

ETSI TS 132 323 V9.0.0 (2010-02)

Technical Specification

**Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Telecommunication management;
Test management Integration Reference Point (IRP);
Common Object Request Broker Architecture (CORBA)
Solution Set (SS)
(3GPP TS 32.323 version 9.0.0 Release 9)**



Reference

RTS/TSGS-0532323v900

Keywords

GSM, LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2010.
All rights reserved.

DECTTM, **PLUGTESTS**TM, **UMTS**TM, **TIPHON**TM, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

LTETM is a Trade Mark of ETSI currently being registered

for the benefit of its Members and of the 3GPP Organizational Partners.

GSM[®] and the GSM logo are Trade Marks registered and owned by the GSM Association.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Contents

Intellectual Property Rights	2
Foreword.....	2
Foreword.....	4
Introduction	4
1 Scope	5
2 References	5
3 Definitions and abbreviations.....	6
3.1 Definitions	6
3.2 Abbreviations	6
3.3 IRP document version number string	6
4 Architectural features	7
4.1 Notification Services	7
4.2 Push and Pull Style.....	7
4.3 Support multiple notifications in one push operation.....	7
5 Mapping	8
5.1 Operation and Notification mapping	8
5.2 Operation parameter mapping	8
5.3 Notification parameter mapping.....	10
6 TestManagementIRPNotification Interface.....	12
6.1 Method push (M).....	12
Annex A (normative): IDL specifications	13
A.1 IDL specification (file name "TestManagementIRPConstDefs.idl")	13
A.2 IDL specification (file name "TestManagementIRPSystem.idl")	15
A.3 IDL specification (file name "TestManagementIRPNotifications.idl")	17
Annex B (informative): Change history	19
History	20

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

- 32.321: "Test management Integration Reference Point (IRP); Requirements"
- 32.322: "Test management Integration Reference Point (IRP): Information Service (IS)"
- 32.323: "Test management Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)"**
- 32.325: "Test management Integration Reference Point (IRP): eXtensible Markup Language (XML) definitions"

A 3G telecommunication network is composed of a multitude of different Network Elements (NE). For a successful operation of the network the operator must be provided with mechanisms allowing him to manage the network. These management activities can be grouped into several areas: configuration management, fault management, performance management, accounting management and security management.

A management function assisting in different high level management areas such as fault management and performance management is test management. The purpose of testing is to get information about the functionality and performance of the 3G managed network subject to the test.

The present document is part of a TS-family defining defining the Telecommunication Management (TM) of 3G systems. The TM principles are described in 3GPP TS 32.101 [5]. The TM architecture is described in 3GPP TS 32.102 [6]. The other specifications define the interface (Itf-N) between the managing system (manager), which is in general the Network Manager (NM) and the managed system (agent), which is either an Element Manager (EM) or the managed NE itself. The Itf-N is composed of a number of Integration Reference Points (IRPs) defining the information in the agent that is visible for the manager, the operations that the manager may perform on this information and the notifications that are sent from the agent to the manager. One of these IRPs is the Test IRP.

Each IRP is specified by the requirements part, the Information Service part, the CORBA SS and the CMIP SS.

1 Scope

The present document specifies the CORBA SS for the IRP whose semantics is specified in Test Management IRP IS (3GPP TS 32.322 [6]).

Clause 1 to 3 provides background information. Clause 4 provides key architectural features supporting the SS. Clause 5 defines the mapping of operations, notification, parameters and attributes defined in IS to their SS equivalents. Clause 6 describes the notification interface containing the push method. Annex A contains the IDL specification.

This Solution Set specification is related to 3GPP TS 32.322 V9.0.X.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] Object Management Group 98 (November 1998): "*Notification Service: Joint Revised Submission OMG TC Document telecom/98-11-01*". <http://www.omg.org/technology/documents/>
- [2] OMG CORBA Services (November 1996): "Common Object Services Specification" (clause 4 contains the Event Service specification). <http://www.omg.org/technology/documents/>
- [3] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [4] 3GPP TS 32.302: "Telecommunication management; Configuration Management; Notification Integration Reference Point; Information Service version 1".
- [5] 3GPP TS 32.303: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".
- [6] 3GPP TS 32.322: "Telecommunication management; Test management Integration Reference Point (IRP): Information Service (IS)".
- [7] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".
- [8] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management; Requirements".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions defined in 3GPP TS 32.322 [6] apply.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CORBA	Common Object Request Broker Architecture
EM	Element Manager
GSM	Global System for Mobile communications
IDL	Interface Definition Language
IRP	Integration Reference Point
IOC	Information Object Class
IS	Information Service
NE	Network Element
NM	Network Manager
OMG	Object Management Group
QoS	Quality of Service
SS	Solution Set

3.3 IRP document version number string

The IRP document version number (sometimes called "IRP version" or "version number") string is used to identify the present document. The definition of "IRP document version number string" in 3GPP TS 32.311 [8] provides the rule to derive such a string.

