3GPP TSG CN Plenary Meeting #17 **NP-020423**
4 - 6 September 2002, Biarritz, FRANCE

Source: **CN5 (OSA)**

Title: **Rel-4 CRs 29.198-03 OSA API Part 3: Framework**

Agenda item: **7.10**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-020423	29.198-03	045	-	Rel-4	Correction on use of NULL in Framework API	F	4.5.0	N5-020711	OSA1

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020711**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-03**	**CR**	**045**	⌘ **rev**	**-**	⌘ Current version:	**4.5.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network □ Core Network ☒

Title: ⌘	Correction on use of NULL in Framework API		
Source: ⌘	CN5		
Work item code: ⌘	OSA1	***Date:*** ⌘	12/07/2002
Category: ⌘	**F**	***Release:*** ⌘	*REL-4*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘	OMG IDL does not support NULL as a valid value for a data type; attempts to send a null value result in a marshalling exception and a gateway can never receive the call.
Summary of change: ⌘	Occurrences of the use of NULL as a valid Framework API data value have been replaced. An empty string value for serviceID is used to refer to the Framework in the Fault Management service, and an unspecified time interval is used to indicate that the time interval is at the discretion of the interface in question.
Consequences if not approved: ⌘	Failure to correct the API shall result in vendor specific interpretation and interoperability issues.

Clauses affected: ⌘	7.1.2.8; 7.3.3.1; 7.3.3.2; 7.4.3.4; 8.3.4.1; 8.3.4.2	
Other specs affected: ⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments: ⌘		

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

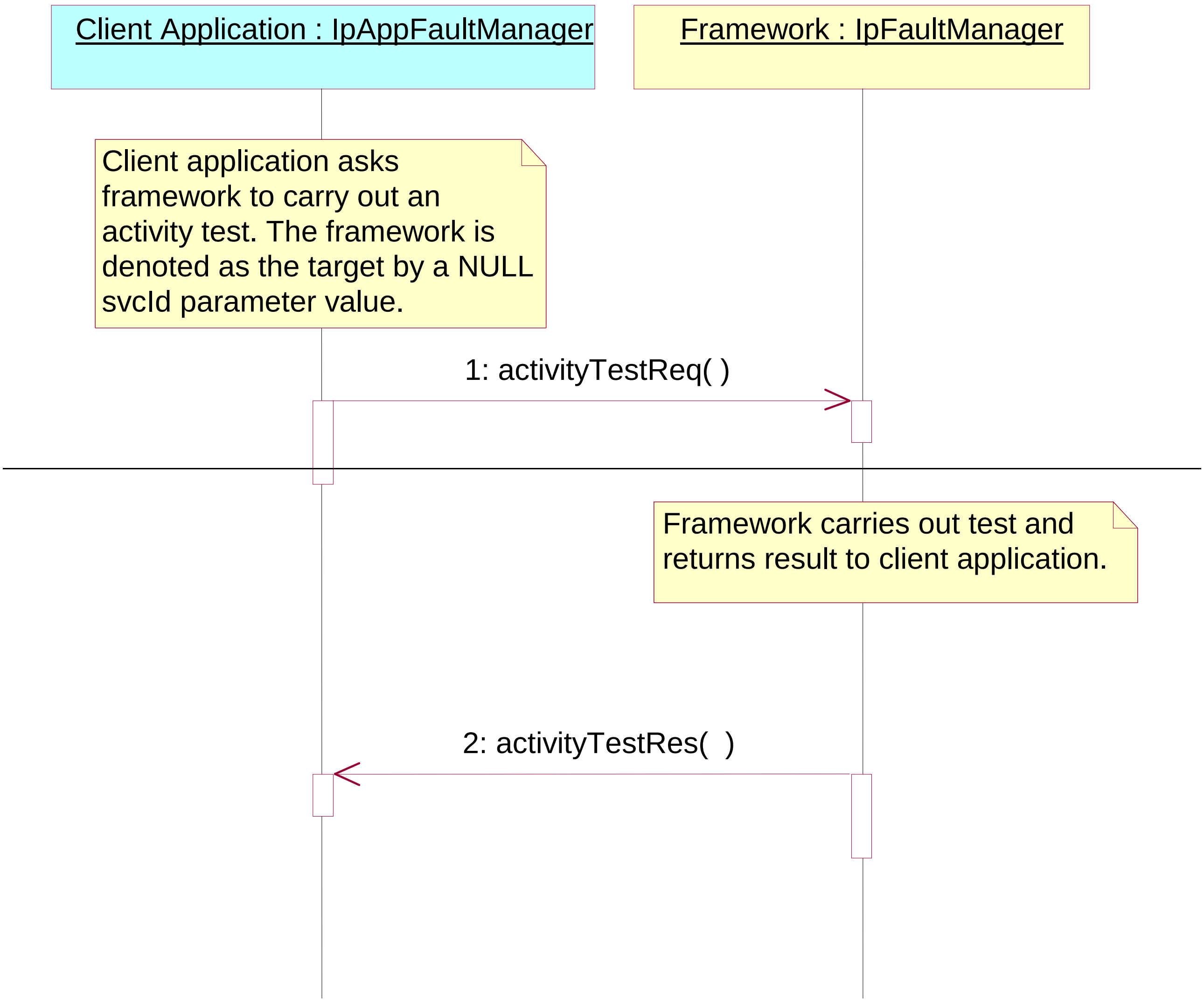
1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

