INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.834.4

Amendment 1

(01/2004)

SERIES Q: SWITCHING AND SIGNALLING

Q3 interface

A CORBA interface specification for Broadband Passive Optical Networks based on UML interface requirements

Amendment 1

ITU-T Recommendation Q.834.4 (2003) – Amendment 1

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4, 5, 6, R1 AND R2	Q.120–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.799
Q3 INTERFACE	**Q.800–Q.849**
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1699
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900–Q.1999
BROADBAND ISDN	Q.2000–Q.2999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Q.834.4

A CORBA interface specification for Broadband Passive Optical Networks based on UML interface requirements

Amendment 1

Summary

This amendment provides several enhancements to the CORBA interface specification for the management interface between a Supplier Management System and an Operator Management System managing Broadband Passive Optical Networks (BPONs). In includes changes needed for management of dynamic bandwidth allocation and automatic ranging of Optical Network Units (ONUs) and Optical Network Terminals (ONTs). It also includes additions required to adequately support testing of telephony and data services, software activity notification, and capacity management.

Source

Amendment 1 to ITU-T Recommendation Q.834.4 (2003) was approved on 13 January 2004 by ITU-T Study Group 4 (2001-2004) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation Q.834.4

A CORBA interface specification for Broadband Passive Optical Networks based on UML interface requirements

Amendment 1

1 Scope

This amendment provides several enhancements to the CORBA interface specification for the management interface between a Supplier Management System and an Operator Management System managing Broadband Passive Optical Networks (BPONs). In includes changes needed for management of dynamic bandwidth allocation and automatic ranging of Optical Network Units (ONUs) and Optical Network Terminals (ONTs). It also includes additions required to adequately support testing of telephony and data services, software activity notification, and capacity management.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[1] ITU-T Recommendation G.983.4 (2001), *A broadband optical access system with increase service capability using dynamic bandwidth assignment (DBA)*.

[2] ITU-T Recommendation G.983.7 (2001), *ONT Management and Control Interface specification for Dynamic Bandwidth Assignment (DBA) B-PON system*.

[3] ITU-T Recommendation G.983.4 (2001)/Amd.1 (2003), *A broadband optical access system with increased service capability using dynamic bandwidth assignment (DBA), Amendment 1: New Annex A – Performance monitoring parameters*.

[4] ITU-T Recommendation G.983.1 (1998), *Broadband optical access systems based on Passive Optical Networks (PON)*.

3 Terms and definitions

For the purposes of this amendment, the following definitions apply:

3.1 Terms imported from ITU-T Rec. G.983.4

The following terms from ITU-T Rec. G.983.4 [1] are used in this amendment.

– Dynamic Bandwidth assignment;
– T-CONT;
– Guaranteed Bandwidth;
– Fixed Bandwidth;
– Assured Bandwidth;
– Maximum Bandwidth.

3.2 New terms

This amendment defines no new terms.

4 Abbreviations

This amendment uses no new abbreviations in addition to those found in the main Recommendation.

5 Conventions

This amendment has no new conventions in addition to those found in the main Recommendation.

6 Enhancements to Build and Common interfaces

This clause describes changes to Q834::Build and Q834::Common interface required to support the management of dynamic bandwidth allocation as well as the automatic ranging of ONTs or ONUs following one of the ranging mechanisms defined in ITU-T Rec. G.983.1. First, operations used to construct or modify attributes for ONT or ONU network elements are modified to include input parameters required to properly characterize such network elements supporting dynamic bandwidth allocation and automatic ranging. Second, new operations are added to provide the capability of construction of TCONTs, a partitioning of bandwidth intended for use at an ONT or ONU in conjunction with the dynamic bandwidth allocation protocol mechanisms. These operations are called buildTCONT, modifyTCONTParameters, and deleteTCONT. The new operations create the need for a new exception called UnsupportedTCONTType, the enhancement of the definition of ParameterViolation exception, and the new use of an existing exception, InsufficientPONBW, by Q834::Build interface.

Third, the data structures used to characterize ONTs and ONUs in Q834::Common are modified to include these new attributes. Finally, new HistoryData record types are added to Q834::Common in order to allow for the performance monitoring of the dynamic bandwidth allocation function. They are called DBAFairnessPMHistoryDataType and TCONTTEPMHistoryDataType.

All changes to Q834::Common and associated changes to Annexes A and B are combined with other enhancements and presented in clauses 12.2 , 10, and 11, respectively.