This string is returned in `get_test_management_IRP_versions` method and is carried in the first field of the notification header of all notifications related to Test Management IRP. This string is also returned in `get_notification_categories` method of the Notification IRPAgent, in case that IRPAgent is responsible for emitting notifications related to Test Management IRP.

4 Architectural features

The overall architectural feature of Test Management IRP is specified in 3GPP TS 32.322 [6]. The present document specifies features that are specific to the CORBA SS.

4.1 Notification Services

In implementations of CORBA SS, IRPAgent conveys Test Management information to IRPManager via OMG Notification Service (OMG Notification Service [1]).

OMG Event Service [2] provides event routing and distribution capabilities. OMG Notification Service provides, in addition to Event Service, event filtering and Quality of Service (QoS) as well.

A necessary and sufficient sub set of OMG Notification Services shall be used to support `TestManagementIRPNotifications` notifications as specified in 3GPP TS 32.322 [6].

4.2 Push and Pull Style

OMG Notification Service defines two styles of interaction. One is called push style. In this style, IRPAgent pushes notifications to IRPManager as soon as they are available. The other is called pull style. In this style, IRPAgent keeps the notifications till IRPManager requests for them.

The Notification CORBA SS in 3GPP TS 32.303 [5] specifies that support of Push style is Mandatory (M) and that support of Pull style is Optional (O).

4.3 Support multiple notifications in one `push` operation

For efficiency reasons, IRPAgent may send multiple notifications using one single `push` operation. To pack multiple notifications into one `push` operation, IRPAgent may wait and not invoke the `push` operation as soon as notifications are available. To avoid IRPAgent to wait for an extended period of time that is objectionable to IRPManager, IRPAgent shall implement an IRPAgent wide timer configurable by administrator. On expiration of this timer, IRPAgent shall invoke `push` if there is at least one notification to be conveyed to IRPManager. This timer is re-started after each `push` invocation.

5 Mapping

5.1 Operation and Notification mapping

The Test Management IRP IS in 3GPP TS 32.322 [6] defines semantics of operation and notification visible across the Test Management IRP. Table 1 indicates mapping of these operations and notifications to their equivalents defined in the present SS.

Table 1: Mapping from IS Operations and Notification to SS equivalents

IS Operations/ notification TS 32.322 [6]	SS Method	Qualifier
initiateTests	initiate_tests	M
terminateTests	terminate_tests	M
monitorTest	monitor_test	M
getIRPVersion	get_test_management_IRP_versions	M
getOperationProfile (see note)	get_test_management_IRP_operation_profile	O
getNotificationProfile (see note)	get_test_management_IRP_notification_profile	O
notifyTestResult	push_structured_event (See subclause 6.1)	M
NOTE: This operation is of ManagedGenericIRP IOC specified in 3GPP TS 32.312 [7]. The TestManagementIRP IOC of [6] inherits from it.		

5.2 Operation parameter mapping

The Test Management IRP IS in 3GPP TS 32.322 [6] defines semantics of parameters carried in operations across the Test Management IRP. Table2 gives the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table 2: Mapping from IS initiateTests parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
testInvocationInitiator	TestManagementIRPConstDefs::TestInvocationInitiator test_invocation_initiator	M
toBeInitiatedTests	TestManagementIRPConstDefs::ToBeInitiatedTestSeq to_be_initiated_test_seq	M
response	TestManagementIRPConstDefs::InitiateTestsResponse Exceptions: InitiateTests, ManagedGenericRPSysyem::ParameterNotSupported, ManagedGenericRPSysyem::InvalidParameter	M