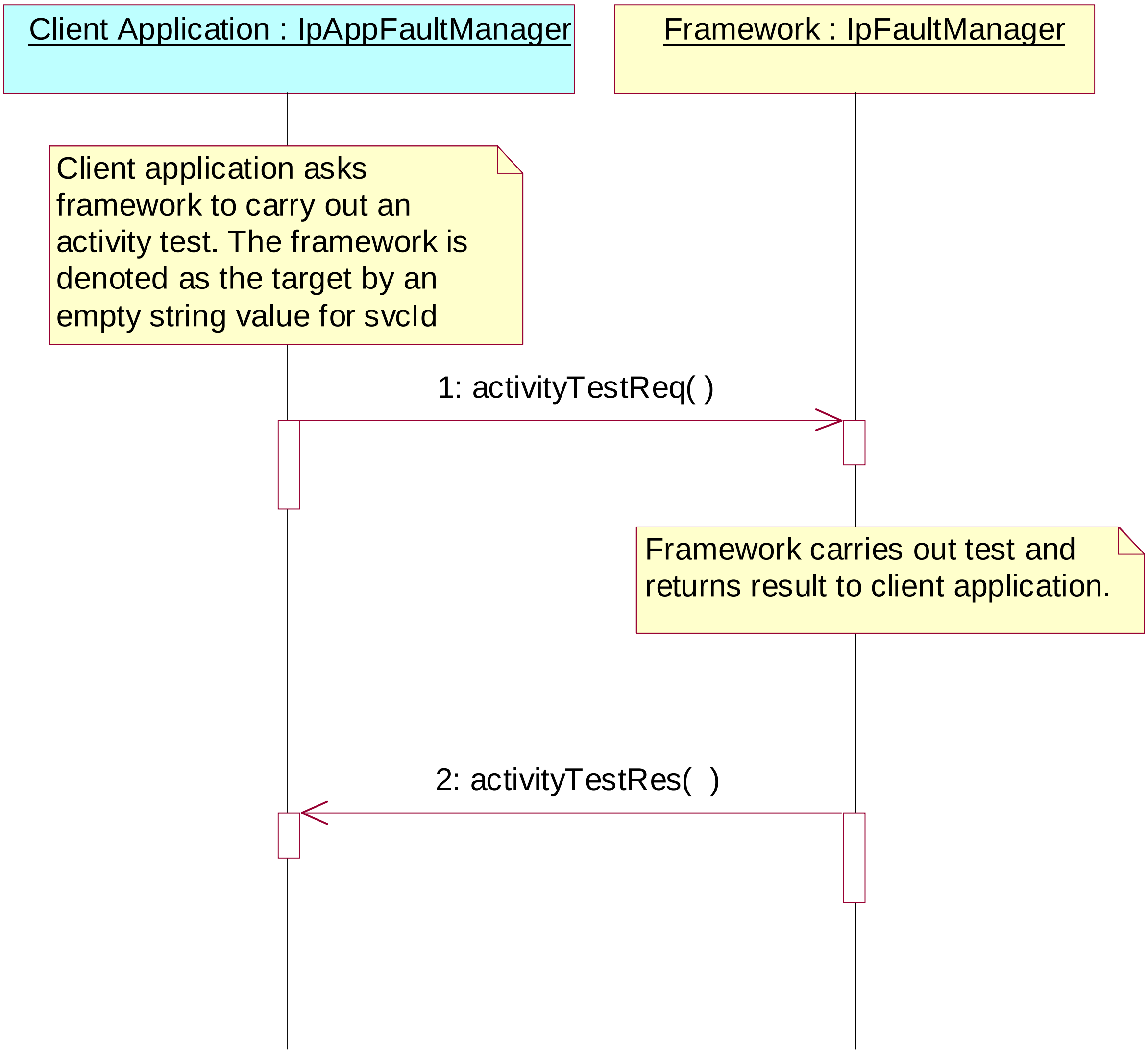
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

************************************ START OF FIRST CHANGE **************************************

7.1.2.8 Fault Management: Application requests a Framework activity test


Client Application : IpAppFaultManager
Framework : IpFaultManager
Client application asks framework to carry out an activity test. The framework is denoted as the target by a NULL svcId parameter value.
1: activityTestReq( )
Framework carries out test and returns result to client application.
2: activityTestRes( )


Client Application : IpAppFaultManager
Framework : IpFaultManager
Client application asks framework to carry out an activity test. The framework is denoted as the target by an empty string value for svcId
1: activityTestReq( )
Framework carries out test and returns result to client application.
2: activityTestRes( )

1: The client application asks the framework to do an activity test. The client identifies that it would like the activity test done for the framework, rather then a service, by supplying an empty string ~~NULL~~ value for the svcId parameter.

2: The framework does the requested activity test and sends the result to the client application.

