

CommonAPITests

Generated by Doxygen 1.8.6

Tue Jan 19 2016 02:00:16

Contents

1	Main Page	1
2	Test List	3
3	File Index	17
3.1	File List	17
4	File Documentation	19
4.1	mainpagetests/01_mainpage.dox File Reference	19
4.2	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- AFManaged.cpp File Reference	19
4.2.1	Function Documentation	19
4.2.1.1	AFManaged_AddRemoveManagedInterfaceSingle	19
4.2.1.2	AFManaged_AddRemoveManagedInterfaceMultiple	20
4.2.1.3	AFManaged_AddRemoveMultipleManagedInterfacesSingle	20
4.2.1.4	AFManaged_AddRemoveMultipleManagedInterfacesMultiple	20
4.2.1.5	AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistration- Explicit	20
4.2.1.6	AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistration- ExplicitAll	20
4.2.1.7	AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistration- Implicit	21
4.2.1.8	AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistration- Explicit	21
4.2.1.9	AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistration- ExplicitAll	21
4.2.1.10	AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistration- Implicit	21
4.2.1.11	AFManaged_ProxyManagerTestPrimitiveMethods	21
4.2.1.12	AFManaged_ProxyManagerTestNonPrimitiveMethodsSync	22
4.2.1.13	AFManaged_ProxyManagerTestNonPrimitiveMethodsAsync	22
4.2.1.14	AFManaged_DISABLED_ProxyManagerTestGetInstanceAvailabilityStatusAsync	22
4.2.1.15	main	23
4.2.2	Variable Documentation	23
4.2.2.1	domain	23

4.3	/home/jg/work/sources/capi/asccgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- AFPolymorph.cpp File Reference	23
4.3.1	Function Documentation	23
4.3.1.1	AFPolymorph_SetAndGetAttributeTypedef	23
4.3.1.2	AFPolymorph_SetAndGetAttributeEnum	24
4.3.1.3	AFPolymorph_SetAndGetAttributeUInt	24
4.3.1.4	AFPolymorph_SetAndGetAttributeString	24
4.3.1.5	AFPolymorph_SetAndGetAttributeStruct	24
4.3.1.6	AFPolymorph_MethodCall	24
4.3.1.7	AFPolymorph_Broadcast	24
4.3.1.8	main	24
4.3.2	Variable Documentation	24
4.3.2.1	domain	24
4.3.2.2	testAddress	24
4.3.2.3	connectionId_client	24
4.3.2.4	connectionId_service	24
4.4	/home/jg/work/sources/capi/asccgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- AFSelective.cpp File Reference	24
4.4.1	Function Documentation	25
4.4.1.1	AFSelective_SelectiveBroadcastRejected	25
4.4.1.2	AFSelective_SelectiveBroadcast	25
4.4.1.3	AFSelective_SelectiveMultiBroadcast	25
4.4.1.4	AFSelective_DISABLED_SelectiveRejectedMultiBroadcast	26
4.4.1.5	main	26
4.4.2	Variable Documentation	26
4.4.2.1	serviceId	26
4.4.2.2	clientId	26
4.4.2.3	otherclientId	26
4.4.2.4	domain	26
4.4.2.5	testAddress	26
4.4.2.6	tasync	26
4.5	/home/jg/work/sources/capi/asccgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- CMAAttributes.cpp File Reference	26
4.5.1	Function Documentation	26
4.5.1.1	CMAAttributes_AttributeGetSynchronous	26
4.5.1.2	CMAAttributes_AttributeGetAsynchronous	27
4.5.1.3	CMAAttributes_AttributeSetSynchronous	27
4.5.1.4	CMAAttributes_AttributeSetAsynchronous	27
4.5.1.5	CMAAttributes_AttributeSubscription	27
4.5.1.6	main	28
4.5.2	Variable Documentation	28

4.5.2.1	serviceld	28
4.5.2.2	clientId	28
4.5.2.3	domain	28
4.5.2.4	testAddress	28
4.5.2.5	tasync	28
4.6	/home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-CMAAttributeSubscription.cpp File Reference	28
4.6.1	Typedef Documentation	29
4.6.1.1	ProxyPtr	29
4.6.2	Function Documentation	29
4.6.2.1	testSubscription	29
4.6.2.2	CMAAttributeSubscription_SubscriptionStandard	29
4.6.2.3	CMAAttributeSubscription_SubscriptionOnAvailable	29
4.6.2.4	CMAAttributeSubscription_SubscriptionMultithreading	29
4.6.2.5	CMAAttributeSubscription_SubscriptionUnsubscribeFromCallback	29
4.6.2.6	CMAAttributeSubscription_SubscribeAndUnsubscribeTwoCallbacksCoexistent	30
4.6.2.7	CMAAttributeSubscription_SubscribeAndUnsubscribeSequentially	30
4.6.2.8	CMAAttributeSubscription_SubscribeAndUnsubscribeUnsubscribe	31
4.6.2.9	CMAAttributeSubscription_SubscribeServiceNotAvailable	31
4.6.2.10	CMAAttributeSubscription_SubscribeUnregisterSetValueRegisterService	31
4.6.2.11	CMAAttributeSubscription_SubscribeUnregisterNoValueSetRegisterService	32
4.6.2.12	CMAAttributeSubscription_SubscribeSecondProxyLater	32
4.6.2.13	CMAAttributeSubscription_SubscribeThreeCallbacksServiceNotAvailable	32
4.6.2.14	CMAAttributeSubscription_SubscribeThreeCallbacksServiceAvailable	32
4.6.2.15	main	33
4.6.3	Variable Documentation	33
4.6.3.1	daemonId	33
4.6.3.2	clientId	33
4.6.3.3	serviceld	33
4.6.3.4	domain	33
4.6.3.5	testAddress	33
4.6.3.6	daemonAddress	33
4.6.3.7	wt	33
4.6.3.8	mut	33
4.6.3.9	data_queue	33
4.6.3.10	data_cond	33
4.7	/home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-CMBroadcasts.cpp File Reference	33
4.7.1	Function Documentation	33
4.7.1.1	CMBroadcasts_NormalBroadcast	33

4.7.1.2	CMBroadcasts_SelectiveBroadcastRejected	34
4.7.1.3	CMBroadcasts_SelectiveBroadcast	34
4.7.1.4	CMBroadcasts_BroadcastStubGoesOfflineOnlineAgain	34
4.7.1.5	CMBroadcasts_SelectiveBroadcastStubGoesOfflineOnlineAgain	34
4.7.1.6	main	35
4.7.2	Variable Documentation	35
4.7.2.1	serviceld	35
4.7.2.2	clientId	35
4.7.2.3	domain	35
4.7.2.4	testAddress	35
4.7.2.5	tasync	35
4.8	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-CMMethodCalls.cpp File Reference	35
4.8.1	Function Documentation	35
4.8.1.1	CMMethodCalls_SynchronousMethodCall	35
4.8.1.2	CMMethodCalls_AsynchronousMethodCall	35
4.8.1.3	main	36
4.8.2	Variable Documentation	36
4.8.2.1	serviceld	36
4.8.2.2	clientId	36
4.8.2.3	domain	36
4.8.2.4	testAddress	36
4.8.2.5	tasync	36
4.9	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-DTAdvanced.cpp File Reference	36
4.9.1	Function Documentation	36
4.9.1.1	DTAdvanced_SendAndReceive	36
4.9.1.2	DTAdvanced_SendAndReceiveInvalid	36
4.9.1.3	DTAdvanced_AttributeSetInvalid	36
4.9.1.4	DTAdvanced_AttributeSetAsyncInvalid	36
4.9.1.5	DTAdvanced_AttributeSet	37
4.9.1.6	DTAdvanced_BroadcastReceive	37
4.9.1.7	main	37
4.9.2	Variable Documentation	37
4.9.2.1	domain	37
4.9.2.2	testAddress	37
4.9.2.3	connectionIdService	37
4.9.2.4	connectionIdClient	37
4.10	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-DTCombined.cpp File Reference	37
4.10.1	Function Documentation	38

4.10.1.1	DTCombined_SendAndReceive	38
4.10.1.2	main	38
4.10.2	Variable Documentation	38
4.10.2.1	domain	38
4.10.2.2	testAddress	38
4.10.2.3	connectionIdService	38
4.10.2.4	connectionIdClient	38
4.11	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- DTDerived.cpp File Reference	38
4.11.1	Function Documentation	38
4.11.1.1	DTDerived_SendAndReceive	38
4.11.1.2	DTDerived_AttributeSet	38
4.11.1.3	DTDerived_BroadcastReceive	39
4.11.1.4	main	39
4.11.2	Variable Documentation	39
4.11.2.1	domain	39
4.11.2.2	testAddress	39
4.11.2.3	connectionId_client	39
4.11.2.4	connectionId_service	39
4.12	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- DTPrimitive.cpp File Reference	39
4.12.1	Function Documentation	39
4.12.1.1	DTPrimitive_SendAndReceive	39
4.12.1.2	DTPrimitive_AttributeSet	40
4.12.1.3	DTPrimitive_BroadcastReceive	40
4.12.1.4	DTPrimitive_EmptyBroadcastReceive	40
4.12.1.5	main	40
4.12.2	Variable Documentation	40
4.12.2.1	domain	40
4.12.2.2	testAddress	40
4.12.2.3	connectionIdService	40
4.12.2.4	connectionIdClient	40
4.13	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- PFComplex.cpp File Reference	40
4.13.1	Function Documentation	41
4.13.1.1	PFComplex_Ping_Pong_Complex_Synchronous	41
4.13.1.2	PFComplex_Ping_Pong_Complex_Asynchronous	41
4.13.1.3	main	41
4.13.2	Variable Documentation	41
4.13.2.1	usecPerSecond	41
4.13.2.2	serviceId	41