Table 3: Mapping from IS terminateTests parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
toBeTerminatedTests	TestManagementIRPConstDefs::ToBeTerminatedTestSeq to_be_terminated_test_seq	M
response	TestManagementIRPConstDefs::TerminateTestsResponse Exceptions: TerminateTests, ManagedGenericRPSysyem::OperationNotSupported, ManagedGenericRPSysyem::ParameterNotSupported, ManagedGenericRPSysyem::InvalidParameter	M

Table 4: Mapping from IS `monitorTest` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
toBeMonitoredTO	TestManagementIRPConstDefs::ToBeMonitoredTO to_be_monitored_TO	M
monitoredAttributeValues	TestManagementIRPConstDefs::TOAttributes tO_attributes	M
error	ManagedGenericIRPConstDefs::Signal Exceptions: MonitorTest, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter	M

5.3 Notification parameter mapping

The Test Management IRP IS in 3GPP TS 32.322 [6] defines semantics of parameters carried in notifications. The following table indicates the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [1]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [1], is:

```

Header
  Fixed Header
    domain_name
    type_name
    event_name
  Variable Header
Body
  filterable_body_fields
  remaining_body

```

Table 5 lists in the second column all OMG Structured Event attributes. The first column identifies notification parameters defined in 3GPP TS 32.322 [6], Test Management: Information Service (IS).

Table 5: Mapping for notifyTestResult

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS parameter.	domain_name	M	It carries the IRP document version number string. See subclause 3.3. It indicates the syntax and semantics of the Structured Event as defined by the present document.
notificationType	type_name	M	This is the NOTIFY_TM_TEST_RESULT of module of TestManagementIRPConstDefs.
There is no corresponding IS parameter.	event_name	M	It carries no information.
There is no corresponding IS parameter.	Variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in TS 32.303 [5] Notification IRP: CORBA SS.
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in TS 32.303 [5] Notification IRP: CORBA SS.
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS [5].
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in TS 32.303 [5] Notification IRP: CORBA SS.
testInvocationInitiator	One NV pair of filterable_body_fields	M	Name of NV pair is the TEST_INVOCATION_INITIATOR of module TestManagementIRPConstDefs. Value of NV pair is a string.

testOutcome	One NV pair of remaining_body	O	Name of NV pair is the TEST_OUTCOME of module TestManagementIRPCConstDefs. Value of NV pair is a enum {Inconclusive, Pass, Fail, TimeOut, PrematureTermination}
mORT	One NV pair of remaining_body	O	Name of NV pair is the MORT of module TestManagementIRPCConstDefs. Value of NV pair is a string.
proposedRepairActions	One NV pair of remaining_body	O	Name of NV pair is the PROPOSED_REPAIR_ACTIONS of module TestManagementIRPCConstDefs. Value of NV pair is a string.
additionalInformation	One NV pair of remaining_body	O	Name of NV pair is the ADDITIONAL_INFORMATION of module TestManagementIRPCConstDefs. Value of NV pair is a string.
fileReference	One NV pair of remaining_body	M (Note 1)	Name of NV pair is the FILE_REFERENCE of module TestManagementIRPCConstDefs. Value of NV pair is a string.
fileExpiryDate	One NV pair of remaining_body	M (Note 2)	Name of NV pair is the FILE_EXPIRY_DATE of module TestManagementIRPCConstDefs. Value of NV pair is a string.
<p>Note 1: It shall contain no information or be absent if there is no test result captured in a file. It shall contain information if the test results are captured in a file.</p> <p>Note 2: It shall contain no information or be absent if fileReference carries no information or absent. Otherwise, it shall contain a valid future date and time.</p>			

6 TestManagementIRPNotification Interface

OMG CORBA Notification push operation is used to realise the notification of TestManagementIRPNotifications. All the notifications in this interface are implemented using this `push_structured_event` method.

6.1 Method `push` (M)

```
module CosNotifyComm {  
    ...  
    Interface SequencePushConsumer : NotifyPublish {  
        void push_structured_events(  
            in CosNotification::EventBatch notifications)  
            raises( CosEventComm::Disconnected);  
        ...  
    }; // SequencePushConsumer  
    ...  
}; // CosNotifyComm
```

NOTE 1: The `push_structured_events` method takes an input parameter of type `EventBatch` as defined in the `OMG CosNotification` module (OMG Notification Service [1]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter will contain a sequence of Structured Events being delivered to `IRPManager` by `IRPAgent` to which it is connected.