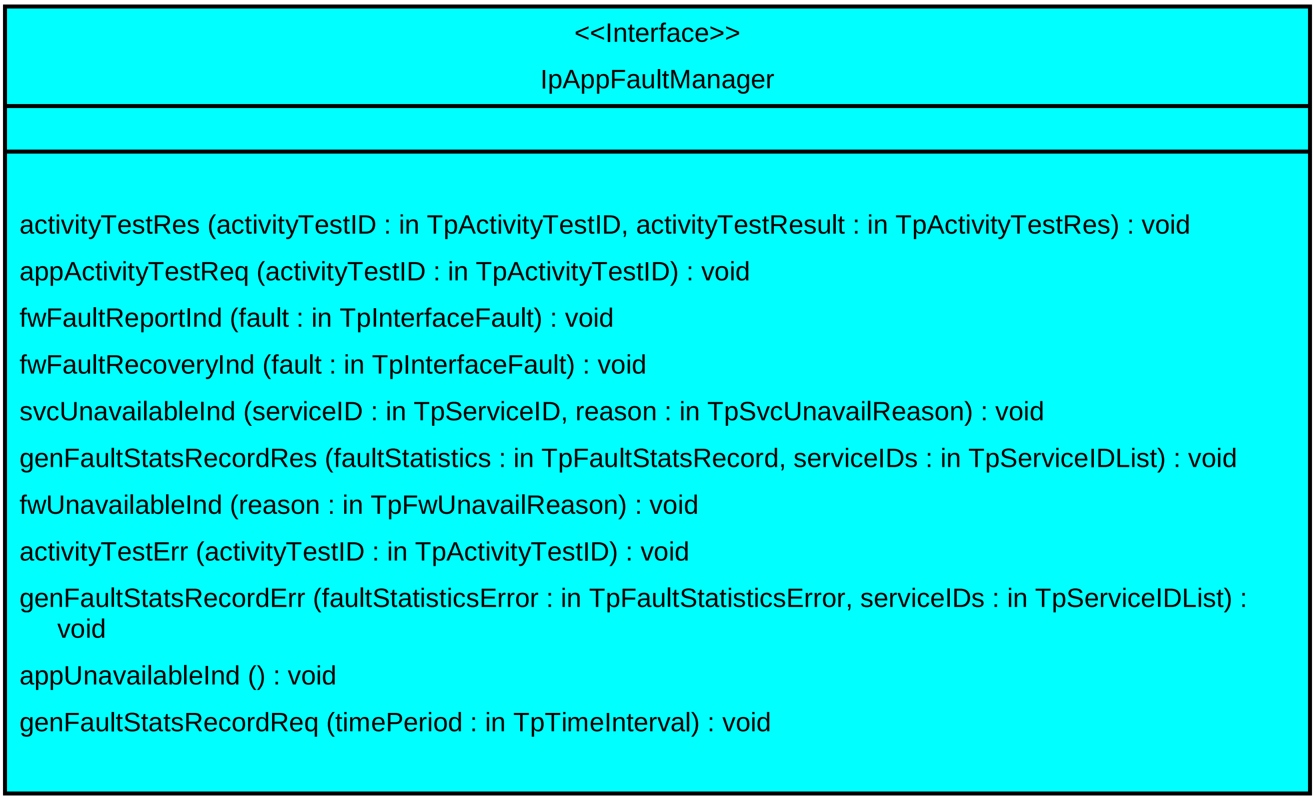
*********************************** END OF FIRST CHANGE *************************************

********************************** START OF SECOND CHANGE **********************************

7.3.3.1 Interface Class IpAppFaultManager

Inherits from: IpInterface.

This interface is used to inform the application of events that affect the integrity of the Framework, Service or Client Application. The Fault Management Framework will invoke methods on the Fault Management Application Interface that is specified when the client application obtains the Fault Management interface: i.e. by use of the obtainInterfaceWithCallback operation on the IpAccess interface



<<Interface>>

IpAppFaultManager

activityTestRes (activityTestID : in TpActivityTestID, activityTestResult : in TpActivityTestRes) : void

appActivityTestReq (activityTestID : in TpActivityTestID) : void

fwFaultReportInd (fault : in TpInterfaceFault) : void

fwFaultRecoveryInd (fault : in TpInterfaceFault) : void

svcUnavailableInd (serviceID : in TpServiceID, reason : in TpSvcUnavailReason) : void

genFaultStatsRecordRes (faultStatistics : in TpFaultStatsRecord, serviceIDs : in TpServiceIDList) : void

fwUnavailableInd (reason : in TpFwUnavailReason) : void

activityTestErr (activityTestID : in TpActivityTestID) : void

genFaultStatsRecordErr (faultStatisticsError : in TpFaultStatisticsError, serviceIDs : in TpServiceIDList) : void

appUnavailableInd () : void

genFaultStatsRecordReq (timePeriod : in TpTimeInterval) : void

Method

activityTestRes()

The framework uses this method to return the result of a client application-requested activity test.

Parameters

activityTestID : in TpActivityTestID

Used by the client application to correlate this response (when it arrives) with the original request.

activityTestResult : in TpActivityTestRes

The result of the activity test.

Method

appActivityTestReq()

The framework invokes this method to test that the client application is operational. On receipt of this request, the application must carry out a test on itself, to check that it is operating correctly. The application reports the test result by invoking the appActivityTestRes method on the IpFaultManager interface.

Parameters

activityTestID : in TpActivityTestID

The identifier provided by the framework to correlate the response (when it arrives) with this request.

Method

fwFaultReportInd()

The framework invokes this method to notify the client application of a failure within the framework. The client application must not continue to use the framework until it has recovered (as indicated by a fwFaultRecoveryInd).

Parameters

fault : in TpInterfaceFault

Specifies the fault that has been detected by the framework.

Method

fwFaultRecoveryInd()

The framework invokes this method to notify the client application that a previously reported fault has been rectified. The application may then resume using the framework.

Parameters

fault : in TpInterfaceFault

Specifies the fault from which the framework has recovered.

Method

svcUnavailableInd()

The framework invokes this method to inform the client application that it can no longer use its instance of the indicated service. On receipt of this request, the client application must act to reset its use of the specified service (using the normal mechanisms, such as the discovery and authentication interfaces, to stop use of this service instance and begin use of a different service instance).

Parameters

serviceID : in TpServiceID

Identifies the affected service.

reason : in TpSvcUnavailReason

Identifies the reason why the service is no longer available

Method

genFaultStatsRecordRes()

This method is used by the framework to provide fault statistics to a client application in response to a genFaultStatsRecordReq method invocation on the IpFaultManager interface.

Parameters

faultStatistics : in TpFaultStatsRecord

The fault statistics record.

serviceIDs : in TpServiceIDList

Specifies the framework or services that are included in the general fault statistics record. If the serviceIDs parameter is an empty list, then the fault statistics are for the framework.

Method

fwUnavailableInd()

The framework invokes this method to inform the client application that it is no longer available.