6.1 Modifications to clause 9.2.1.1, "buildNode"

a) Replace:

```
"                in AdministrationDomainType administrationDomain)
                 raises (UnrecognisedVersion, InvalidSerialNumSyntax,
                 DuplicateSerialNumber, UnknownProfiles,"
```

with:

```
"                in RegistrationIdType registrationId,
                 in boolean sRInd,
                 in AdministrationDomainType administrationDomain)
                 raises (UnrecognisedVersion, InvalidSerialNumSyntax,
                 DuplicateSerialNumber, UnknownProfiles, ParameterViolation,"
```

b) Replace:

"The input **administrationDomain** identifies the domain to which the NE belongs."

with:

"The input **registrationId** provides the logical key to associate this provisioning data with an installed NE. The input **sRInd** specifies whether or not the NE should be status reporting if the NE constructed is an ONT or ONU and dynamic bandwidth allocation is supported. The input **administrationDomain** identifies the domain to which the NE belongs."

6.2 Modifications to clause 9.2.1.2, "assignUserLabelsToNE"

a) Replace:

```
"                       in AdministrationDomainType administrationDomain)"
```

with:

```
"                       in RegistrationIdType registrationId,
                        in AdministrationDomainType administrationDomain)"
```

b) Replace:

"The input **administrationDomain** identifies the domain to which the NE is assigned."

with:

"The input **registrationId** provides the logical key to associate this provisioning data with an installed NE. The input **administrationDomain** identifies the domain to which the NE belongs."

6.3 Modifications to clause 9.2.1.3, "modifyNode"

a) Replace:

```
"                   in AdministrationDomainType administrationDomain)
                    raises (UnknownManagedEntity, UnknownNE,
                    InvalidSlotAssignmentList, UnknownProfiles,
                    DuplicateUserLabel, AccessDenied, InvalidExternalTime,
                    ProfileSuspended);"
```

with:

```
"                   in RegistrationIdType registrationId,
                    in boolean sRInd,
                    in AdministrationDomainType administrationDomain)
                    raises (UnknownManagedEntity, UnknownNE,
                    InvalidSlotAssignmentList, UnknownProfiles,
                    DuplicateUserLabel, AccessDenied, InvalidExternalTime,
                    ProfileSuspended, ParameterViolation);"
```

b) Replace:

"The input **administrationDomain** identifies the domain to which the NE is reassigned."

with:

"The input **registrationId** provides the logical key to associate this provisioning data with an installed NE. The input **sRInd** specifies the new value for the status reporting characteristic of the ONT or ONU. The input **administrationDomain** identifies the domain to which the NE belongs."

6.4 buildTCONT

This operation builds a TCONT in the Supplier Management System. The ONT or ONU associated with the TCONT must be provisioned prior to this operation.

The operation signature for **buildTCONT** is shown below:

```
ManagedEntityIdType  buildTCONT (
                        in ManagedEntityIdType nEId,
                        in TCONTType  typeOFTCONT,
                        in BWType maxBW,
                        in BWType fixedBW,
                        in BWType guaranteedBW,
                        in UserLabelType userLabel)
                        raises (UnknownNE, AccessDenied, InsufficientPONBW,
                        UnsupportedTCONTType, DuplicateUserLabel);
```

The input **nEId** identifies the ONT or ONU where the TCONT is to be terminated. The input parameter **typeOFTCONT** identifies which of the five TCONT types is to be constructed. The input parameters **maxBW**, **fixedBW**, and **guaranteedBW** identify the bandwidth attributes that characterize the logical link end. The input **userLabel** provides a unique operator designation for the constructed TCONT, if such identification is desired.

The return value of type **ManagedEntityIdType** provides unique identifier for the constructed TCONT.

6.5 modifyTCONTParameters

This operation modifies the parameters associated with an existing TCONT in the Supplier Management System.

The operation signature for **modifyTCONTParameters** is shown below:

```
void modifyTCONTParameters (
                        in ManagedEntityIdType tCONTId,
                        in BWType maxBW,
                        in BWType fixedBW,
                        in BWType guaranteedBW,
                        in UserLabelType userLabel)
                        raises (UnknownManagedEntity, AccessDenied,
                        InsufficientPONBW, DuplicateUserLabel);
```

The input **tCONTId** identifies the TCONT whose characteristics are to be modified. The input parameters **maxBW**, **fixedBW**, and **guaranteedBW** identify the new bandwidth attributes that should characterize the logical link end. The input **userLabel** provides a unique operator designation for the constructed TCONT, if such identification is desired.