NOTE 2: The maximum number of events that will be transmitted within a single invocation of this operation is controlled by `IRPAgent` wide configuration parameter.

NOTE 3: The amount of time the supplier (`IRPAgent`) of a sequence of Structured Events will accumulate individual events into the sequence before invoking this operation is controlled by `IRPAgent` wide configuration parameter as well.

NOTE 4: `IRPAgent` may push `EventBatch` with only one Structured Event.

Annex A (normative): IDL specifications

A.1 IDL specification (file name "TestManagementIRPConstDefs.idl")

```
// File: TestManagementIRPConstDefs.idl

#ifndef _TESTMANAGEMENTIRPCONSTDEFS_IDL_
#define _TESTMANAGEMENTIRPCONSTDEFS_IDL_

#include "CosNotification.idl"
#include "ManagedGenericIRPConstDefs.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: TestManagementIRPConstDefs
This module contains commonly used definitions for Test Management IRP
=====
*/
module TestManagementIRPConstDefs
{
    /*
    This defines the notification type of this Test Management
    IRP.
    */
    const string NOTIFY_TM_TEST_RESULT = "x1";

    /*
    This enum defines the test state
    */
    enum TestStateType {
        NotInitialized,
        Idle,
        Initializing,
        Testing,
        Terminating,
        Disabled
    };

    /*
    This enum defines the test outcome
    */
    enum TestOutcomeType {
        Inconclusive,
        Pass,
        Fail,
        TimeOut,
        PrematureTermination
    };

    /*
    This block defines notification attributes of this IRP.
    These attribute values should not clash with those used
    in Notification header (see IDL of Notification IRP).
    */
    interface AttributeNameValue
    {
        const string TEST_INVOCATION_INITIATOR = "f";
        const string TEST_INVOCATION_ID = "g";
        const string TEST_ACTUAL_START_TIME = "h";
        const string TEST_ACTUAL_STOP_TIME = "i";
        const string TEST_OUTCOME = "j";
        const string MORT = "k";
        const string PROPOSED_REPAIR_ACTIONS = "l";
        const string ADDITIONAL_INFORMATION = "m";
    };
};

```

```

const string FILE_REFERENCE = "n";
const string FILE_EXPIRY_DATE = "o";
const string TEST_SOURCE_ADDRESS = "p";
const string TEST_DESTINATION_ADDRESS = "q";
const string TEST_LOOPBACK_ADDRESS = "r";
const string TEST_PACKET_INFORMATION = "s";
};

typedef string TestInvocationInitiator;
typedef string ToBeMonitoredTO;

typedef CosNotification::PropertySeq NVPairs;

/*
Define a seq of to-be-initiated-test
*/
struct ToBeInitiatedTest
{
    unsigned long max_testing_state_duration;//seconds;0->no limit
    string toBeTestedMORT; //MORT DN
    string tOClass; //Tester object class
    string tODN; //Tester object DN
    NVPairs tONVPair; //Tester object attributes in NV pairs
};
typedef sequence <ToBeInitiatedTest> ToBeInitiatedTestSeq;

/*
Define the structure returned by initiate_tests
*/
struct InitiateTestsResponseElement
{
    // If failureReason is NULL, the test is initiated successfully and
    // testInvocationId contains the invocation id. In case the tester object name is not
    // provided in the request, it shall be carried by testerObjectDN. In case the tester
    // object name is provided in the request tODN shall be NULL.
    // Else, the test initiation fails and failureReason contains
    // the failure reason and testInvocationId contains garbage.
    string failureReason;
    string testInvocationId;
    string tODN;
};
typedef sequence <InitiateTestsResponseElement> InitiateTestsResponse;

/*
Define a seq of to-be-terminated-test
*/
typedef string TestInvocationId;
typedef sequence <TestInvocationId> ToBeTerminatedTestSeq;

/*
Define the structure returned by terminate_tests
*/
struct TerminateTestsResponseElement
{
    // If failureReason is NULL, the test has terminated successfully and
    // testInvocationId identifies the terminated invocation.
    // Else, the test termination fails and failureReason contains
    // the failure reason and testInvocationId contains garbage.
    string failureReason;
    string testInvocationId;
};
typedef sequence <TerminateTestsResponseElement> TerminateTestsResponse;

/*
Define the structure of a TOAttributes.
*/
struct TOAttributes
{
    TestStateType testState;
    TestOutcomeType testOutcome;
    NVPairs attributesInNVPairs;
};
};
#endif // _TESTMANAGEMENTIRPCONSTDEFS_IDL_

