 const std::string testAddress = "commonapi.performance.complex.TestInterface"
4.9.2.6 const int tasync = 100000
4.9.2.7 const int maxArraySize = 4096 / 16
4.9.2.8 const int loopCountPerPayload = 1000

4.10 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-
core.verification/src/PFPrimitive.cpp File Reference

Functions

- void PFPrimitive_Ping_Pong_Primitive_Synchronous ( )
- void PFPrimitive_Ping_Pong_Primitive_Asynchronous ( )
- int main ( int argc, char * * argv)

Variables

- const std::string serviceld = "service-sample"
- const std::string clientId = "client-sample"
- const std::string domain = "local"
- const std::string testAddress = "commonapi.performance.primitive.TestInterface"
- const int tasync = 100000
- const int usecPerSecond = 1000000
- const int maxPrimitiveArraySize = 1024*16
- const int loopCountPerPayload = 1000

Generated on Wed Jul 29 2015 01:00:38 for CommonAPITests by Doxygen
4.10.1 Function Documentation

4.10.1.1 void PFPrimitive_Ping_Pong_Primitive_Synchronous ( )

Test  Test synchronous ping pong function call

• primitive array is array of UInt_8
  – The stub just set the in array to the out array
  – CallStatus and array content will be used to verify the sync call has succeeded
  – Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
  – Doing primitiveLoopSize loops to build the mean time

4.10.1.2 void PFPrimitive_Ping_Pong_Primitive_Asynchronous ( )

Test  Test asynchronous ping pong function call

• primitive array is array of UInt_8
  – The stub just set (copies) the in array to the out array
  – Only the CallStatus will be used to verify the async call has succeeded
  – Using double payload every cycle, starting with 1 end with maxPrimitiveArraySize
  – Doing primitiveLoopSize loops to build the mean time

4.10.1.3 int main ( int argc, char ∗∗ argv )

4.10.2 Variable Documentation

4.10.2.1 const std::string serviceId = "service-sample"
4.10.2.2 const std::string clientId = "client-sample"
4.10.2.3 const std::string domain = "local"
4.10.2.4 const std::string testAddress = "commonapi.performance.primitive.TestInterface"
4.10.2.5 const int tasync = 100000
4.10.2.6 const int usecPerSecond = 1000000
4.10.2.7 const int maxPrimitiveArraySize = 1024 * 16
4.10.2.8 const int loopCountPerPayload = 1000

4.11  /home/jg/work/sources/capi/asccgit017.CommonAPI-Tools/org.genivi.commonapi.core.-
core.verification/src/RTBuildProxiesAndStubs.cpp File Reference

Functions

• void RTBuildProxiesAndStubs_LoadedRuntimeCanBuildProxiesAndStubs ()
• void RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes ()
• int main (int argc, char ∗∗argv)
Variables

- const std::string domain = "local"
- const std::string testAddress = "commonapi.runtime.TestInterface"
- const std::string applicationNameService = "service-sample"
- const std::string applicationNameClient = "client-sample"

4.11.1 Function Documentation

4.11.1.1 void RTBuildProxiesAndStubs_LoadedRuntimeCanBuildProxiesAndStubs ( )

Test Loads Runtime, creates proxy and stub/service.
- Calls CommonAPI::Runtime::get() and checks if return value is true.
- Checks if test proxy with domain and test instance can be created.
- Checks if test stub can be created.
- Register the test service.
- Unregister the test service.

4.11.1.2 void RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes ( )

Test Loads Runtime, creates proxy and stub/service two times.
- Calls CommonAPI::Runtime::get() and checks if return value is true
- Create stub and register service
- Create proxy
- Do some synchronous calls
- Unregister the service.
- Create stub and register service
- Create proxy
- Checks whether proxy is available
- Unregister the service

4.11.3 int main ( int argc, char ** argv )

4.11.2 Variable Documentation

4.11.2.1 const std::string domain = "local"

4.11.2.2 const std::string testAddress = "commonapi.runtime.TestInterface"

4.11.2.3 const std::string applicationNameService = "service-sample"

4.11.2.4 const std::string applicationNameClient = "client-sample"

4.12 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-
core.verification/src/RTLoadingRuntime.cpp File Reference

Functions

- void RTLoadingRuntime_LoadsDefaultRuntime ()
- int main (int argc, char **argv)
4.12.1 Function Documentation

4.12.1.1 void RTLoadingRuntime_LoadsDefaultRuntime ( )

**Test** Loads Default Runtime.

- Calls CommonAPI::Runtime::get().
- Success if return value is true.