The return value is of type **void**.

6.6 deleteTCONT

This operation deletes a TCONT provisioning from the Supplier Management System. As a consequence of this operation, all managed entities automatically created as a result of the corresponding buildTCONT operation are deleted as well. A TCONT can not be deleted if there are outstanding active subnetwork connections associated with it.

The operation signature for **deleteTCONT** is shown below:

```
void deleteTCONT (
            in ManagedEntityIdType tCONTId)
            raises (UnknownManagedEntity, AccessDenied,
            RemainingSubnetworkConnections);
```

The input **tCONTId** identifies the bridge to be deleted.

The return value is of type **void**.

6.7 Modifications to clause 9.2.1.20, "Exceptions"

a) Replace:

"The exception **ParameterViolation** is raised when the VPI is out of range or a duplicate."

with:

"The exception **ParameterViolation** is raised when the VPI is out of range or a duplicate or when the version of ONT or ONU being provisioned can not support the status reporting option specified in the request."

b) Add the following in alphabetical order:

"The exception **InsufficientPONBW** is raised when the TCONT can not be built or modified due to insufficient bandwidth on the APONLink.

UnsupportedTCONTType exception is raised if the TCONT type (1-5) specified in the operation is not supported by the provisioned ONT or ONU."

7 Enhancements to Test interface

In order to sectionalize and identify problems with telephony and data service offerings supported by BPON technology, Q834::Test interface is enhanced to include three new test types: metallicDropTest, mACLayerTest, and drawDialtoneBreakTest. Scheduled versions of these test types are also required and included. As there are now four different types of scheduled tests, generalization of modifyResourceSelfTest and cancelResourceSelfTest (called modifyTestSchedule and cancelScheduledTest) are provided as replacements. Finally, wording is added to 9.15.1.7 to enhance the interpretation of conductResourceSelfTest to support testing of BORSHT functionality on an ONT or NT.

All changes to C.15 are combined with other enhancements to Annex C found in the original Recommendation and are presented in 12.4.

7.1 metallicDropTest

This operation is used to conduct a metallic drop test following the identification of a system fault or a subscriber service complaint.

The operation signature for the **metallicDropTest** is shown below:

```
DropTestResultsType metallicDropTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId)
                  raises (AccessDenied,
                        CommFailure,
                        UnknownManagedEntity);
```

The input parameter **testRequestorId** is used to identify the initiator of the test. The input parameter **port** identifies the UNI telephony service port. The input parameter **serviceInstanceId** identifies the service instance associated with the drop test request.

The return value is of type **DropTestResultsType** and provides information on the test, specifically showing all pass or indicating the failure measurement results for the point at which the metallic drop test failed.

7.2 scheduleMetallicDropTest

This operation is used to schedule a metallic drop test. This operation is used by the OMS to set up metallic drop tests for telephony service ports to be executed at a regular basis. Having a scheduler object set up is a pre-requisite for initiating this operation.

The operation signature for the **scheduleMetallicDropTest** is shown below:

```
TestTrackingObjectIdType scheduleMetallicDropTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId,
                  in UserLabelType schedulerName)
                  raises (AccessDenied,
                        UnknownManagedEntity,
                        UnknownScheduler,
                        InvalidScheduler);
```

The input parameter **testRequestorId** is used to identify the initiator of the test. The input parameter **port** identifies the UNI telephony service port. The input parameter **serviceInstanceId** identifies the service instance associated with the test request. The input parameter **schedulerName** is used to reference the applicable scheduler for this test.

The return value of type **TestTrackingObjectIdType** uniquely identifies the scheduled testing. The Test Tracking Object exists until it is explicitly cancelled using the **cancelScheduledTest** operation or if the scheduler end time is reached.

The test results are logged. The **DropTestResultsType** data type defines part of the information that is logged.

7.3 mACLayerTest

This operation is used to conduct a MAC layer test following the identification of a system fault or a subscriber service complaint. The test uses the broadcast ARP messaging to detect the presence of customer-provided equipment, powered on and connected to a LAN port of an ONT or NT.