4.13.2.3	clientId	41
4.13.2.4	domain	41
4.13.2.5	testAddress	41
4.13.2.6	tasync	41
4.13.2.7	maxArraySize	41
4.13.2.8	loopCountPerPayload	41
4.14	/home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- PFPrimitive.cpp File Reference	41
4.14.1	Function Documentation	42
4.14.1.1	PFPrimitive_Ping_Pong_Primitive_Synchronous	42
4.14.1.2	PFPrimitive_Ping_Pong_Primitive_Asynchronous	42
4.14.1.3	main	42
4.14.2	Variable Documentation	42
4.14.2.1	serviceld	42
4.14.2.2	clientId	42
4.14.2.3	domain	42
4.14.2.4	testAddress	42
4.14.2.5	tasync	43
4.14.2.6	usecPerSecond	43
4.14.2.7	maxPrimitiveArraySize	43
4.14.2.8	loopCountPerPayload	43
4.15	/home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- RTBuildProxiesAndStubs.cpp File Reference	43
4.15.1	Function Documentation	43
4.15.1.1	RTBuildProxiesAndStubs_LoadedRuntimeCanBuildProxiesAndStubs	43
4.15.1.2	RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes	43
4.15.1.3	main	44
4.15.2	Variable Documentation	44
4.15.2.1	domain	44
4.15.2.2	testAddress	44
4.15.2.3	applicationNameService	44
4.15.2.4	applicationNameClient	44
4.16	/home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- RTLLoadingRuntime.cpp File Reference	44
4.16.1	Function Documentation	44
4.16.1.1	RTLLoadingRuntime_LoadsDefaultRuntime	44
4.16.1.2	main	44
4.17	/home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/- StabilitySP.cpp File Reference	44
4.17.1	Function Documentation	45
4.17.1.1	StabilitySP_RepeatedRegistrations	45

4.17.1.2	StabilitySP_MultipleMethodCalls	45
4.17.1.3	StabilitySP_MultipleAttributeSets	45
4.17.1.4	StabilitySP_MultipleAttributeGets	45
4.17.1.5	StabilitySP_MultipleAttributeGetAsyncs	46
4.17.1.6	StabilitySP_MultipleAttributeSetAsyncs	46
4.17.1.7	StabilitySP_MultipleAttributeSubscriptions	46
4.17.1.8	main	47
4.17.2	Variable Documentation	47
4.17.2.1	serviceld	47
4.17.2.2	clientId	47
4.17.2.3	domain	47
4.17.2.4	testAddress	47
4.17.2.5	COMMONAPI_CONFIG_SUFFIX	47
4.17.2.6	MAXSERVERCOUNT	47
4.17.2.7	MAXTHREADCOUNT	47
4.17.2.8	MAXMETHODCALLS	47
4.17.2.9	MAXREGLOOPS	47
4.17.2.10	MAXREGCOUNT	47
4.17.2.11	MESSAGESIZE	47
4.17.2.12	MAXSUBSCRIPTIONSETS	47
4.18	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-THMainLoopIndependence.cpp File Reference	47
4.18.1	Function Documentation	47
4.18.1.1	THMainLoopIndependence_ProxyReceivesAnswerOnlyIfStubMainLoopRuns	47
4.18.1.2	THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers	48
4.18.1.3	main	48
4.18.2	Variable Documentation	48
4.18.2.1	domain	48
4.18.2.2	instance6	48
4.18.2.3	instance7	48
4.18.2.4	instance8	48
4.18.2.5	mainloopName1	48
4.18.2.6	mainloopName2	48
4.18.2.7	thirdPartyServiceId	48
4.19	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-THMainLoopIntegration.cpp File Reference	48
4.19.1	Function Documentation	49
4.19.1.1	THMainLoopIntegration_VerifyCommunicationWithMainLoop	49
4.19.1.2	THMainLoopIntegration_VerifyTransportReading	49
4.19.1.3	THMainLoopIntegration_VerifySyncCallMessageHandlingOrder	49

4.19.1.4	main	49
4.19.2	Variable Documentation	49
4.19.2.1	domain	49
4.19.2.2	instance	49
4.19.2.3	connection_client	49
4.19.2.4	connection_service	49
4.20	/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/-THMainLoopTwoThreads.cpp File Reference	49
4.20.1	Function Documentation	50
4.20.1.1	THMainLoopTwoThreads_ProxyGetsAvailableStatus	50
4.20.1.2	THMainLoopTwoThreads_ProxyGetsFunctionResponse	50
4.20.1.3	main	50
4.20.2	Variable Documentation	50
4.20.2.1	domain	50
4.20.2.2	instance	50
Index		51

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 **AFManaged_AddRemoveManagedInterfaceMultiple ()** .

Subscribe on the events about availability status changes at the manager

- Add a managed interface to the manager
- Check that the client is notified about the newly added interface
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

Global **AFManaged_AddRemoveManagedInterfaceSingle ()** .

Subscribe on the events about availability status changes at the manager

- Add a managed interface to the manager
- Check that the client is notified about the newly added interface
- Remove the managed interface from the manager
- Check that the client is notified about the removed interface

Global **AFManaged_AddRemoveMultipleManagedInterfacesMultiple ()** .

Add a managed interface to the manager

- Check that the client is notified about the newly added interface
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

Global **AFManaged_AddRemoveMultipleManagedInterfacesSingle ()** .

Add a managed interface to the manager

- Check that the client is notified about the newly added interface
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

Global **AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationExplicit ()** .

Subscribe on the events about availability status changes at the manager

- Add managed interfaces to the manager
 - Check that the client is notified about the newly added interfaces
 - Build proxies through the manager to the managed interfaces
 - Call a method on the managed interfaces and check call status
 - Explicitly deregister managed interfaces through their instance name
- Global [AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationExplicitAll \(\)](#)** .
- Subscribe on the events about availability status changes at the manager
- Add managed interfaces to the manager
 - Check that the client is notified about the newly added interfaces
 - Build proxies through the manager to the managed interfaces
 - Call a method on the managed interfaces and check call status
 - Deregister all managed interfaces through manager's stub adapter
- Global [AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationImplicit \(\)](#)** .
- Subscribe on the events about availability status changes at the manager
- Add managed interfaces to the manager
 - Check that the client is notified about the newly added interfaces
 - Build proxies through the manager to the managed interfaces
 - Call a method on the managed interfaces and check call status
 - Don't deregister managed interfaces. This is done in dtor of manager's StubAdapterInternal when manager service is unregistered in TearDown() method.
- Global [AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationExplicit \(\)](#)** .
- Subscribe on the events about availability status changes at the manager
- Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Build a proxy through the manager to the managed device
 - Call a method on the managed device and check call status
 - Explicitly deregister managed interface through its instance name
- Global [AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationExplicitAll \(\)](#)** .
- Subscribe on the events about availability status changes at the manager
- Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Build a proxy through the manager to the managed device
 - Call a method on the managed device and check call status
 - Deregister all managed interfaces through manager's stub adapter
- Global [AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationImplicit \(\)](#)** .
- Subscribe on the events about availability status changes at the manager
- Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Build a proxy through the manager to the managed device
 - Call a method on the managed device and check call status
 - Don't deregister managed interfaces. This is done in dtor of manager's StubAdapterInternal when manager service is unregistered in TearDown() method.
- Global [AFManaged_DISABLED_ProxyManagerTestGetInstanceAvailabilityStatusAsync \(\)](#)** .
- Add a managed interface to the manager
- Check that the client is notified about the newly added interface

- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatusAsync method to check that all returned instances by getAvailableInstances are available
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatusAsync method to check that all returned instances by getAvailableInstances are available
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatusAsync method to check that all returned instances by getAvailableInstances are available
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

Global **AFManaged_ProxyManagerTestNonPrimitiveMethodsAsync ()** .

Add a managed interface to the manager

- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstancesAsync method to check that all registered instances are returned
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstancesAsync method to check that all registered instances are returned
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstancesAsync method to check that all registered instances are returned
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

Global **AFManaged_ProxyManagerTestNonPrimitiveMethodsSync ()** .

Add a managed interface to the manager

- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatus method to check that all returned instances by getAvailableInstances are available
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatus method to check that all returned instances by getAvailableInstances are available
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatus method to check that all returned instances by getAvailableInstances are available
- Remove all the managed interfaces from the manager

- Check that the client is notified about the removed interfaces

Global [AFManaged_ProxyManagerTestPrimitiveMethods](#) () .

Test the getConnectionId, getDomain and getInterface methods available via the ProxyManager of the respective managed interfaces of the manager

Global [AFPolymorph_Broadcast](#) () .

Call a method with a special value that tells the stub to send a broadcast signal

- verify that the received data matches the transmitted data

Global [AFPolymorph_MethodCall](#) () .

Call a method whose input and output parameters are polymorphic structures

- verify that the received data matches the transmitted data

Global [AFPolymorph_SetAndGetAttributeEnum](#) () .

Set and get a enum-type attribute through a polymorphic structure

- verify that the received data matches the transmitted data

Global [AFPolymorph_SetAndGetAttributeString](#) () .

Set and get a string-type attribute through a polymorphic structure

- verify that the received data matches the transmitted data

Global [AFPolymorph_SetAndGetAttributeStruct](#) () .

Set and get a struct-type attribute through a polymorphic structure

- verify that the received data matches the transmitted data

Global [AFPolymorph_SetAndGetAttributeTypedef](#) () .

Set and get a typedef-type attribute through a polymorphic structure

- verify that the received data matches the transmitted data

Global [AFPolymorph_SetAndGetAttributeUInt](#) () .