4.12.1.2 int main ( int argc, char ** argv )

4.13 Functions

- void StabilitySP_RepeatedRegistrations ( )
- void StabilitySP_MultipleMethodCalls ( )
- void StabilitySP_MultipleAttributeSets ( )
- void StabilitySP_MultipleAttributeGets ( )
- void StabilitySP_MultipleAttributeGetAsyncs ( )
- void StabilitySP_MultipleAttributeSetAsyncs ( )
- void StabilitySP_MultipleAttributeSubscriptions ( )
- int main (int argc, char **argv)

4.13 Variables

- const std::string serviceld = "service-sample"
- const std::string clientld = "client-sample"
- const std::string domain = "local"
- const std::string testAddress = "commonapi.stability.sp.TestInterface"
- const std::string COMMONAPI_CONFIG_SUFFIX = ".conf"
- const int MAXSERVERCOUNT = 40
- const int MAXTHREADCOUNT = 8
- const int MAXMETHODCALLS = 80
- const int MAXREGLOOPS = 16
- const int MAXREGCOUNT = 16
- const int MESSAGESIZE = 80
- const int MAXSUBSCRIPTIONSETS = 10

4.13.1 Function Documentation

4.13.1.1 void StabilitySP_RepeatedRegistrations ( )

**Test** Register and unregister services in a loop.

- do MAXREGLOOPS times:
  - register MAXREGCOUNT addresses as services
  - unregister the addresses that were just registered
  - check the return code of each register/unregister call
  - test fails if any of the return codes are false
4.13.1.2 void StabilitySP_MultipleMethodCalls()

**Test** Create a number of services and proxies and send messages through them.

- Register MAXSERVERCOUNT addresses as services
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sends MAXMETHODCALLS messages to each.
- Each message is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the return message from the server is not correct

4.13.1.3 void StabilitySP_MultipleAttributeSets()

**Test** Create a number of services and proxies and set attributes through them.

- Register MAXSERVERCOUNT addresses as services
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times to each.
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the return attribute from the server is not correct

4.13.1.4 void StabilitySP_MultipleAttributeGets()

**Test** Create a number of services and proxies and get attributes through them.

- Register MAXSERVERCOUNT addresses as services
  - Set the attribute for service, at the stub side.
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then gets attributes MAXMETHODCALLS times for each.
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the returned attribute from the server is not correct

4.13.1.5 void StabilitySP_MultipleAttributeGetAsyncs()

**Test** Create a number of services and proxies and get attributes through them.

- Register MAXSERVERCOUNT addresses as services
  - Set the attribute for service, at the stub side.
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then gets attributes MAXMETHODCALLS times for each asynchronously
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won’t get available or if the callbacks are not called correct number of times
4.13.6 void StabilitySP_MultipleAttributeSetAsyncs ( )

**Test** Create a number of services and proxies and set attributes through them.

- Register MAXSERVERCOUNT addresses as services
  - Set the attribute for service, at the stub side.
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times for each asynchronously
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won't get available or if the callbacks are not called correct number of times

4.13.7 void StabilitySP_MultipleAttributeSubscriptions ( )

**Test** Create a number of services and proxies and set attributes through them.

- Register MAXSERVERCOUNT addresses as services
  - Set the attribute for service, at the stub side.
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times for each asynchronously
- Each attribute is MESSAGESIZE bytes long.
- Test fails if any of the services fail to get registered or if any of the proxies won't get available or if the callbacks are not called correct number of times

4.13.8 int main ( int argc, char ** argv )

4.13.2 Variable Documentation

4.13.2.1 const std::string serviceId = "service-sample"

4.13.2.2 const std::string clientId = "client-sample"

4.13.2.3 const std::string domain = "local"

4.13.2.4 const std::string testAddress = "commonapi.stability.sp.TestInterface"

4.13.2.5 const std::string COMMONAPI_CONFIG_SUFFIX = ".conf"

4.13.2.6 const int MAXSERVERCOUNT = 40

4.13.2.7 const int MAXTHREADCOUNT = 8

4.13.2.8 const int MAXMETHODCALLS = 80

4.13.2.9 const int MAXREGLOOPS = 16

4.13.2.10 const int MAXREGCOUNT = 16

4.13.2.11 const int MESSAGESIZE = 80

4.13.2.12 const int MAXSUBSCRIPTIONSETS = 10

Generated on Wed Jul 29 2015 01:00:38 for CommonAPITests by Doxygen
4.14 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-core.verification/src/THMainLoopIndependence.cpp File Reference