The operation signature for the **mACLayerTest** is shown below:

```
short mACLayerTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId)
                  raises (AccessDenied,
                        CommFailure,
                        UnknownManagedEntity);
```

The input parameter **testRequestorId** is used to identify the initiator of the test. The input parameter **port** identifies the UNI LAN service port. The input parameter **serviceInstanceId** identifies the service instance associated with the test request.

The return value is of type **short** providing the count of customer premises devices detected by this test.

7.4 scheduleMACLayerTest

This operation is used to schedule a MAC Layer test. This operation is used by the OMS to set up this type of test for LAN service ports to be executed at a regular basis. Having a scheduler object set up is a pre-requisite for initiating this operation.

The operation signature for the **scheduleMACLayerTest** is shown below:

```
TestTrackingObjectIdType scheduleMACLayerTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId,
                  in UserLabelType schedulerName)
                  raises (AccessDenied,
                        UnknownManagedEntity,
                        UnknownScheduler,
                        InvalidScheduler);
```

The input parameter **testRequestorId** is used to identify the initiator of the test. The input parameter **port** identifies the UNI LAN service port. The input parameter **serviceInstanceId** identifies the service instance associated with the test request. The input parameter **schedulerName** is used to reference the applicable scheduler for this test.

The return value of type **TestTrackingObjectIdType** uniquely identifies the scheduled testing. The Test Tracking Object exists until it is explicitly cancelled using the **cancelScheduledTest** operation or if the scheduler end time is reached. The test results are logged.

7.5 drawDialToneBreakTest

This operation is used to conduct a telephony service related test following the identification of a system fault or a subscriber service complaint. The test verifies the signalling path between an ONT or NT and a local switch for a telephone service instance.

The operation signature for the **drawDialToneBreakTest** is shown below:

```
boolean drawDialToneBreakTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId)
                  raises (AccessDenied,
                        CommFailure,
                        UnknownManagedEntity);
```

The input parameter **testRequestorId** is used to identify the initiator of the test. The input parameter **port** identifies the UNI telephony service port. The input parameter **serviceInstanceId** identifies the service instance associated with the drop test request.

The return value is of type **boolean** indicating pass/fail.

7.6 scheduleDrawDialToneBreakTest

This operation is used by the OMS to set up draw dialtone/break dialtone testing for telephony service ports to be executed at a regular basis. Having a scheduler object set up is a prerequisite for initiating this operation.

The operation signature for the **scheduleDrawDialToneBreakTest** is shown below:

```
TestTrackingObjectIdType scheduleDrawDialToneBreakTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId,
                  in UserLabelType schedulerName)
                  raises (AccessDenied,
                        UnknownManagedEntity,
                        UnknownScheduler,
                        InvalidScheduler);
```

The input parameter **testRequestorId** is used to identify the initiator of the test. The input parameter **port** identifies the UNI telephony service port. The input parameter **serviceInstanceId** identifies the service instance associated with the test request. The input parameter **schedulerName** is used to reference the applicable scheduler for this test.

The return value of type **TestTrackingObjectIdType** uniquely identifies the scheduled testing. The Test Tracking Object exists until it is explicitly cancelled using the **cancelScheduledTest** operation or if the scheduler end time is reached. The test results are logged.

7.7 Modifications to clause 9.15.1.5, "modifyResourceSelfTestSchedule"

Replace the entire clause with the following:

"**9.15.1.5 modifyTestSchedule**

This operation is used to modify the schedule for a regularly conducted test. If successful, the test initiation is changed with the next iteration.

The operation signature for **modifyTestSchedule** is shown below:

```
void  modifyTestSchedule (
                        in TestTrackingObjectIdType testTrackingObjectId,
                        in UserLabelType  newSchedulerName)
                        raises (UnknownTest, UnknownScheduler, InvalidScheduler,
                        AccessDenied);
```

The input parameter **testTrackingObjectId** is used to identify the scheduled test. The input parameter **newSchedulerName** is used to identify the new schedule.

The return value is of type **void**."

7.8 Modifications to clause 9.15.1.6, "cancelScheduledResourceSelfTest"

Replace the entire clause with the following:

"**9.15.1.6 cancelScheduledTest**

This operation is used to cancel a regularly scheduled test. If successful, this operation cancels the test prior to its initiation with the next trigger time.