Set and get a uint-type attribute through a polymorphic structure

- verify that the received data matches the transmitted data

Global [AFSelective_DISABLED_SelectiveRejectedMultiBroadcast](#) ()

Test multiple selective broadcasts, with rejection.

- subscribe to stub three times: once from proxy2, once from proxy1 (accepted) once from proxy2 (rejected)
- This should result with two subscription callbacks being called from broadcast.

Global [AFSelective_SelectiveBroadcast](#) ()

Test selective broadcasts.

- inform stub to start accepting subscriptions
- subscribe to the selective broadcast
- check that no error was received (in a reasonable time)
- inform stub to send a broadcast
- check that a correct value is received

Global [AFSelective_SelectiveBroadcastRejected](#) ()

Test selective broadcasts.

- inform stub to stop accepting subscriptions
- try to subscribe to the selective broadcast
- check that an error was received
- inform stub to send a broadcast
- check that nothing was received in a reasonable time

Global AFSelective_SelectiveMultiBroadcast ()

Test multiple selective broadcasts.

- inform stub to start accepting subscriptions
- subscribe to the selective broadcast
- check that no error was received (in a reasonable time)
- inform stub to send a broadcast
- check that a correct value is received

Global CMAAttributes_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 CMAAttributes_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 CMAAttributes_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 CMAAttributes_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 CMAAttributes_AttributeSubscription ()

Test subscription API function for attributes

Global CMAAttributeSubscription_SubscribeAndUnsubscribeSequentially ()

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

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 **CMAAttributeSubscription_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 **CMAAttributeSubscription_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 **CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMAAttributeSubscription_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 `CMBroadcasts_BroadcastStubGoesOfflineOnlineAgain ()`

Test BroadcastStubGoesOfflineOnlineAgain.

- service offline
- subscribe to broadcast
- service online

- fire broadcast -> proxy should receive
- service offline
- service online
- fire again -> proxy should receive again

Global **CMBroadcasts_NormalBroadcast ()**

Test broadcasts. Subscribe to a broadcast, and see that the value is correctly received.

Global **CMBroadcasts_SelectiveBroadcast ()**

Test selective broadcasts.

- inform stub to start accepting subscriptions
- subscribe to the selective broadcast
- check that no error was received (in a reasonable time)
- inform stub to send a broadcast
- check that a correct value is received

Global **CMBroadcasts_SelectiveBroadcastRejected ()**

Test selective broadcasts.

- inform stub to stop accepting subscriptions
- try to subscribe to the selective broadcast
- check that an error was received
- inform stub to send a broadcast
- check that nothing was received in a reasonable time

Global **CMBroadcasts_SelectiveBroadcastStubGoesOfflineOnlineAgain ()**

Test SelectiveBroadcastStubGoesOfflineOnlineAgain.

- service offline
- subscribe to selective broadcast
- service online
- fire selective broadcast -> proxy should receive
- service offline
- service online
- fire again -> proxy should receive again

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_AttributeSetAsyncInvalid \(\)](#)

Test attribute asynchronous functions with invalid values

- Call set asynch function of attributes with invalid types
- Callback should be called with error status
- Check that attribute value has not changed

Global [DTAdvanced_AttributeSetInvalid \(\)](#)

Test attribute functions with invalid values

- Call set function of attributes with invalid types
- Check that the attribute's value has not changed

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_EmptyBroadcastReceive \(\)](#)

Test broadcast with empty broadcast

- Subscribe to broadcast which does not contain any datatypes
- Call function twice to cause the stub to fire a broadcast event
- Check if the callback function was called twice

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 struc 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

Global **PFComplex_Ping_Pong_Complex_Synchronous ()**

Test synchronous ping pong function call

- complex array is array of a struct containing an union and another struc 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 **RTLodingRuntime_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 MESSAGE_SIZE 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 MESSAGE_SIZE 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 MESSAGE_SIZE 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
- Create MAXTHREADCOUNT threads, each of which creates a proxy for each service address and then sets attributes MAXMETHODCALLS times to each.
- Each attribute is MESSAGE_SIZE 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 MESSAGE_SIZE 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 MESSAGE_SIZE 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/ AF-Managed.cpp	19
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ AF-Polymorph.cpp	23
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ AF-Selective.cpp	24
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ CM-Attributes.cpp	26
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ CM-AttributeSubscription.cpp	28
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ CM-Broadcasts.cpp	33
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ CM-MethodCalls.cpp	35
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ DT-Advanced.cpp	36
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ DT-Combined.cpp	37
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ DT-Derived.cpp	38
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ DT-Primitive.cpp	39
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ PF-Complex.cpp	40
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ PF-Primitive.cpp	41
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ RT-BuildProxiesAndStubs.cpp	43
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ RT-LoadingRuntime.cpp	44
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ Stability-SP.cpp	44
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ TH-MainLoopIndependence.cpp	47
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ TH-MainLoopIntegration.cpp	48
/home/jg/work/sources/capi/ascgit017.CommonAPI-Tools/org.genivi.commonapi.core.verification/src/ TH-MainLoopTwoThreads.cpp	49

Chapter 4

File Documentation

4.1 mainpagetests/01_mainpage.dox File Reference

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

Functions

- void [AFManaged_AddRemoveManagedInterfaceSingle](#) ()
- void [AFManaged_AddRemoveManagedInterfaceMultiple](#) ()
- void [AFManaged_AddRemoveMultipleManagedInterfacesSingle](#) ()
- void [AFManaged_AddRemoveMultipleManagedInterfacesMultiple](#) ()
- void [AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationExplicit](#) ()
- void [AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationExplicitAll](#) ()
- void [AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationImplicit](#) ()
- void [AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationExplicit](#) ()
- void [AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationExplicitAll](#) ()
- void [AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationImplicit](#) ()
- void [AFManaged_ProxyManagerTestPrimitiveMethods](#) ()
- void [AFManaged_ProxyManagerTestNonPrimitiveMethodsSync](#) ()
- void [AFManaged_ProxyManagerTestNonPrimitiveMethodsAsync](#) ()
- void [AFManaged_DISABLED_ProxyManagerTestGetInstanceAvailabilityStatusAsync](#) ()
- int [main](#) (int argc, char **argv)

Variables

- const std::string & [domain](#) = "local"

4.2.1 Function Documentation

4.2.1.1 void [AFManaged_AddRemoveManagedInterfaceSingle](#) ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Remove the managed interface from the manager
 - Check that the client is notified about the removed interface

4.2.1.2 void AFManaged_AddRemoveManagedInterfaceMultiple ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Add a second instance of the same managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Remove all the managed interfaces from the manager
 - Check that the client is notified about the removed interfaces

4.2.1.3 void AFManaged_AddRemoveMultipleManagedInterfacesSingle ()

- Test**
- Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Add a different managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Remove all the managed interfaces from the manager
 - Check that the client is notified about the removed interfaces

4.2.1.4 void AFManaged_AddRemoveMultipleManagedInterfacesMultiple ()

- Test**
- Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Add a different managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Add a second instance of the same managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Remove all the managed interfaces from the manager
 - Check that the client is notified about the removed interfaces

4.2.1.5 void AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationExplicit ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Build a proxy through the manager to the managed device
 - Call a method on the managed device and check call status
 - Explicitly deregister managed interface through its instance name

4.2.1.6 void AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationExplicitAll ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Build a proxy through the manager to the managed device
 - Call a method on the managed device and check call status
 - Deregister all managed interfaces through manager's stub adapter

4.2.1.7 void AFManaged_BuildProxyThroughManagerAndMethodCallSingleDeregistrationImplicit ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add a managed interface to the manager
 - Check that the client is notified about the newly added interface
 - Build a proxy through the manager to the managed device
 - Call a method on the managed device and check call status
 - Don't deregister managed interfaces. This is done in dtor of manager's StubAdapterInternal when manager service is unregistered in TearDown() method.

4.2.1.8 void AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationExplicit ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add managed interfaces to the manager
 - Check that the client is notified about the newly added interfaces
 - Build proxies through the manager to the managed interfaces
 - Call a method on the managed interfaces and check call status
 - Explicitly deregister managed interfaces through their instance name

4.2.1.9 void AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationExplicitAll ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add managed interfaces to the manager
 - Check that the client is notified about the newly added interfaces
 - Build proxies through the manager to the managed interfaces
 - Call a method on the managed interfaces and check call status
 - Deregister all managed interfaces through manager's stub adapter

4.2.1.10 void AFManaged_BuildProxyThroughManagerAndMethodCallMultipleDeregistrationImplicit ()

- Test**
- Subscribe on the events about availability status changes at the manager
 - Add managed interfaces to the manager
 - Check that the client is notified about the newly added interfaces
 - Build proxies through the manager to the managed interfaces
 - Call a method on the managed interfaces and check call status
 - Don't deregister managed interfaces. This is done in dtor of manager's StubAdapterInternal when manager service is unregistered in TearDown() method.