Functions

- void THMainLoopIndependence_ProxyReceivesAnswerOnlyIfStubMainLoopRuns()
- void THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers()
- int main(int argc, char **argv)

Variables

- const std::string domain = "local"
- const std::string instance6 = "my.test.commonapi.address.six"
- const std::string instance7 = "my.test.commonapi.address.seven"
- const std::string instance8 = "my.test.commonapi.address.eight"
- const std::string mainloopName1 = "client-sample"
- const std::string mainloopName2 = "service-sample"
- const std::string thirdPartyServiceId = "mainloop-thirdParty"

4.14.1 Function Documentation

4.14.1.1 void THMainLoopIndependence_ProxyReceivesAnswerOnlyIfStubMainLoopRuns()

Test Proxy Receives Answer Only If Stub MainLoop Runs.

- start proxy in thread 1 and call testPredefinedTypeMethod
- proxy should not receive answer, if the stub mainloop does not run
- run mainloop of stub
- now the stub mainloop also runs, so the proxy should receive the answer

4.14.1.2 void THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers()

Test Proxy Receives Just His Own Answers.

- start 2 proxies in own threads
- call test method in each proxy
- now each proxy should have received the answer to his own request

4.14.1.3 int main(int argc, char **argv)

4.14.2 Variable Documentation

4.14.2.1 const std::string domain = "local"

4.14.2.2 const std::string instance6 = "my.test.commonapi.address.six"

4.14.2.3 const std::string instance7 = "my.test.commonapi.address.seven"

4.14.2.4 const std::string instance8 = "my.test.commonapi.address.eight"
const std::string mainloopName1 = "client-sample"

const std::string mainloopName2 = "service-sample"

const std::string thirdPartyServiceId = "mainloop-thirdParty"

void THMainLoopIntegration_VerifyCommunicationWithMainLoop()
void THMainLoopIntegration_VerifyTransportReading()
void THMainLoopIntegration_VerifySyncCallMessageHandlingOrder()
int main (int argc, char **argv)

const std::string domain = "local"
const std::string instance = "my.test.commonapi.address"
const std::string connection_client = "client-sample"
const std::string connection_service = "service-sample"

Test Verifies Transport Reading When Dispatching Watches.

  • get proxy with available flag = true
  • generate big test data
  • send asynchronous test message
  • dispatch dispatchSource: the message must not be arrived
  • dispatch watches (reads transport).
  • dispatch dispatchSources again: now the message must be arrived.

Test Verifies Transport Reading When Dispatching Watches.

  • get proxy with available flag = true
  • generate big test data
  • send asynchronous test message
  • dispatch dispatchSource: the message must not be arrived
  • dispatch watches (reads transport).
  • dispatch dispatchSources again: now the message must be arrived.
4.15.1.3 void THMainLoopIntegration_VerifySyncCallMessageHandlingOrder ( )

Test Verifies Synchronous Call Message Handling Order.

- get proxy with available flag = true
- subscribe for broadcast event
- generate 5 test broadcasts
- 5 broadcasts should arrive in the right order

4.15.1.4 int main ( int argc, char ** argv )

4.15.2 Variable Documentation

4.15.2.1 const std::string domain = "local"

4.15.2.2 const std::string instance = "my.test.commonapi.address"

4.15.2.3 const std::string connection_client = "client-sample"

4.15.2.4 const std::string connection_service = "service-sample"

4.16 /home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.-
core.verification/src/THMainLoopTwoThreads.cpp File Reference

Functions

- void THMainLoopTwoThreads_ProxyGetsAvailableStatus ()
- void THMainLoopTwoThreads_ProxyGetsFunctionResponse ()
- int main (int argc, char **argv)

Variables

- const std::string domain = "local"
- const std::string instance = "my.test.commonapi.address"

4.16.1 Function Documentation

4.16.1.1 void THMainLoopTwoThreads_ProxyGetsAvailableStatus ( )

Test Proxy Receives Available when MainLoop Dispatched sourced out to other thread.