The operation signature for **cancelScheduledTest** is shown below:

```
void  cancelScheduledTest (
            in TestTrackingObjectIdType testTrackingObjectId)
            raises (UnknownTest, UncontrolledTestInProgress, AccessDenied);
```

The input parameter **testTrackingObjectId** is used to identify the scheduled test to terminate.

The return value is of type **void**."

7.9 Modifications to clause 9.15.1.7, "conductResourceSelfTest"

Replace:

"This operation is used to initiate a resource self test following the identification of a system fault or a subscriber service complaint."

with:

"This operation is used to initiate a resource self test following the identification of a system fault or a subscriber service complaint. Resource self test can be used to test BORSHT functionality in an ONT or NT."

8 Enhancements to Event Supplier interface

In order to notify the Operator Management System of the completion of scheduled or lengthy management activities requested by the Operator Management System, another interface and mapping to the structured event object is provided. At the moment, the primary use for this interface is notify of completion for NE software management activities.

All associated changes to C.6 are combined with other enhancements to Annex C found in the original Recommendation and are presented in 12.3.

8.1 ActivityCompletionEventSupplier

The purpose of this interface is to announce events to the Operator Management System via the OMG Notification Service concerning the completion of scheduled or time-consuming management requests made by the Operator Management System to the Supplier Management System. This interface has no operations. However, it does provide the fixed header mapping as well as the filterable data mappings for the structured event object used to push event information through the event channel of the OMG Notification Service.

In the fixed header, the **domain_type** is set to "telecommunications", the **type_name** is set to "ActivityCompletion", and the **event_name** is set to a constant string that has one of the following values: "SoftwareDownload", "SoftwareCommit", or "SoftwareActivation".[1]

The mapping in the filterable data consists of pairs of items. The first component in the pair is a string identifier for a data name and the second is the value for that data element. The string identifiers are constants that are defined in this interface. Furthermore, the filterable data pairs must occur in a specific order.

The order of the filterable items for an event_name of "SoftwareDownload", "SoftwareCommit", or "SoftwareActivation" is as follows:

- EventTime;
- NotificationIdentifier;
- CorrelatedNotifications;
- TrackingObjectId;
- SuccessIndication;
- AdditionalText.

The value for EventTime has syntax of GeneralizedTimeType and refers to the moment when the software activity was completed in the network element.

---

[1] For the moment this interface only deals with activities involving software download to network elements.

The value for NotificationIdentifier has syntax NotificationIdentifierType and it provides a reference sequence number for the event. The value for CorrelatedNotifications has syntax of CorrelatedNotificationType and supplies a list of reference numbers for other event notifications provided by the Supplier Management System for associated software activities. If there are no related notifications the value of the empty set is supplied.

The value for TrackingObjectId has syntax TrackingObjectIdType and identifies the scheduled or time-consuming management activity requested by the Operator Management System. This identifier was provided by the Supplier Management System at the time of the request. The value of SuccessIndication has boolean syntax, and indicates whether or not the activity was successfully completed.

Finally, the value for AdditionalText has syntax string. This data item provides a location to pass any textual information from the Supplier Management System concerning the failure of the activity to complete. If there is no additional information, the null string will be passed.

8.2 Table 1, "q834_4 module organization"

Replace line C.6 with the following:

"

C.6	Q834EventPublisher	AlarmEventSupplier, SecurityEventSupplier, DiscoveryEventSupplier ActivityCompletionSupplier	Publish Events

"

9 Enhancements to Common Interface for Capacity Management

Interface Q834::Common is enhanced by the addition of an OLT resource record as well as several port resource records. These record types are required to assist in the transfer of information concerning utilization of physical and logical resources associated with BPON network elements. These items are called OLTResourceRecordType, ATMPortResourceRecordType, TDMPhysicalPortResourceRecordType, and EthernetPhysicalPortResourceRecordType. The only place where these additions are made in the original Recommendation is in C.3.

All changes to C.3 are combined with other enhancements to Annex C found in the original Recommendation and are presented in 12.2.