4.2.1.11 void AFManaged_ProxyManagerTestPrimitiveMethods ()

- Test**
- Test the getConnectionId, getDomain and getInterface methods available via the ProxyManager of the respective managed interfaces of the manager

4.2.1.12 void AFManaged_ProxyManagerTestNonPrimitiveMethodsSync ()

Test

- Add a managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatus method to check that all returned instances by getAvailableInstances are available
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatus method to check that all returned instances by getAvailableInstances are available
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatus method to check that all returned instances by getAvailableInstances are available
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

4.2.1.13 void AFManaged_ProxyManagerTestNonPrimitiveMethodsAsync ()

Test

- Add a managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstancesAsync method to check that all registered instances are returned
- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstancesAsync method to check that all registered instances are returned
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstancesAsync method to check that all registered instances are returned
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

4.2.1.14 void AFManaged_DISABLED_ProxyManagerTestGetInstanceAvailabilityStatusAsync ()

Test

- Add a managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatusAsync method to check that all returned instances by getAvailableInstances are available

- Add a different managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatusAsync method to check that all returned instances by getAvailableInstances are available
- Add a second instance of the same managed interface to the manager
- Check that the client is notified about the newly added interface
- Use the ProxyManager's getAvailableInstances method to check that all registered instances are returned
- Use the ProxyManager's checkInstanceAvailabilityStatusAsync method to check that all returned instances by getAvailableInstances are available
- Remove all the managed interfaces from the manager
- Check that the client is notified about the removed interfaces

4.2.1.15 `int main (int argc, char ** argv)`

4.2.2 Variable Documentation

4.2.2.1 `const std::string& domain = "local"`

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

Functions

- void [AFPPolymorph_SetAndGetAttributeTypedef](#) ()
- void [AFPPolymorph_SetAndGetAttributeEnum](#) ()
- void [AFPPolymorph_SetAndGetAttributeUInt](#) ()
- void [AFPPolymorph_SetAndGetAttributeString](#) ()
- void [AFPPolymorph_SetAndGetAttributeStruct](#) ()
- void [AFPPolymorph_MethodCall](#) ()
- void [AFPPolymorph_Broadcast](#) ()
- int [main](#) (int argc, char **argv)

Variables

- const std::string [domain](#) = "local"
- const std::string [testAddress](#) = "commonapi.advanced.polymorph.TestInterface"
- const std::string [connectionId_client](#) = "client-sample"
- const std::string [connectionId_service](#) = "service-sample"

4.3.1 Function Documentation

4.3.1.1 `void AFPPolymorph_SetAndGetAttributeTypedef ()`

Test

- Set and get a typedef-type attribute through a polymorphic structure
- verify that the received data matches the transmitted data

4.3.1.2 void AFPolymorph_SetAndGetAttributeEnum ()

- Test**
- Set and get a enum-type attribute through a polymorphic structure
 - verify that the received data matches the transmitted data

4.3.1.3 void AFPolymorph_SetAndGetAttributeUInt ()

- Test**
- Set and get a uint-type attribute through a polymorphic structure
 - verify that the received data matches the transmitted data

4.3.1.4 void AFPolymorph_SetAndGetAttributeString ()

- Test**
- Set and get a string-type attribute through a polymorphic structure
 - verify that the received data matches the transmitted data

4.3.1.5 void AFPolymorph_SetAndGetAttributeStruct ()

- Test**
- Set and get a struct-type attribute through a polymorphic structure
 - verify that the received data matches the transmitted data

4.3.1.6 void AFPolymorph_MethodCall ()

- Test**
- Call a method whose input and output parameters are polymorphic structures
 - verify that the received data matches the transmitted data

4.3.1.7 void AFPolymorph_Broadcast ()

- Test**
- Call a method with a special value that tells the stub to send a broadcast signal
 - verify that the received data matches the transmitted data

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

4.3.2 Variable Documentation

4.3.2.1 const std::string domain = "local"

4.3.2.2 const std::string testAddress = "commonapi.advanced.polymorph.TestInterface"

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

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

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

Functions

- void [AFSelective_SelectiveBroadcastRejected](#) ()

- void [AFSelective_SelectiveBroadcast](#) ()
- void [AFSelective_SelectiveMultiBroadcast](#) ()
- void [AFSelective_DISABLED_SelectiveRejectedMultiBroadcast](#) ()
- int [main](#) (int argc, char **argv)

Variables

- const std::string [serviceId](#) = "service-sample"
- const std::string [clientId](#) = "client-sample"
- const std::string [otherclientId](#) = "other-client-sample"
- const std::string [domain](#) = "local"
- const std::string [testAddress](#) = "commonapi.advanced.bselective.TestInterface"
- const int [tasync](#) = 100000

4.4.1 Function Documentation

4.4.1.1 void [AFSelective_SelectiveBroadcastRejected](#) ()

Test Test selective broadcasts.

- inform stub to stop accepting subscriptions
- try to subscribe to the selective broadcast
- check that an error was received
- inform stub to send a broadcast
- check that nothing was received in a reasonable time

4.4.1.2 void [AFSelective_SelectiveBroadcast](#) ()

Test Test selective broadcasts.

- inform stub to start accepting subscriptions
- subscribe to the selective broadcast
- check that no error was received (in a reasonable time)
- inform stub to send a broadcast
- check that a correct value is received

4.4.1.3 void [AFSelective_SelectiveMultiBroadcast](#) ()

Test Test multiple selective broadcasts.

- inform stub to start accepting subscriptions
- subscribe to the selective broadcast
- check that no error was received (in a reasonable time)
- inform stub to send a broadcast
- check that a correct value is received

4.4.1.4 void AFSelective_DISABLED_SelectiveRejectedMultiBroadcast ()

Test Test multiple selective broadcasts, with rejection.

- subscribe to stub three times: once from proxy2, once from proxy1 (accepted) once from proxy2 (rejected)
- This should result with two subscription callbacks being called from broadcast.

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

4.4.2 Variable Documentation

4.4.2.1 const std::string servid = "service-sample"

4.4.2.2 const std::string clientid = "client-sample"

4.4.2.3 const std::string otherclientid = "other-client-sample"

4.4.2.4 const std::string domain = "local"

4.4.2.5 const std::string testAddress = "commonapi.advanced.bselective.TestInterface"

4.4.2.6 const int tasync = 100000

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

Functions

- void [CMAAttributes_AttributeGetSynchronous](#) ()
- void [CMAAttributes_AttributeGetAsynchronous](#) ()
- void [CMAAttributes_AttributeSetSynchronous](#) ()
- void [CMAAttributes_AttributeSetAsynchronous](#) ()
- void [CMAAttributes_AttributeSubscription](#) ()
- int [main](#) (int argc, char **argv)

Variables

- const std::string [servid](#) = "service-sample"
- const std::string [clientid](#) = "client-sample"
- const std::string [domain](#) = "local"
- const std::string [testAddress](#) = "commonapi.communication.TestInterface"
- const int [tasync](#) = 100000

4.5.1 Function Documentation

4.5.1.1 void CMAAttributes_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.

4.5.1.2 void CMAAttributes_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.

4.5.1.3 void CMAAttributes_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.

4.5.1.4 void CMAAttributes_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.

4.5.1.5 void CMAAttributes_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.

4.5.1.6 `int main (int argc, char ** argv)`

4.5.2 Variable Documentation

4.5.2.1 `const std::string servid = "service-sample"`

4.5.2.2 `const std::string clientid = "client-sample"`

4.5.2.3 `const std::string domain = "local"`

4.5.2.4 `const std::string testAddress = "commonapi.communication.TestInterface"`

4.5.2.5 `const int tasync = 100000`

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

Typedefs

- `typedef std::shared_ptr`
`< v1_0::commonapi::communication::TestInterfaceProxy<> > ProxyPtr`

Functions

- `void testSubscription (ProxyPtr pp)`
- `void CMAAttributeSubscription_SubscriptionStandard ()`
- `void CMAAttributeSubscription_SubscriptionOnAvailable ()`
- `void CMAAttributeSubscription_SubscriptionMultithreading ()`
- `void CMAAttributeSubscription_SubscriptionUnsubscribeFromCallback ()`
- `void CMAAttributeSubscription_SubscribeAndUnsubscribeTwoCallbacksCoexistent ()`
- `void CMAAttributeSubscription_SubscribeAndUnsubscribeSequentially ()`
- `void CMAAttributeSubscription_SubscribeAndUnsubscribeUnsubscribe ()`
- `void CMAAttributeSubscription_SubscribeServiceNotAvailable ()`
- `void CMAAttributeSubscription_SubscribeUnregisterSetValueRegisterService ()`
- `void CMAAttributeSubscription_SubscribeUnregisterNoValueSetRegisterService ()`
- `void CMAAttributeSubscription_SubscribeSecondProxyLater ()`
- `void CMAAttributeSubscription_SubscribeThreeCallbacksServiceNotAvailable ()`
- `void CMAAttributeSubscription_SubscribeThreeCallbacksServiceAvailable ()`
- `int main (int argc, char **argv)`

Variables

- `const std::string daemonId = "service-sample"`
- `const std::string clientId = "client-sample"`
- `const std::string servid = "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`

4.6.1 Typedef Documentation

4.6.1.1 `typedef std::shared_ptr<v1_0::commonapi::communication::TestInterfaceProxy<> > ProxyPtr`

4.6.2 Function Documentation

4.6.2.1 `void testSubscription (ProxyPtr pp)`

4.6.2.2 `void CMAAttributeSubscription_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.

4.6.2.3 `void CMAAttributeSubscription_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.

4.6.2.4 `void CMAAttributeSubscription_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.

4.6.2.5 `void CMAAttributeSubscription_SubscriptionUnsubscribeFromCallback ()`

Test 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.