```

A.2 IDL specification (file name "TestManagementIRPSystem.idl")

```
// File: TestManagementIRPSystem.idl

#ifndef _TESTMANAGEMENTIRPSYSTEM_IDL_
#define _TESTMANAGEMENTIRPSYSTEM_IDL_

#include "TestManagementIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: TestManagementIRPSystem
This module contains the specification of all methods of TestManagement IRP Agent.
=====
*/
module TestManagementIRPSystem
{
    /*
    System may fail to complete an operation. System can provide reason
    to qualify the failed reason. The semantics carried in reason
    is outside the scope of this IRP.
    */
    exception GetTestManagementIRPVersions { string reason; };
    exception GetTestManagementIRPOperationsProfile { string reason; };
    exception GetTestManagementIRPNotificationProfile { string reason; };
    exception InitiateTests { string reason; };
    exception TerminateTests { string reason; };
    exception MonitorTest { string reason; };

    interface TestManagementIRP
    {
        /*
        Return the list of all supported TestManagement IRP versions.
        */
        ManagedGenericIRPConstDefs::VersionNumberSet
        get_Test_Management_IRP_versions (
        )
        raises (GetTestManagementIRPVersions);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific TestManagement IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_Test_Management_IRP_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            test_management_irp_version
        )
        raises (GetTestManagementIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

        /*
        Return the list of all supported notifications and their supported
        parameters for a specific TestManagement IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_Test_Management_IRP_notification_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            test_management_irp_version
        )
        raises (GetTestManagementIRPNotificationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);
    }
}

```



```
/*
Request to initiate tests.
*/
TestManagementIRPConstDefs::InitiateTestsResponse
initiate_tests (
    in TestManagementIRPConstDefs::TestInvocationInitiator
        test_invocation_initiator,
    in TestManagementIRPConstDefs::ToBeInitiatedTestSeq
        to_be_initiated_test_seq
)
raises (InitiateTests,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Request to terminate tests.
*/
TestManagementIRPConstDefs::TerminateTestsResponse
terminate_tests (
    in TestManagementIRPConstDefs::ToBeTerminatedTestSeq
        to_be_terminated_test_seq
)
raises (TerminateTests,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Request test info (to monitor a test).
*/
ManagedGenericIRPConstDefs::Signal_monitor_test (
    in TestManagementIRPConstDefs::ToBeMonitoredTO
        to_be_monitored_TO,
    out TestManagementIRPConstDefs::TOAttributes to_attributes
)
raises (MonitorTest,
        ManagedGenericIRPSystem::InvalidParameter);

};
#endif // _TESTMANAGEMENTIRPSYSTEM_IDL_
```

A.3 IDL specification (file name "TestManagementIRPNotifications.idl")

```
// File: TestManagementIRPNotifications.idl

#ifndef _TESTMANAGEMENTIRPNOTIFICATIONS_IDL
#define _TESTMANAGEMENTIRPNOTIFICATIONS_IDL

#include "TestManagementIRPConstDefs.idl"
#include "NotificationIRPConstDefs.idl"
#include "NotificationIRPNotifications.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: TestManagementIRPNotifications
This module contains the specification of all notifications of Test Management IRP Agent.
=====
*/

module TestManagementIRPNotificationsfDefs
{
  /**
   * Constant definitions for the notifyTestResult notification
   */
  interface notifyTestResult: NotificationIRPNotifications::Notify
  {
    const string EVENT_TYPE = "notifyTestResult";

    /**
     * This constant defines the name of the period property,
     * which is transported in the filterable_body fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestInvocationInitiator.
     */
    const string TEST_INVOCATION_INITIATOR =
      TestManagementIRPConstDefs::AttributeNameValue::TEST_INVOCATION_INITIATOR;

    /**
     * This constant defines the name of the
     * TestInvocationId property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestInvocationId.
     */
    const string TEST_INVOCATION_ID =
      TestManagementIRPConstDefs::AttributeNameValue::TEST_INVOCATION_ID;

    /**
     * This constant defines the name of the
     * TestActualStartTime property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestActualStartTime.
     */
    const string TEST_ACTUAL_START_TIME =
      TestManagementIRPConstDefs::AttributeNameValue::TEST_ACTUAL_START_TIME;

    /**
     * This constant defines the name of the
     * TestActualStopTime property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestActualStopTime.
     */
    const string TEST_ACTUAL_STOP_TIME =
      TestManagementIRPConstDefs::AttributeNameValue::TEST_ACTUAL_STOP_TIME;

    /**
     * This constant defines the name of the