4.16.1.2 void THMainLoopTwoThreads_ProxyGetsFunctionResponse ( )

Test Proxy gets function response when MainLoop Dispatched sourced out to other thread.
4.16.1.3 int main ( int argc, char ∗∗ argv )

4.16.2 Variable Documentation

4.16.2.1 const std::string domain = "local"

4.16.2.2 const std::string instance = "my.test.commonapi.address"
DTDerived_AttributeSet, 25
DTDerived_BroadcastReceive, 25
DTDerived_SendAndReceive, 25
domain, 26
main, 25
testAddress, 26
DTDerived_AttributeSet
DTDerived.cpp, 25
DTDerived_BroadcastReceive
DTDerived.cpp, 25
DTDerived_SendAndReceive
DTDerived.cpp, 25
DTPrimitive.cpp
collectionIdClient, 27
collectionIdService, 27
DTPrimitive_AttributeSet, 26
DTPrimitive_BroadcastReceive, 26
DTPrimitive_SendAndReceive, 26
domain, 27
main, 27
testAddress, 27
DTPrimitive_AttributeSet
DTPrimitive.cpp, 26
DTPrimitive_BroadcastReceive
DTPrimitive.cpp, 26
DTPrimitive_SendAndReceive
DTPrimitive.cpp, 26
daemonAddress
CMAttributeSubscription.cpp, 22
daemonId
CMAttributeSubscription.cpp, 22
data_cond
CMAttributeSubscription.cpp, 22
data_queue
CMAttributeSubscription.cpp, 22
domain
CMAttributes.cpp, 16
CMAttributeSubscription.cpp, 22
CMMethodCalls.cpp, 23
DAdvanced.cpp, 24
DCCombined.cpp, 24
DTDerived.cpp, 26
DTPrimitive.cpp, 27
PFComplex.cpp, 28
PPPrimitive.cpp, 29
RTBuildProxiesAndStubs.cpp, 30
StabilitySP.cpp, 33
THMainLoopIndependence.cpp, 34
THMainLoopIntegration.cpp, 36
THMainLoopTwoThreads.cpp, 36
mainloopName1
THMainLoopIndependence.cpp, 34
mainloopName2
THMainLoopIndependence.cpp, 35
mainpagetests/01_mainpage.dox, 15
maxArraySize
PFComplex.cpp, 28
maxPrimitiveArraySize
PPPrimitive.cpp, 29
mut
CMAttributeSubscription.cpp, 22
PFComplex.cpp
clientld, 28
domain, 28
loopCountPerPaylod, 28
main, 28
maxArraySize, 28
PFComplex_Ping_Pong_Complex_Asynchronous, 27
PFComplex_Ping_Pong_Complex_Synchronous, 27
serviceld, 28
THMainLoopIntegration_VerifyTransportReading, 35
THMainLoopIntegration_VerifyCommunicationWithMainLoop
THMainLoopIntegration.cpp, 35
THMainLoopIntegration_VerifySyncCallMessageHandlingOrder
THMainLoopIntegration.cpp, 35
THMainLoopIntegration_VerifyTransportReading
THMainLoopIntegration.cpp, 35
THMainLoopTwoThreads.cpp
domain, 37
instance, 37
main, 36
THMainLoopTwoThreads_ProxyGetsAvailableStatus, 36
THMainLoopTwoThreads_ProxyGetsFunctionResponse, 36
THMainLoopTwoThreads_ProxyGetsAvailableStatus
THMainLoopTwoThreads.cpp, 36
THMainLoopTwoThreads_ProxyGetsFunctionResponse
THMainLoopTwoThreads.cpp, 36
tasync
CMAttributes.cpp, 17
CMMethodCalls.cpp, 23
PFComplex.cpp, 28
PFPrimitive.cpp, 29
testAddress
CMAttributes.cpp, 17
CMAttributeSubscription.cpp, 22
CMMethodCalls.cpp, 23
DTAdvanced.cpp, 24
DTCOMbined.cpp, 25
DTDerived.cpp, 26
DTPrimitive.cpp, 27
PFComplex.cpp, 28
PFPrimitive.cpp, 29
RTBuildProxiesAndStubs.cpp, 30
StabilitySP.cpp, 33
testSubscription
CMAttributeSubscription.cpp, 18
thirdPartyServiceId
THMainLoopIndependence.cpp, 35
usecPerSecond
PFComplex.cpp, 28
PFPrimitive.cpp, 29
wt
CMAttributes.cpp, 22