Parameters

reason : in TpFwUnavailReason

Identifies the reason why the framework is no longer available

Method

activityTestErr()

The framework uses this method to indicate that an error occurred during an application-initiated activity test.

Parameters

activityTestID : in TpActivityTestID

Used by the application to correlate this response (when it arrives) with the original request.

Method

genFaultStatsRecordErr()

This method is used by the framework to indicate an error fulfilling the request to provide fault statistics, in response to a genFaultStatsRecordReq method invocation on the IpFaultManager interface.

Parameters

faultStatisticsError : in TpFaultStatisticsError

The fault statistics error.

serviceIDs : in TpServiceIDList

Specifies the framework or services that were included in the general fault statistics record request. If the serviceIDs parameter is an empty list, then the fault statistics were requested for the framework.

Method

appUnavailableInd()

The framework invokes this method to indicate to the application that the service instance has detected that it is not responding. On receipt of this indication, the application must end its current session with the service instance.

Parameters

No Parameters were identified for this method

Method

genFaultStatsRecordReq()

This method is used by the framework to solicit fault statistics from the client application, for example when the framework was asked for these statistics by a service instance by using the genFaultStatsRecordReq operation on the IpFwFaultManager interface. On receipt of this request, the client application must produce a fault statistics record, for the application during the specified time interval, which is returned to the framework using the genFaultStatsRecordRes operation on the IpFaultManager interface.

Parameters

timePeriod : in TpTimeInterval

The period over which the fault statistics are to be generated. ~~A null~~Supplying both a start time and stop t ime as empty strings ~~value~~ leaves ~~this~~ the time period to the discretion of the client application.

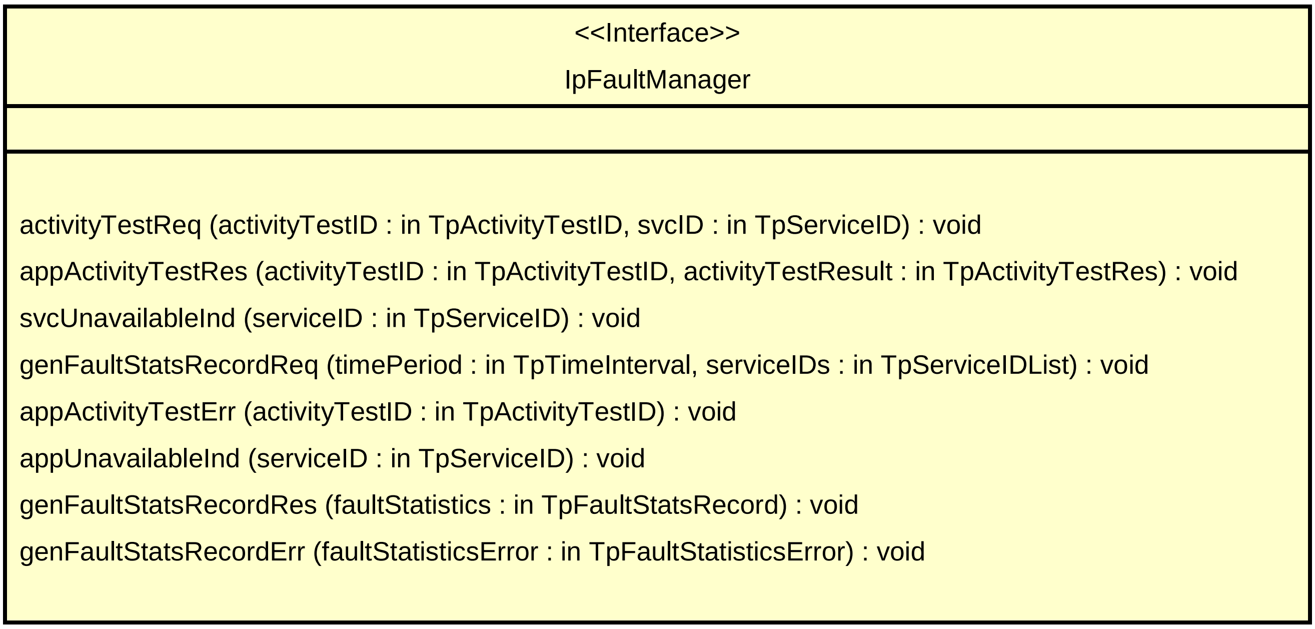
******************************* END OF SECOND CHANGE *******************************

******************************* START OF THIRD CHANGE *******************************

7.3.3.2 Interface Class IpFaultManager

Inherits from: IpInterface.

This interface is used by the application to inform the framework of events that affect the integrity of the framework and services, and to request information about the integrity of the system. The fault manager operations do not exchange callback interfaces as it is assumed that the client application supplies its Fault Management callback interface at the time it obtains the Framework's Fault Management interface, by use of the obtainInterfaceWithCallback operation on the IpAccess interface.



<<Interface>>

IpFaultManager

activityTestReq (activityTestID : in TpActivityTestID, svcID : in TpServiceID) : void

appActivityTestRes (activityTestID : in TpActivityTestID, activityTestResult : in TpActivityTestRes) : void

svcUnavailableInd (serviceID : in TpServiceID) : void

genFaultStatsRecordReq (timePeriod : in TpTimeInterval, serviceIDs : in TpServiceIDList) : void

appActivityTestErr (activityTestID : in TpActivityTestID) : void

appUnavailableInd (serviceID : in TpServiceID) : void

genFaultStatsRecordRes (faultStatistics : in TpFaultStatsRecord) : void

genFaultStatsRecordErr (faultStatisticsError : in TpFaultStatisticsError) : void

Method

activityTestReq()

The application invokes this method to test that the framework or its instance of a service is operational. On receipt of this request, the framework must carry out a test on itself or on the client's instance of the specified service, to check that it is operating correctly. The framework reports the test result by invoking the activityTestRes method on the IpAppFaultManager interface. If the application does not have access to a service instance with the specified serviceID, the P_UNAUTHORISED_PARAMETER_VALUE exception shall be thrown. The extraInformation field of the exception shall contain the corresponding serviceID.