4.6.2.6 void CMAAttributeSubscription_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

4.6.2.7 void CMAAttributeSubscription_SubscribeAndUnsubscribeSequentially ()

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

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

4.6.2.8 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

4.6.2.9 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

4.6.2.10 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

4.6.2.11 void CMAAttributeSubscription_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

4.6.2.12 void CMAAttributeSubscription_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

4.6.2.13 void CMAAttributeSubscription_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

4.6.2.14 void CMAAttributeSubscription_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

4.6.2.15 `int main (int argc, char ** argv)`

4.6.3 Variable Documentation

4.6.3.1 `const std::string daemonId = "service-sample"`

4.6.3.2 `const std::string clientId = "client-sample"`

4.6.3.3 `const std::string serviceId = "test-service"`

4.6.3.4 `const std::string domain = "local"`

4.6.3.5 `const std::string testAddress = "commonapi.communication.TestInterface"`

4.6.3.6 `const std::string daemonAddress = "commonapi.communication.Daemon"`

4.6.3.7 `const unsigned int wt = 100000`

4.6.3.8 `std::mutex mut`

4.6.3.9 `std::deque<uint32_t> data_queue`

4.6.3.10 `std::condition_variable data_cond`

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

Functions

- void [CMBroadcasts_NormalBroadcast](#) ()
- void [CMBroadcasts_SelectiveBroadcastRejected](#) ()
- void [CMBroadcasts_SelectiveBroadcast](#) ()
- void [CMBroadcasts_BroadcastStubGoesOfflineOnlineAgain](#) ()
- void [CMBroadcasts_SelectiveBroadcastStubGoesOfflineOnlineAgain](#) ()
- 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

4.7.1 Function Documentation

4.7.1.1 `void CMBroadcasts_NormalBroadcast ()`

Test Test broadcasts. Subscribe to a broadcast, and see that the value is correctly received.

4.7.1.2 void CMBroadcasts_SelectiveBroadcastRejected ()

Test Test selective broadcasts.

- inform stub to stop accepting subscriptions
- try to subscribe to the selective broadcast
- check that an error was received
- inform stub to send a broadcast
- check that nothing was received in a reasonable time

4.7.1.3 void CMBroadcasts_SelectiveBroadcast ()

Test Test selective broadcasts.

- inform stub to start accepting subscriptions
- subscribe to the selective broadcast
- check that no error was received (in a reasonable time)
- inform stub to send a broadcast
- check that a correct value is received

4.7.1.4 void CMBroadcasts_BroadcastStubGoesOfflineOnlineAgain ()

Test Test BroadcastStubGoesOfflineOnlineAgain.

- service offline
- subscribe to broadcast
- service online
- fire broadcast -> proxy should receive
- service offline
- service online
- fire again -> proxy should receive again

4.7.1.5 void CMBroadcasts_SelectiveBroadcastStubGoesOfflineOnlineAgain ()

Test Test SelectiveBroadcastStubGoesOfflineOnlineAgain.

- service offline
- subscribe to selective broadcast
- service online
- fire selective broadcast -> proxy should receive
- service offline
- service online
- fire again -> proxy should receive again

4.7.1.6 `int main (int argc, char ** argv)`

4.7.2 Variable Documentation

4.7.2.1 `const std::string servid = "service-sample"`

4.7.2.2 `const std::string clientid = "client-sample"`

4.7.2.3 `const std::string domain = "local"`

4.7.2.4 `const std::string testAddress = "commonapi.communication.TestInterface"`

4.7.2.5 `const int tasync = 100000`

4.8 /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 [servid](#) = "service-sample"
- const std::string [clientid](#) = "client-sample"
- const std::string [domain](#) = "local"
- const std::string [testAddress](#) = "commonapi.communication.TestInterface"
- const int [tasync](#) = 100000

4.8.1 Function Documentation

4.8.1.1 `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.

4.8.1.2 `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.

4.8.1.3 `int main (int argc, char ** argv)`

4.8.2 Variable Documentation

4.8.2.1 `const std::string servid = "service-sample"`

4.8.2.2 `const std::string clientid = "client-sample"`

4.8.2.3 `const std::string domain = "local"`

4.8.2.4 `const std::string testAddress = "commonapi.communication.TestInterface"`

4.8.2.5 `const int tasync = 100000`

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

Functions

- void [DTAdvanced_SendAndReceive](#) ()
- void [DTAdvanced_SendAndReceiveInvalid](#) ()
- void [DTAdvanced_AttributeSetInvalid](#) ()
- void [DTAdvanced_AttributeSetAsyncInvalid](#) ()
- 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"

4.9.1 Function Documentation

4.9.1.1 `void DTAdvanced_SendAndReceive ()`

4.9.1.2 `void DTAdvanced_SendAndReceiveInvalid ()`

4.9.1.3 `void DTAdvanced_AttributeSetInvalid ()`

Test Test attribute functions with invalid values

- Call set function of attributes with invalid types
- Check that the attribute's value has not changed

4.9.1.4 `void DTAdvanced_AttributeSetAsyncInvalid ()`

Test Test attribute asynchronous functions with invalid values

- Call set asynch function of attributes with invalid types

- Callback should be called with error status
- Check that attribute value has not changed

4.9.1.5 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

4.9.1.6 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

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

4.9.2 Variable Documentation

4.9.2.1 const std::string domain = "local"

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

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

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

4.10 /home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.- core.verification/src/DTCCombined.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"

4.10.1 Function Documentation

4.10.1.1 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

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

4.10.2 Variable Documentation

4.10.2.1 const std::string domain = "local"

4.10.2.2 const std::string testAddress = "commonapi.datatypes.combined.TestInterface"

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

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

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

Functions

- void [DTDerived_SendAndReceive](#) ()
- void [DTDerived_AttributeSet](#) ()
- void [DTDerived_BroadcastReceive](#) ()
- int [main](#) (int argc, char **argv)

Variables

- const std::string [domain](#) = "local"
- const std::string [testAddress](#) = "commonapi.datatypes.derived.TestInterface"
- const std::string [connectionId_client](#) = "client-sample"
- const std::string [connectionId_service](#) = "service-sample"

4.11.1 Function Documentation

4.11.1.1 void DTDerived_SendAndReceive ()

4.11.1.2 void DTDerived_AttributeSet ()

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

4.11.1.3 void DTDerived_BroadcastReceive ()

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

4.11.1.4 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.datatypes.derived.TestInterface"

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

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

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

Functions

- void [DTPPrimitive_SendAndReceive](#) ()
- void [DTPPrimitive_AttributeSet](#) ()
- void [DTPPrimitive_BroadcastReceive](#) ()
- void [DTPPrimitive_EmptyBroadcastReceive](#) ()
- 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"

4.12.1 Function Documentation

4.12.1.1 void DTPPrimitive_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

4.12.1.2 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

4.12.1.3 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

4.12.1.4 void DTPrimitive_EmptyBroadcastReceive ()

Test Test broadcast with empty broadcast

- Subscribe to broadcast which does not contain any datatypes
- Call function twice to cause the stub to fire a broadcast event
- Check if the callback function was called twice

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

4.12.2 Variable Documentation

4.12.2.1 const std::string domain = "local"

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

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

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

4.13 /home/jg/work/sources/capi/asccgit017.CommonAPI-Tools/org.genivi.commonapi.-core.verifcation/src/PFComplex.cpp File Reference

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

4.13.1 Function Documentation

4.13.1.1 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 struc 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

4.13.1.2 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 struc 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

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

4.13.2 Variable Documentation

4.13.2.1 const int usecPerSecond = 1000000

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

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

4.13.2.4 const std::string domain = "local"

4.13.2.5 const std::string testAddress = "commonapi.performance.complex.TestInterface"

4.13.2.6 const int tasync = 100000

4.13.2.7 const int maxArraySize = 4096 / 16

4.13.2.8 const int loopCountPerPayload = 1000

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

Functions

- void `PFPPrimitive_Ping_Pong_Primitive_Synchronous()`
- void `PFPPrimitive_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

4.14.1 Function Documentation

4.14.1.1 void PFPPrimitive_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

4.14.1.2 void PFPPrimitive_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

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

4.14.2 Variable Documentation

4.14.2.1 const std::string serviceld = "service-sample"

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

4.14.2.3 const std::string domain = "local"

4.14.2.4 const std::string testAddress = "commonapi.performance.primitive.TestInterface"

4.14.2.5 `const int tasync = 100000`

4.14.2.6 `const int usecPerSecond = 1000000`

4.14.2.7 `const int maxPrimitiveArraySize = 1024*16`

4.14.2.8 `const int loopCountPerPayload = 1000`

4.15 /home/jg/work/sources/capi/asggit017.CommonAPI-Tools/org.genivi.commonapi.- 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"

4.15.1 Function Documentation

4.15.1.1 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.

4.15.1.2 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

4.15.1.3 `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 testAddress = "commonapi.runtime.TestInterface"`

4.15.2.3 `const std::string applicationNameService = "service-sample"`

4.15.2.4 `const std::string applicationNameClient = "client-sample"`

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

Functions

- void [RTLoadingRuntime_LoadsDefaultRuntime](#) ()
- int [main](#) (int argc, char **argv)

4.16.1 Function Documentation

4.16.1.1 `void RTLoadingRuntime_LoadsDefaultRuntime ()`

Test Loads Default Runtime.