```

```
* testOutcome property,
* which is transported in the remaining_body
* fields.
* The data type for the value of this property
* is TestManagementIRPConstDefs:: testOutcome.
*/
const string TEST_OUTCOME = TestManagementIRPConstDefs::AttributeNameValue::TEST_OUTCOME;

/*
* This constant defines the name of the
* MORT property,
* which is transported in the remaining_body
* fields.
* The data type for the value of this property
* is TestManagementIRPConstDefs::MORT.
*/
const string MORT = TestManagementIRPConstDefs::AttributeNameValue::MORT;

/*
* This constant defines the name of the
* ProposedRepairActions property,
* which is transported in the remaining_body
* fields.
* The data type for the value of this property
* is TestManagementIRPConstDefs::ProposedRepairActions.
*/
const string PROPOSED_REPAIR_ACTIONS =
    TestManagementIRPConstDefs::AttributeNameValue::PROPOSED_REPAIR_ACTIONS;

/*
* This constant defines the name of the
* AdditionalInformation property,
* which is transported in the remaining_body
* fields.
* The data type for the value of this property
* is TestManagementIRPConstDefs:: AdditionalInformation.
*/
const string ADDITIONAL_INFORMATION =
    TestManagementIRPConstDefs::AttributeNameValue::ADDITIONAL_INFORMATION;

/*
* This constant defines the name of the
* FileReference property,
* which is transported in the remaining_body
* fields.
* The data type for the value of this property
* is TestManagementIRPConstDefs:: FdditionalInformation.
*/
const string FILE_REFERENCE = TestManagementIRPConstDefs::AttributeNameValue::FILE_REFERENCE;

/*
* This constant defines the name of the
* FileExpiryDate property,
* which is transported in the remaining_body
* fields.
* The data type for the value of this property
* is TestManagementIRPConstDefs:: FileExpiryDate.
*/
const string FILE_EXPIRY_DATE =
    TestManagementIRPConstDefs::AttributeNameValue::FILE_EXPIRY_DATE;

};

};

#endif // _TESTMANAGEMENTIRPNOTIFICATIONS_IDL_
```

Annex B (informative): Change history

Change history								
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Cat	Old	New
Jun 2002	SA_16	SP-020328	--	--	Submitted to TSG SA #16 for Information	--	1.0.0	
Sep 2002	SA_17	SP-020458	--	--	Submitted to TSG SA #17 for Approval	--	2.0.0	5.0.0
Dec 2002	--	--	--	--	Cosmetics	--	5.0.0	5.0.1
Jun 2004	SA_24	SP-040243	0001	--	Add missing parameter to the operation initiateTests	F	5.0.1	5.1.0
Sep 2004	SA_25	SP-040541	--	--	Automatic upgrade to Rel- 6 (no CR) as per request in SP-040541 SA5_presentation_SA_25.ppt (slide 17)	--	5.1.0	6.0.0
Dec 2004	SA_26	SP-040797	0002	--	Align the IDL style inthe CORBA SS with the IDL Style Guide in 32.150	F	6.0.0	6.1.0
Mar 2005	SA_27	SP-050035	0003	--	IDL incompliant to the style guide	F	6.1.0	6.2.0
Mar 2006	SA_31	SP-060089	0004	--	Correct case sensitive parameter name	F	6.2.0	6.3.0
Jun 2007	SA_36	--	--	--	Automatic upgrade to Rel-7 (no CR) at freeze of Rel-7. Deleted reference to CMIP SS, discontinued from R7 onwards.	--	6.3.0	7.0.0
Mar 2008	SP-39	SP-080056	0006	--	Correct filterability of testInvocationInitiator, alignment with 32.322 IS	A	7.0.0	7.1.0
Jun 2008	SP-40	SP-080329	0007	--	Add new IOCs for connection and loopback test - Align with 32.322 IS	B	7.1.0	8.0.0
Dec 2009	-	-	-	-	Update to Rel-9 version (MCC)	-	8.0.0	9.0.0

History

Document history		
V9.0.0	February 2010	Publication