For security reasons the client application has access to the service ID rather than the service instance ID. However, as there is a one to one relationship between the client application and a service, i.e. there is only one service instance of the specified service per client application, it is the obligation of the framework to determine the service instance ID from the service ID.

Parameters

activityTestID : in TpActivityTestID

The identifier provided by the client application to correlate the response (when it arrives) with this request.

svcID : in TpServiceID

Identifies either the framework or a service for testing. The framework is designated by an empty string ~~null value~~.

Raises

TpCommonExceptions,P_INVALID_SERVICE_ID, P_UNAUTHORISED_PARAMETER_VALUE

Method

appActivityTestRes()

The client application uses this method to return the result of a framework-requested activity test.

Parameters

activityTestID : in TpActivityTestID

Used by the framework to correlate this response (when it arrives) with the original request.

activityTestResult : in TpActivityTestRes

The result of the activity test.

Raises

TpCommonExceptions,P_INVALID_SERVICE_ID,P_INVALID_ACTIVITY_TEST_ID

Method

svcUnavailableInd()

This method is used by the client application to inform the framework that it can no longer use its instance of the indicated service (either due to a failure in the client application or in the service instance itself). On receipt of this request, the framework should take the appropriate corrective action. The framework assumes that the session between this client application and service instance is to be closed and updates its own records appropriately as well as attempting to inform the service instance and/or its administrator. Attempts by the client application to continue using this session should be rejected. If the application does not have access to a service instance with the specified serviceID, the P_UNAUTHORISED_PARAMETER_VALUE exception shall be thrown. The extraInformation field of the exception shall contain the corresponding serviceID.

Parameters

serviceID : in TpServiceID

Identifies the service that the application can no longer use.

Raises

TpCommonExceptions ,P_INVALID_SERVICE_ID, P_UNAUTHORISED_PARAMETER_VALUE

Method **genFaultStatsRecordReq()**

This method is used by the application to solicit fault statistics from the framework. On receipt of this request the framework must produce a fault statistics record, for either the framework or for the client's instances of the specified services during the specified time interval, which is returned to the client application using the genFaultStatsRecordRes operation on the IpAppFaultManager interface. If the application does not have access to a service instance with the specified serviceID, the P_UNAUTHORISED_PARAMETER_VALUE exception shall be thrown. The extraInformation field of the exception shall contain the corresponding serviceID.

Parameters

timePeriod : in TpTimeInterval

The period over which the fault statistics are to be generated. ~~A null~~Supplying both a start time and stop time as empty strings~~ value~~ leaves ~~this ~~the time period to the discretion of the framework.

serviceIDs : in TpServiceIDList

Specifies either the framework or services to be included in the general fault statistics record. If this parameter is not an empty list, the fault statistics records of the client's instances of the specified services are returned, otherwise the fault statistics record of the framework is returned.