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

4.16.1.2 `int main (int argc, char ** argv)`

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

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)

Variables

- const std::string [serviceld](#) = "service-sample"
- const std::string [clientId](#) = "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 MESSAGE_SIZE = 80
- const int MAXSUBSCRIPTIONSETS = 10

4.17.1 Function Documentation

4.17.1.1 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

4.17.1.2 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 MESSAGE_SIZE 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

4.17.1.3 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 MESSAGE_SIZE 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

4.17.1.4 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 MESSAGE_SIZE 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

4.17.1.5 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 MESSAGE_SIZE 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

4.17.1.6 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 MESSAGE_SIZE 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

4.17.1.7 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 MESSAGE_SIZE 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

4.17.1.8 `int main (int argc, char ** argv)`

4.17.2 Variable Documentation

4.17.2.1 `const std::string serviceId = "service-sample"`

4.17.2.2 `const std::string clientId = "client-sample"`

4.17.2.3 `const std::string domain = "local"`

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

4.17.2.5 `const std::string COMMONAPI_CONFIG_SUFFIX = ".conf"`

4.17.2.6 `const int MAXSERVERCOUNT = 40`

4.17.2.7 `const int MAXTHREADCOUNT = 8`

4.17.2.8 `const int MAXMETHODCALLS = 80`

4.17.2.9 `const int MAXREGLOOPS = 16`

4.17.2.10 `const int MAXREGCOUNT = 16`

4.17.2.11 `const int MESSAGE_SIZE = 80`

4.17.2.12 `const int MAXSUBSCRIPTIONSETS = 10`

4.18 /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"

4.18.1 Function Documentation

4.18.1.1 `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

4.18.1.2 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

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

4.18.2 Variable Documentation

4.18.2.1 const std::string domain = "local"

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

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

4.18.2.4 const std::string instance8 = "my.test.commonapi.address.eight"

4.18.2.5 const std::string mainloopName1 = "client-sample"

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

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

4.19 /home/jg/work/sources/capi/asccgit017.CommonAPI-Tools/org.genivi.commonapi.-core.verifcation/src/THMainLoopIntegration.cpp File Reference

Functions

- void [THMainLoopIntegration_VerifyCommunicationWithMainLoop](#) ()
- void [THMainLoopIntegration_VerifyTransportReading](#) ()
- void [THMainLoopIntegration_VerifySyncCallMessageHandlingOrder](#) ()
- int [main](#) (int argc, char **argv)

Variables

- 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"

4.19.1 Function Documentation

4.19.1.1 void THMainLoopIntegration_VerifyCommunicationWithMainLoop ()

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.

4.19.1.2 void THMainLoopIntegration_VerifyTransportReading ()

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.

4.19.1.3 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

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

4.19.2 Variable Documentation

4.19.2.1 const std::string domain = "local"

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

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

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

4.20 /home/jg/work/sources/capi/asggit017.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"

4.20.1 Function Documentation

4.20.1.1 void `THMainLoopTwoThreads_ProxyGetsAvailableStatus` ()

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

4.20.1.2 void `THMainLoopTwoThreads_ProxyGetsFunctionResponse` ()

Test Proxy gets function response when MainLoop Dispatched sourced out to other thread.

4.20.1.3 int `main` (int *argc*, char ** *argv*)

4.20.2 Variable Documentation

4.20.2.1 const std::string `domain` = "local"

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

Index

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
AFManaged.cpp, [19](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
AFPolymorph.cpp, [23](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
AFSelective.cpp, [24](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
CMAAttributeSubscription.cpp, [28](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
CMAAttributes.cpp, [26](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
CMBroadcasts.cpp, [33](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
CMMethodCalls.cpp, [35](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
DTAdvanced.cpp, [36](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
DTCombined.cpp, [37](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
DTDerived.cpp, [38](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
DTPrimitive.cpp, [39](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
PFCComplex.cpp, [40](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
PFPPrimitive.cpp, [41](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
RTBuildProxiesAndStubs.cpp, [43](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
RTLLoadingRuntime.cpp, [44](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
StabilitySP.cpp, [44](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
THMainLoopIndependence.cpp, [47](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
THMainLoopIntegration.cpp, [48](#)

/home/jg/work/sources/capi/asggit017.CommonAPI--
Tools/org.genivi.commonapi.core.verification/src/-
THMainLoopTwoThreads.cpp, [49](#)

AFManaged.cpp

AFManaged_AddRemoveManagedInterface-
Multiple, [19](#)

AFManaged_AddRemoveManagedInterface-
Single, [19](#)

AFManaged_AddRemoveMultipleManaged-
InterfacesMultiple, [20](#)

AFManaged_AddRemoveMultipleManaged-
InterfacesSingle, [20](#)

AFManaged_BuildProxyThroughManagerAnd-
MethodCallMultipleDeregistrationExplicit, [21](#)

AFManaged_BuildProxyThroughManagerAnd-
MethodCallMultipleDeregistrationExplicitAll, [21](#)

AFManaged_BuildProxyThroughManagerAnd-
MethodCallMultipleDeregistrationImplicit, [21](#)

AFManaged_BuildProxyThroughManagerAnd-
MethodCallSingleDeregistrationExplicit, [20](#)

AFManaged_BuildProxyThroughManagerAnd-
MethodCallSingleDeregistrationExplicitAll, [20](#)

AFManaged_BuildProxyThroughManagerAnd-
MethodCallSingleDeregistrationImplicit, [20](#)

AFManaged_ProxyManagerTestNonPrimitive-
MethodsAsync, [22](#)

AFManaged_ProxyManagerTestNonPrimitive-
MethodsSync, [21](#)

AFManaged_ProxyManagerTestPrimitiveMethods, [21](#)

domain, [23](#)

main, [23](#)

AFManaged_AddRemoveManagedInterfaceMultiple
AFManaged.cpp, [19](#)

AFManaged_AddRemoveManagedInterfaceSingle
AFManaged.cpp, [19](#)

AFManaged_AddRemoveMultipleManagedInterfaces-
Multiple
AFManaged.cpp, [20](#)

AFManaged_AddRemoveMultipleManagedInterfaces-
Single
AFManaged.cpp, [20](#)

- AFManaged_BuildProxyThroughManagerAndMethod-
CallMultipleDeregistrationExplicit
AFManaged.cpp, 21
- AFManaged_BuildProxyThroughManagerAndMethod-
CallMultipleDeregistrationExplicitAll
AFManaged.cpp, 21
- AFManaged_BuildProxyThroughManagerAndMethod-
CallMultipleDeregistrationImplicit
AFManaged.cpp, 21
- AFManaged_BuildProxyThroughManagerAndMethod-
CallSingleDeregistrationExplicit
AFManaged.cpp, 20
- AFManaged_BuildProxyThroughManagerAndMethod-
CallSingleDeregistrationExplicitAll
AFManaged.cpp, 20
- AFManaged_BuildProxyThroughManagerAndMethod-
CallSingleDeregistrationImplicit
AFManaged.cpp, 20
- AFManaged_ProxyManagerTestNonPrimitiveMethods-
Async
AFManaged.cpp, 22
- AFManaged_ProxyManagerTestNonPrimitiveMethods-
Sync
AFManaged.cpp, 21
- AFManaged_ProxyManagerTestPrimitiveMethods
AFManaged.cpp, 21
- AFPolymorph.cpp
 - AFPolymorph_Broadcast, 24
 - AFPolymorph_MethodCall, 24
 - AFPolymorph_SetAndGetAttributeEnum, 23
 - AFPolymorph_SetAndGetAttributeString, 24
 - AFPolymorph_SetAndGetAttributeStruct, 24
 - AFPolymorph_SetAndGetAttributeTypedef, 23
 - AFPolymorph_SetAndGetAttributeUInt, 24
 - connectionId_client, 24
 - connectionId_service, 24
 - domain, 24
 - main, 24
 - testAddress, 24
- AFPolymorph_Broadcast
AFPolymorph.cpp, 24
- AFPolymorph_MethodCall
AFPolymorph.cpp, 24
- AFPolymorph_SetAndGetAttributeEnum
AFPolymorph.cpp, 23
- AFPolymorph_SetAndGetAttributeString
AFPolymorph.cpp, 24
- AFPolymorph_SetAndGetAttributeStruct
AFPolymorph.cpp, 24
- AFPolymorph_SetAndGetAttributeTypedef
AFPolymorph.cpp, 23
- AFPolymorph_SetAndGetAttributeUInt
AFPolymorph.cpp, 24
- AFSelective.cpp
 - AFSelective_SelectiveBroadcast, 25
 - AFSelective_SelectiveBroadcastRejected, 25
 - AFSelective_SelectiveMultiBroadcast, 25
 - clientId, 26
 - domain, 26
 - main, 26
 - otherclientId, 26
 - serviceld, 26
 - tasync, 26
 - testAddress, 26
- AFSelective_DISABLED_SelectiveRejectedMulti-
Broadcast
AFSelective.cpp, 25
- AFSelective_SelectiveBroadcast
AFSelective.cpp, 25
- AFSelective_SelectiveBroadcastRejected
AFSelective.cpp, 25
- AFSelective_SelectiveMultiBroadcast
AFSelective.cpp, 25
- applicationNameClient
RTBuildProxiesAndStubs.cpp, 44
- applicationNameService
RTBuildProxiesAndStubs.cpp, 44
- CMAAttributeSubscription.cpp
 - CMAAttributeSubscription_SubscribeAndUnsubscribe-
Sequentially, 30
 - CMAAttributeSubscription_SubscribeAndUnsubscribe-
TwoCallbacksCoexistent, 30
 - CMAAttributeSubscription_SubscribeAndUnsubscribe-
Unsubscribe, 30
 - CMAAttributeSubscription_SubscribeSecondProxy-
Later, 32
 - CMAAttributeSubscription_SubscribeServiceNot-
Available, 31
 - CMAAttributeSubscription_SubscribeThreeCallbacks-
ServiceAvailable, 32
 - CMAAttributeSubscription_SubscribeThreeCallbacks-
ServiceNotAvailable, 32
 - CMAAttributeSubscription_SubscribeUnregisterNo-
ValueSetRegisterService, 31
 - CMAAttributeSubscription_SubscribeUnregisterSet-
ValueRegisterService, 31
 - CMAAttributeSubscription_SubscriptionMultithreading, 29
 - CMAAttributeSubscription_SubscriptionOnAvailable, 29
 - CMAAttributeSubscription_SubscriptionStandard, 29
 - CMAAttributeSubscription_SubscriptionUnsubscribe-
FromCallback, 29
 - clientId, 33
 - daemonAddress, 33
 - daemonId, 33
 - data_cond, 33
 - data_queue, 33
 - domain, 33
 - main, 32
 - mut, 33
 - ProxyPtr, 29
 - serviceld, 33
 - testAddress, 33
 - testSubscription, 29