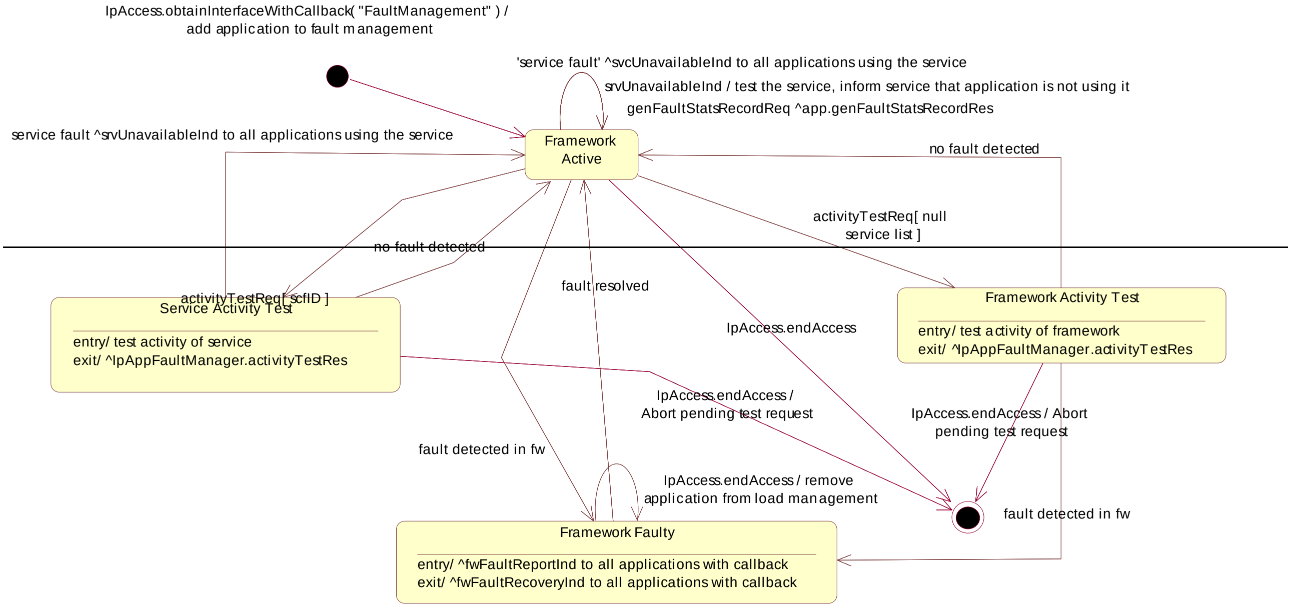
Raises

TpCommonExceptions, P_INVALID_SERVICE_ID, P_UNAUTHORISED_PARAMETER_VALUE

*********************************** END OF THIRD CHANGE ************************************

*********************************** START OF FOURTH CHANGE ************************************

7.4.3.4 State Transition Diagrams for IpFaultManager


IpAccess.obtainInterfaceWithCallback("FaultManagement") /
add application to fault management
'service fault' ^svcUnavailableInd to all applications using the service
srvUnavailableInd / test the service, inform service that application is not using it
genFaultStatsRecordReq ^app.genFaultStatsRecordRes
service fault ^srvUnavailableInd to all applications using the service
Framework
Active
no fault detected
activityTestReq[null
service list]
no fault detected
activityTestReq[scfID]
Service Activity Test
entry/ test activity of service
exit/ ^IpAppFaultManager.activityTestRes
fault resolved
IpAccess.endAccess
Framework Activity Test
entry/ test activity of framework
exit/ ^IpAppFaultManager.activityTestRes
IpAccess.endAccess /
Abort pending test request
IpAccess.endAccess / Abort
pending test request
fault detected in fw
IpAccess.endAccess / remove
application from load management
fault detected in fw
Framework Faulty
entry/ ^fwFaultReportInd to all applications with callback
exit/ ^fwFaultRecoveryInd to all applications with callback

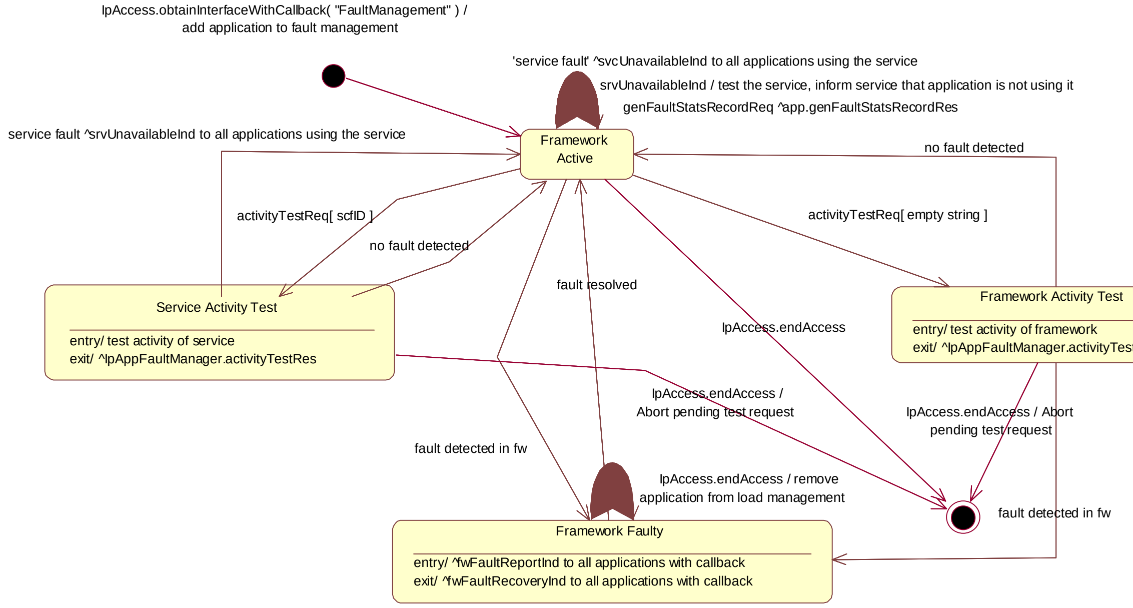

IpAccess.obtainInterfaceWithCallback("FaultManagement") /
add application to fault management
'service fault' ^svcUnavailableInd to all applications using the service
srvUnavailableInd / test the service, inform service that application is not using it
genFaultStatsRecordReq ^app.genFaultStatsRecordRes
service fault ^srvUnavailableInd to all applications using the service
Framework
Active
no fault detected
activityTestReq[scfID]
activityTestReq[empty string]
no fault detected
fault resolved
Service Activity Test
entry/ test activity of service
exit/ ^IpAppFaultManager.activityTestRes
IpAccess.endAccess
Framework Activity Test
entry/ test activity of framework
exit/ ^IpAppFaultManager.activityTestRes
IpAccess.endAccess /
Abort pending test request
IpAccess.endAccess / Abort
pending test request
fault detected in fw
IpAccess.endAccess / remove
application from load management
fault detected in fw
Framework Faulty
entry/ ^fwFaultReportInd to all applications with callback
exit/ ^fwFaultRecoveryInd to all applications with callback

Figure : State Transition Diagram for IpFaultManager

********************************** END OF FOURTH CHANGE **********************************

********************************** START OF FIFTH CHANGE **********************************

8.3.4.1 Interface Class IpFwFaultManager

Inherits from: IpInterface.

This interface is used by the service instance to inform the framework of events which affect the integrity of the API, and request fault management status information from the framework. The fault manager operations do not exchange callback interfaces as it is assumed that the service instance has supplied its Fault Management callback interface at the time it obtains the Framework's Fault Management interface, by use of the obtainInterfaceWithCallback operation on the IpAccess interface.

<<Interface>> IpFwFaultManager
activityTestReq (activityTestID : in TpActivityTestID, testSubject : in TpSubjectType) : void svcActivityTestRes (activityTestID : in TpActivityTestID, activityTestResult : in TpActivityTestRes) : void appUnavailableInd () : void genFaultStatsRecordReq (timePeriod : in TpTimeInterval, recordSubject : in TpSubjectType) : void svcUnavailableInd (reason : in TpSvcUnavailReason) : void svcActivityTestErr (activityTestID : in TpActivityTestID) : void genFaultStatsRecordRes (faultStatistics : in TpFaultStatsRecord, serviceIDs : in TpServiceIDList) : void genFaultStatsRecordErr (faultStatisticsError : in TpFaultStatisticsError, serviceIDs : in TpServiceIDList) : void

Method
activityTestReq()

The service instance invokes this method to test that the framework or the client application is operational. On receipt of this request, the framework must carry out a test on itself or on the application, to check that it is operating correctly. The framework reports the test result by invoking the activityTestRes method on the IpSvcFaultManager interface.