- wt, [33](#)
- CMAAttributeSubscription_SubscribeAndUnsubscribe-Sequentially
 - CMAAttributeSubscription.cpp, [30](#)
- CMAAttributeSubscription_SubscribeAndUnsubscribe-TwoCallbacksCoexistent
 - CMAAttributeSubscription.cpp, [30](#)
- CMAAttributeSubscription_SubscribeAndUnsubscribe-Unsubscribe
 - CMAAttributeSubscription.cpp, [30](#)
- CMAAttributeSubscription_SubscribeSecondProxyLater
 - CMAAttributeSubscription.cpp, [32](#)
- CMAAttributeSubscription_SubscribeServiceNotAvailable
 - CMAAttributeSubscription.cpp, [31](#)
- CMAAttributeSubscription_SubscribeThreeCallbacks-ServiceAvailable
 - CMAAttributeSubscription.cpp, [32](#)
- CMAAttributeSubscription_SubscribeThreeCallbacks-ServiceNotAvailable
 - CMAAttributeSubscription.cpp, [32](#)
- CMAAttributeSubscription_SubscribeUnregisterNoValue-SetRegisterService
 - CMAAttributeSubscription.cpp, [31](#)
- CMAAttributeSubscription_SubscribeUnregisterSet-ValueRegisterService
 - CMAAttributeSubscription.cpp, [31](#)
- CMAAttributeSubscription_SubscriptionMultithreading
 - CMAAttributeSubscription.cpp, [29](#)
- CMAAttributeSubscription_SubscriptionOnAvailable
 - CMAAttributeSubscription.cpp, [29](#)
- CMAAttributeSubscription_SubscriptionStandard
 - CMAAttributeSubscription.cpp, [29](#)
- CMAAttributeSubscription_SubscriptionUnsubscribe-FromCallback
 - CMAAttributeSubscription.cpp, [29](#)
- CMAAttributes.cpp
 - CMAAttributes_AttributeGetAsynchronous, [27](#)
 - CMAAttributes_AttributeGetSynchronous, [26](#)
 - CMAAttributes_AttributeSetAsynchronous, [27](#)
 - CMAAttributes_AttributeSetSynchronous, [27](#)
 - CMAAttributes_AttributeSubscription, [27](#)
 - clientId, [28](#)
 - domain, [28](#)
 - main, [27](#)
 - serviceld, [28](#)
 - tasync, [28](#)
 - testAddress, [28](#)
- CMAAttributes_AttributeGetAsynchronous
 - CMAAttributes.cpp, [27](#)
- CMAAttributes_AttributeGetSynchronous
 - CMAAttributes.cpp, [26](#)
- CMAAttributes_AttributeSetAsynchronous
 - CMAAttributes.cpp, [27](#)
- CMAAttributes_AttributeSetSynchronous
 - CMAAttributes.cpp, [27](#)
- CMAAttributes_AttributeSubscription
 - CMAAttributes.cpp, [27](#)
- CMBroadcasts.cpp
 - CMBroadcasts_BroadcastStubGoesOfflineOnline-Again, [34](#)
 - CMBroadcasts_NormalBroadcast, [33](#)
 - CMBroadcasts_SelectiveBroadcast, [34](#)
 - CMBroadcasts_SelectiveBroadcastRejected, [33](#)
 - CMBroadcasts_SelectiveBroadcastStubGoes-OfflineOnlineAgain, [34](#)
 - clientId, [35](#)
 - domain, [35](#)
 - main, [34](#)
 - serviceld, [35](#)
 - tasync, [35](#)
 - testAddress, [35](#)
- CMBroadcasts_BroadcastStubGoesOfflineOnlineAgain
 - CMBroadcasts.cpp, [34](#)
- CMBroadcasts_NormalBroadcast
 - CMBroadcasts.cpp, [33](#)
- CMBroadcasts_SelectiveBroadcast
 - CMBroadcasts.cpp, [34](#)
- CMBroadcasts_SelectiveBroadcastRejected
 - CMBroadcasts.cpp, [33](#)
- CMBroadcasts_SelectiveBroadcastStubGoesOffline-OnlineAgain
 - CMBroadcasts.cpp, [34](#)
- CMMethodCalls.cpp
 - CMMethodCalls_AsynchronousMethodCall, [35](#)
 - CMMethodCalls_SynchronousMethodCall, [35](#)
 - clientId, [36](#)
 - domain, [36](#)
 - main, [35](#)
 - serviceld, [36](#)
 - tasync, [36](#)
 - testAddress, [36](#)
- CMMethodCalls_AsynchronousMethodCall
 - CMMethodCalls.cpp, [35](#)
- CMMethodCalls_SynchronousMethodCall
 - CMMethodCalls.cpp, [35](#)
- clientId
 - AFSelective.cpp, [26](#)
 - CMAAttributes.cpp, [28](#)
 - CMAAttributeSubscription.cpp, [33](#)
 - CMBroadcasts.cpp, [35](#)
 - CMMethodCalls.cpp, [36](#)
 - PFCComplex.cpp, [41](#)
 - PFPrimitive.cpp, [42](#)
 - StabilitySP.cpp, [47](#)
- connection_client
 - THMainLoopIntegration.cpp, [49](#)
- connection_service
 - THMainLoopIntegration.cpp, [49](#)
- connectionId_client
 - AFPPolymorph.cpp, [24](#)
 - DTDerived.cpp, [39](#)
- connectionId_service
 - AFPPolymorph.cpp, [24](#)
 - DTDerived.cpp, [39](#)
- connectionIdClient
 - DTAdvanced.cpp, [37](#)

- DTCombined.cpp, 38
- DTPrimitive.cpp, 40
- connectionIdService
 - DTAdvanced.cpp, 37
 - DTCombined.cpp, 38
 - DTPrimitive.cpp, 40
- DTAdvanced.cpp
 - connectionIdClient, 37
 - connectionIdService, 37
 - DTAdvanced_AttributeSet, 37
 - DTAdvanced_AttributeSetAsyncInvalid, 36
 - DTAdvanced_AttributeSetInvalid, 36
 - DTAdvanced_BroadcastReceive, 37
 - DTAdvanced_SendAndReceive, 36
 - DTAdvanced_SendAndReceiveInvalid, 36
 - domain, 37
 - main, 37
 - testAddress, 37
- DTAdvanced_AttributeSet
 - DTAdvanced.cpp, 37
- DTAdvanced_AttributeSetAsyncInvalid
 - DTAdvanced.cpp, 36
- DTAdvanced_AttributeSetInvalid
 - DTAdvanced.cpp, 36
- DTAdvanced_BroadcastReceive
 - DTAdvanced.cpp, 37
- DTAdvanced_SendAndReceive
 - DTAdvanced.cpp, 36
- DTAdvanced_SendAndReceiveInvalid
 - DTAdvanced.cpp, 36
- DTCombined.cpp
 - connectionIdClient, 38
 - connectionIdService, 38
 - DTCombined_SendAndReceive, 38
 - domain, 38
 - main, 38
 - testAddress, 38
- DTCombined_SendAndReceive
 - DTCombined.cpp, 38
- DTDerived.cpp
 - connectionId_client, 39
 - connectionId_service, 39
 - DTDerived_AttributeSet, 38
 - DTDerived_BroadcastReceive, 38
 - DTDerived_SendAndReceive, 38
 - domain, 39
 - main, 39
 - testAddress, 39
- DTDerived_AttributeSet
 - DTDerived.cpp, 38
- DTDerived_BroadcastReceive
 - DTDerived.cpp, 38
- DTDerived_SendAndReceive
 - DTDerived.cpp, 38
- DTPrimitive.cpp
 - connectionIdClient, 40
 - connectionIdService, 40
 - DTPrimitive_AttributeSet, 39
 - DTPrimitive_BroadcastReceive, 40
 - DTPrimitive_EmptyBroadcastReceive, 40
 - DTPrimitive_SendAndReceive, 39
 - domain, 40
 - main, 40
 - testAddress, 40
- DTPrimitive_AttributeSet
 - DTPrimitive.cpp, 39
- DTPrimitive_BroadcastReceive
 - DTPrimitive.cpp, 40
- DTPrimitive_EmptyBroadcastReceive
 - DTPrimitive.cpp, 40
- DTPrimitive_SendAndReceive
 - DTPrimitive.cpp, 39
- daemonAddress
 - CMAAttributeSubscription.cpp, 33
- daemonId
 - CMAAttributeSubscription.cpp, 33
- data_cond
 - CMAAttributeSubscription.cpp, 33
- data_queue
 - CMAAttributeSubscription.cpp, 33
- domain
 - AFManaged.cpp, 23
 - AFPPolymorph.cpp, 24
 - AFSelective.cpp, 26
 - CMAAttributes.cpp, 28
 - CMAAttributeSubscription.cpp, 33
 - CMBroadcasts.cpp, 35
 - CMMMethodCalls.cpp, 36
 - DTAdvanced.cpp, 37
 - DTCombined.cpp, 38
 - DTDerived.cpp, 39
 - DTPrimitive.cpp, 40
 - PFCComplex.cpp, 41
 - PFPPrimitive.cpp, 42
 - RTBuildProxiesAndStubs.cpp, 44
 - StabilitySP.cpp, 47
 - THMainLoopIndependence.cpp, 48
 - THMainLoopIntegration.cpp, 49
 - THMainLoopTwoThreads.cpp, 50
- instance
 - THMainLoopIntegration.cpp, 49
 - THMainLoopTwoThreads.cpp, 50
- instance6
 - THMainLoopIndependence.cpp, 48
- instance7
 - THMainLoopIndependence.cpp, 48
- instance8
 - THMainLoopIndependence.cpp, 48
- loopCountPerPayload
 - PFCComplex.cpp, 41
 - PFPPrimitive.cpp, 43
- MAXMETHODCALLS
 - StabilitySP.cpp, 47
- MAXREGCOUNT