Parameters

activityTestID : in TpActivityTestID

The identifier provided by the service instance to correlate the response (when it arrives) with this request.

testSubject : in TpSubjectType

Identifies the subject for testing (framework or client application).

Raises

TpCommonExceptions

Method

svcActivityTestRes()

The service instance uses this method to return the result of a framework-requested activity test.

Parameters

activityTestID : in TpActivityTestID

Used by the framework to correlate this response (when it arrives) with the original request.

activityTestResult : in TpActivityTestRes

The result of the activity test.

Raises

TpCommonExceptions,P_INVALID_ACTIVITY_TEST_ID

Method

appUnavailableInd()

This method is used by the service instance to inform the framework that the client application is not responding. On receipt of this indication, the framework must act to inform the client application that it should cease use of this service instance.

Parameters

No Parameters were identified for this method

Raises

TpCommonExceptions

Method

genFaultStatsRecordReq()

This method is used by the service instance to solicit fault statistics from the framework. On receipt of this request, the framework must produce a fault statistics record, for the framework or for the application during the specified time interval, which is returned to the service instance using the genFaultStatsRecordRes operation on the IpSvcFaultManager interface.

Parameters

timePeriod : in TpTimeInterval

The period over which the fault statistics are to be generated. ~~A null~~Supplying both a start time and stop time as empty strings~~ value~~ leaves ~~this ~~the time period to the discretion of the framework.

recordSubject : in TpSubjectType

Specifies the subject to be included in the general fault statistics record (framework or application).

Raises

TpCommonExceptions

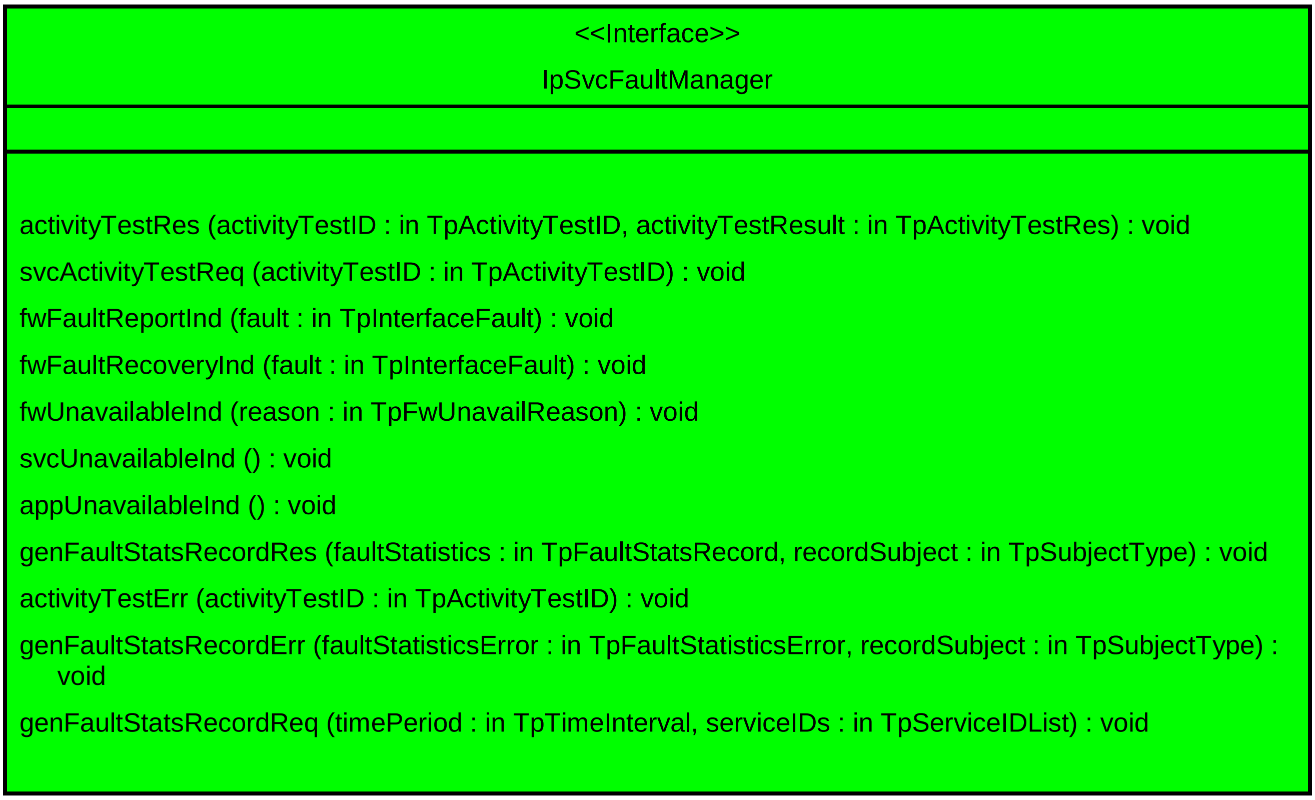
*********************************** END OF FIFTH CHANGE ************************************

*********************************** START OF SIXTH CHANGE ***********************************

8.3.4.2 Interface Class IpSvcFaultManager

Inherits from: IpInterface.

This interface is used to inform the service instance of events that affect the integrity of the Framework, Service or Client Application. The Framework will invoke methods on the Fault Management Service Interface that is specified when the service instance obtains the Fault Management Framework interface: i.e. by use of the obtainInterfaceWithCallback operation on the IpAccess interface



<<Interface>>

IpSvcFaultManager

activityTestRes (activityTestID : in TpActivityTestID, activityTestResult : in TpActivityTestRes) : void

svcActivityTestReq (activityTestID : in TpActivityTestID) : void

fwFaultReportInd (fault : in TpInterfaceFault) : void

fwFaultRecoveryInd (fault : in TpInterfaceFault) : void

fwUnavailableInd (reason : in TpFwUnavailReason) : void

svcUnavailableInd () : void

appUnavailableInd () : void

genFaultStatsRecordRes (faultStatistics : in TpFaultStatsRecord, recordSubject : in TpSubjectType) : void

activityTestErr (activityTestID : in TpActivityTestID) : void

genFaultStatsRecordErr (faultStatisticsError : in TpFaultStatisticsError, recordSubject : in TpSubjectType) : void

genFaultStatsRecordReq (timePeriod : in TpTimeInterval, serviceIDs : in TpServiceIDList) : void

Method
activityTestRes()

The framework uses this method to return the result of a service-requested activity test.

Parameters

activityTestID : in TpActivityTestID

Used by the service to correlate this response (when it arrives) with the original request.

activityTestResult : in TpActivityTestRes

The result of the activity test.

Raises

TpCommonExceptions,P_INVALID_ACTIVITY_TEST_ID

Method

svcActivityTestReq()

The framework invokes this method to test that the service instance is operational. On receipt of this request, the service instance must carry out a test on itself, to check that it is operating correctly. The service instance reports the test result by invoking the svcActivityTestRes method on the IpFwFaultManager interface.

Parameters

activityTestID : in TpActivityTestID

The identifier provided by the framework to correlate the response (when it arrives) with this request.

Raises

TpCommonExceptions

Method

fwFaultReportInd()

The framework invokes this method to notify the service instance of a failure within the framework. The service instance must not continue to use the framework until it has recovered (as indicated by a fwFaultRecoveryInd).

Parameters

fault : in TpInterfaceFault

Specifies the fault that has been detected by the framework.

Raises

TpCommonExceptions

Method

fwFaultRecoveryInd()

The framework invokes this method to notify the service instance that a previously reported fault has been rectified. The service instance may then resume using the framework.

Parameters

`fault : in TpInterfaceFault`

Specifies the fault from which the framework has recovered.

Raises

`TpCommonExceptions`

Method

`fwUnavailableInd()`

The framework invokes this method to inform the service instance that it is no longer available.

Parameters

`reason : in TpFwUnavailReason`

Identifies the reason why the framework is no longer available

Raises

`TpCommonExceptions`

Method

`svcUnavailableInd()`

The framework invokes this method to inform the service instance that the client application has reported that it can no longer use the service instance (either due to a failure in the client application or in the service instance itself). The service should assume that the client application is leaving the service session and the service should act accordingly to terminate the session from its own end too.

Parameters

No Parameters were identified for this method

Raises

`TpCommonExceptions`

Method

`appUnavailableInd()`

The framework invokes this method to inform the service instance that the client application is ceasing its current use of the service. This may be a result of the application reporting a failure. Alternatively, the framework may have detected that the application has failed: e.g. non-response from an activity test, failure to return heartbeats.

Parameters

No Parameters were identified for this method

Raises

TpCommonExceptions

Method

genFaultStatsRecordRes()

This method is used by the framework to provide fault statistics to a service instance in response to a genFaultStatsRecordReq method invocation on the IpFwFaultManager interface.

Parameters

faultStatistics : in TpFaultStatsRecord

The fault statistics record.

recordSubject : in TpSubjectType

Specifies the entity (framework or application) whose fault statistics record has been provided.

Raises

TpCommonExceptions

Method

activityTestErr()

The framework uses this method to indicate that an error occurred during a service-requested activity test.

Parameters

activityTestID : in TpActivityTestID

Used by the service instance to correlate this response (when it arrives) with the original request.

Raises

TpCommonExceptions, P_INVALID_ACTIVITY_TEST_ID

Method

genFaultStatsRecordErr()

This method is used by the framework to indicate an error fulfilling the request to provide fault statistics, in response to a genFaultStatsRecordReq method invocation on the IpFwFaultManager interface.

Parameters

faultStatisticsError : in TpFaultStatisticsError

The fault statistics error.

recordSubject : in TpSubjectType

Specifies the entity (framework or application) whose fault statistics record was requested.

Raises

TpCommonExceptions

Method

genFaultStatsRecordReq()

This method is used by the framework to solicit fault statistics from the service, for example when the framework was asked for these statistics by the client application using the genFaultStatsRecordReq operation on the IpFaultManager interface. On receipt of this request the service must produce a fault statistics record, for either the framework or for the client's instances of the specified services during the specified time interval, which is returned to the framework using the genFaultStatsRecordRes operation on the IpFwFaultManager interface. If the framework does not have access to a service instance with the specified serviceID, the P_UNAUTHORISED_PARAMETER_VALUE exception shall be thrown. The extraInformation field of the exception shall contain the corresponding serviceID.

Parameters

timePeriod : in TpTimeInterval

The period over which the fault statistics are to be generated. ~~A null~~Supplying both a start time and stop time as empty strings~~ value~~ leaves ~~this ~~the time period to the discretion of the service.

serviceIDs : in TpServiceIDList

Specifies the services to be included in the general fault statistics record. This parameter is not allowed to be an empty list.

Raises

TpCommonExceptions, P_INVALID_SERVICE_ID, P_UNAUTHORISED_PARAMETER_VALUE

********************************* END OF SIXTH CHANGE **********************************