- StabilitySP.cpp, 47
- MAXREGLOOPS
 - StabilitySP.cpp, 47
- MAXSERVERCOUNT
 - StabilitySP.cpp, 47
- MAXSUBSCRIPTIONSETS
 - StabilitySP.cpp, 47
- MAXTHREADCOUNT
 - StabilitySP.cpp, 47
- MESSAGESIZE
 - StabilitySP.cpp, 47
- main
 - AFManaged.cpp, 23
 - AFPPolymorph.cpp, 24
 - AFSelective.cpp, 26
 - CMAAttributes.cpp, 27
 - CMAAttributeSubscription.cpp, 32
 - CMBroadcasts.cpp, 34
 - CMMethodCalls.cpp, 35
 - DTAdvanced.cpp, 37
 - DTCombined.cpp, 38
 - DTDerived.cpp, 39
 - DTPrimitive.cpp, 40
 - PFCComplex.cpp, 41
 - PFFPrimitive.cpp, 42
 - RTBuildProxiesAndStubs.cpp, 43
 - RTLLoadingRuntime.cpp, 44
 - StabilitySP.cpp, 46
 - THMainLoopIndependence.cpp, 48
 - THMainLoopIntegration.cpp, 49
 - THMainLoopTwoThreads.cpp, 50
- mainloopName1
 - THMainLoopIndependence.cpp, 48
- mainloopName2
 - THMainLoopIndependence.cpp, 48
- mainpagetests/01_mainpage.dox, 19
- maxArraySize
 - PFCComplex.cpp, 41
- maxPrimitiveArraySize
 - PFFPrimitive.cpp, 43
- mut
 - CMAAttributeSubscription.cpp, 33
- otherclientId
 - AFSelective.cpp, 26
- PFCComplex.cpp
 - clientId, 41
 - domain, 41
 - loopCountPerPayload, 41
 - main, 41
 - maxArraySize, 41
 - PFCComplex_Ping_Pong_Complex_Asynchronous, 41
 - PFCComplex_Ping_Pong_Complex_Synchronous, 41
 - servicId, 41
 - tasync, 41
 - testAddress, 41
 - usecPerSecond, 41
- PFCComplex_Ping_Pong_Complex_Asynchronous
 - PFCComplex.cpp, 41
- PFCComplex_Ping_Pong_Complex_Synchronous
 - PFCComplex.cpp, 41
- PFFPrimitive.cpp
 - clientId, 42
 - domain, 42
 - loopCountPerPayload, 43
 - main, 42
 - maxPrimitiveArraySize, 43
 - PFFPrimitive_Ping_Pong_Primitive_Asynchronous, 42
 - PFFPrimitive_Ping_Pong_Primitive_Synchronous, 42
 - servicId, 42
 - tasync, 42
 - testAddress, 42
 - usecPerSecond, 43
- PFFPrimitive_Ping_Pong_Primitive_Asynchronous
 - PFFPrimitive.cpp, 42
- PFFPrimitive_Ping_Pong_Primitive_Synchronous
 - PFFPrimitive.cpp, 42
- ProxyPtr
 - CMAAttributeSubscription.cpp, 29
- RTBuildProxiesAndStubs.cpp
 - applicationNameClient, 44
 - applicationNameService, 44
 - domain, 44
 - main, 43
 - RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes, 43
 - testAddress, 44
- RTBuildProxiesAndStubs_BuildProxiesAndStubsTwoTimes
 - RTBuildProxiesAndStubs.cpp, 43
- RTBuildProxiesAndStubs_LoadedRuntimeCanBuildProxiesAndStubs
 - RTBuildProxiesAndStubs.cpp, 43
- RTLLoadingRuntime.cpp
 - main, 44
 - RTLLoadingRuntime_LoadsDefaultRuntime, 44
- RTLLoadingRuntime_LoadsDefaultRuntime
 - RTLLoadingRuntime.cpp, 44
- servicId
 - AFSelective.cpp, 26
 - CMAAttributes.cpp, 28
 - CMAAttributeSubscription.cpp, 33
 - CMBroadcasts.cpp, 35
 - CMMethodCalls.cpp, 36
 - PFCComplex.cpp, 41
 - PFFPrimitive.cpp, 42
 - StabilitySP.cpp, 47
- StabilitySP.cpp
 - clientId, 47
 - domain, 47
 - MAXMETHODCALLS, 47

- MAXREGCOUNT, [47](#)
- MAXREGLOOPS, [47](#)
- MAXSERVERCOUNT, [47](#)
- MAXSUBSCRIPTIONSETS, [47](#)
- MAXTHREADCOUNT, [47](#)
- MESSAGE_SIZE, [47](#)
- main, [46](#)
- serviceld, [47](#)
- StabilitySP_MultipleAttributeGetAsyncs, [46](#)
- StabilitySP_MultipleAttributeGets, [45](#)
- StabilitySP_MultipleAttributeSetAsyncs, [46](#)
- StabilitySP_MultipleAttributeSets, [45](#)
- StabilitySP_MultipleAttributeSubscriptions, [46](#)
- StabilitySP_MultipleMethodCalls, [45](#)
- StabilitySP_RepeatedRegistrations, [45](#)
- testAddress, [47](#)
- StabilitySP_MultipleAttributeGetAsyncs
 - StabilitySP.cpp, [46](#)
- StabilitySP_MultipleAttributeGets
 - StabilitySP.cpp, [45](#)
- StabilitySP_MultipleAttributeSetAsyncs
 - StabilitySP.cpp, [46](#)
- StabilitySP_MultipleAttributeSets
 - StabilitySP.cpp, [45](#)
- StabilitySP_MultipleAttributeSubscriptions
 - StabilitySP.cpp, [46](#)
- StabilitySP_MultipleMethodCalls
 - StabilitySP.cpp, [45](#)
- StabilitySP_RepeatedRegistrations
 - StabilitySP.cpp, [45](#)
- THMainLoopIndependence.cpp
 - domain, [48](#)
 - instance6, [48](#)
 - instance7, [48](#)
 - instance8, [48](#)
 - main, [48](#)
 - mainloopName1, [48](#)
 - mainloopName2, [48](#)
 - THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers, [48](#)
 - thirdPartyServiceld, [48](#)
- THMainLoopIndependence_ProxyReceivesAnswerOnlyIfStubMainLoopRuns
 - THMainLoopIndependence.cpp, [47](#)
- THMainLoopIndependence_ProxyReceivesJustHisOwnAnswers
 - THMainLoopIndependence.cpp, [48](#)
- THMainLoopIntegration.cpp
 - connection_client, [49](#)
 - connection_service, [49](#)
 - domain, [49](#)
 - instance, [49](#)
 - main, [49](#)
 - THMainLoopIntegration_VerifyCommunicationWithMainLoop, [49](#)
 - THMainLoopIntegration_VerifySyncCallMessageHandlingOrder, [49](#)
 - THMainLoopIntegration_VerifyTransportReading, [49](#)
 - THMainLoopIntegration_VerifyCommunicationWithMainLoop
 - THMainLoopIntegration.cpp, [49](#)
 - THMainLoopIntegration_VerifySyncCallMessageHandlingOrder
 - THMainLoopIntegration.cpp, [49](#)
 - THMainLoopIntegration_VerifyTransportReading
 - THMainLoopIntegration.cpp, [49](#)
- THMainLoopTwoThreads.cpp
 - domain, [50](#)
 - instance, [50](#)
 - main, [50](#)
 - THMainLoopTwoThreads_ProxyGetsAvailableStatus, [50](#)
 - THMainLoopTwoThreads_ProxyGetsFunctionResponse, [50](#)
 - THMainLoopTwoThreads_ProxyGetsAvailableStatus
 - THMainLoopTwoThreads.cpp, [50](#)
 - THMainLoopTwoThreads_ProxyGetsFunctionResponse
 - THMainLoopTwoThreads.cpp, [50](#)
- tasync
 - AFSelective.cpp, [26](#)
 - CMAAttributes.cpp, [28](#)
 - CMBroadcasts.cpp, [35](#)
 - CMMMethodCalls.cpp, [36](#)
 - PFCComplex.cpp, [41](#)
 - PFFPrimitive.cpp, [42](#)
- testAddress
 - AFPolymorph.cpp, [24](#)
 - AFSelective.cpp, [26](#)
 - CMAAttributes.cpp, [28](#)
 - CMAAttributeSubscription.cpp, [33](#)
 - CMBroadcasts.cpp, [35](#)
 - CMMMethodCalls.cpp, [36](#)
 - DTAdvanced.cpp, [37](#)
 - DTCombined.cpp, [38](#)
 - DTDerived.cpp, [39](#)
 - DTPrimitive.cpp, [40](#)
 - PFCComplex.cpp, [41](#)
 - PFFPrimitive.cpp, [42](#)
 - RTBuildProxiesAndStubs.cpp, [44](#)
 - StabilitySP.cpp, [47](#)
- testSubscription
 - CMAAttributeSubscription.cpp, [29](#)
- thirdPartyServiceld
 - THMainLoopIndependence.cpp, [48](#)
- usecPerSecond
 - PFCComplex.cpp, [41](#)
 - PFFPrimitive.cpp, [43](#)
- wt
 - CMAAttributeSubscription.cpp, [33](#)