10 Enhancements to Annex A, "Data dictionary"

Insert, in alphabetical order, the following items in Table A.1, "Data elements and definitions":

"

BWType	This attribute identifies the amount of bandwidth assigned.	float	
DBAFairnessPMHistoryDataType	Provides the record structure itemizing the performance data collected in a 15-minute interval for monitoring point TCAdaptor at a PON port of the OLT.	struct	
DropTestResultsType	This data element provides the test results following the completion of a metallic drop test.	struct	
RegistrationIdType	This data element provides the correlation key used to build the association between pre-provisioned service connections and an NE yet to be installed.	string	
TCONTTEPMHistoryDataType	Provides the record structure itemizing the performance data collected in a 15-minute interval for monitoring point TCONTBuffer at a PON port of the OLT.	struct	
TCONTType	This attribute specifies the type of TCONT constructed.	short	Permitted values are 1, 2, 3, 4, 5

"

11 Enhancements to Annex B, "Exceptions"

Insert, in alphabetical order, the following item in Table B.1, Exceptions:

UnsupportedTCONTType	This exception is raised if the TCONT type specified in the operation is not supported by the specified ONT or ONU.

12 Enhancements to Annex C, "IDL files"

12.1 Enhancements to clause C.2, "Q834Build.idl"

12.1.1 Imports, typedefs, and exceptions

Add the following imports, typedefs, and exceptions:

"
```
typedef Q834Common::RegistrationIdType RegistrationIdType;
#define InsufficientPONBW   Q834Common::InsufficientPONBW;
#define UnknownNE  Q834Common::UnknownNE;
typedef float  BWType;
typedef short TCONTType;
exception UnsupportedTCONTType {};"
```

12.1.2 Interface Builder

a) Replace the following:

```
"// See 9.2.1.1 for the description of the behaviour of this operation

ManagedEntityIdType buildNode (
                  in NEKindType nEKind,
                  in string supplierName,
                  in string location,
                  in VersionType hWVersion,
                  in SerialNumType serialNum,
                  in NameSeqType alarmSeverityProfiles,
                  in NameSeqType thresholdDataProfiles,
                  in SlotAssignmentSeqType slotAssignmentList,
                  in ManagedEntityIdType port,  // OLT PON port
                  in string modelCode,
                  in string systemTitle,
                  in VersionSeqType softwareVersions,
                  in UserLabelType nEUserLabel,
                  in ExternalTimeType externalTime,
                  in SystemTimingType systemTiming,
                  in AdministrationDomainType administrationDomain )
                  raises (UnrecognisedVersion,
                        InvalidSerialNumSyntax,
                        DuplicateSerialNumber,
                        UnknownProfiles,
                        UnknownManagedEntity,
                        DuplicateUserLabel,
                        AccessDenied,
                        InvalidExternalTime,
                        UnknownSystemTimingSource,
                        ProfileSuspended);"
```

with:

```
"// See 9.2.1.1 for the description of the behaviour of this operation

"ManagedEntityIdType buildNode (
                  in NEKindType nEKind,
                  in string supplierName,
                  in string location,
                  in VersionType hWVersion,
                  in SerialNumType serialNum,
                  in NameSeqType alarmSeverityProfiles,
                  in NameSeqType thresholdDataProfiles,
                  in SlotAssignmentSeqType slotAssignmentList,
                  in ManagedEntityIdType port,  // OLT PON port
                  in string modelCode,
                  in string systemTitle,
                  in VersionSeqType softwareVersions,
                  in UserLabelType nEUserLabel,
                  in ExternalTimeType externalTime,
                  in SystemTimingType systemTiming,
                  in RegistrationIdType registrationId,
                  in boolean sRInd,
```

```
                in AdministrationDomainType administrationDomain )
                raises (UnrecognisedVersion,
                      InvalidSerialNumSyntax,
                      DuplicateSerialNumber,
                      UnknownProfiles,
                      ParameterViolation
                      UnknownManagedEntity,
                      DuplicateUserLabel,
                      AccessDenied,
                      InvalidExternalTime,
                      UnknownSystemTimingSource,
                      ProfileSuspended,
                      ParameterViolation);"
```

b) Replace the following:

```
"// See 9.2.1.2 for the description of the behaviour of this operation

            void assignUserLabelsToNE (
                  in SerialNumType serialNum,
                  in UserLabelType nEUserLabel,
                  in AdministrationDomainType administrationDomain)
                  raises (InvalidSerialNumSyntax,
                        DuplicateSerialNumber,
                        DuplicateUserLabel,
                        AccessDenied);"
```

with:

```
"// See 9.2.1.2 for the description of the behaviour of this operation

            void assignUserLabelsToNE (
                  in SerialNumType serialNum,
                  in UserLabelType nEUserLabel,
                  in RegistrationIdType registrationId,
                  in AdministrationDomainType administrationDomain)
                  raises (InvalidSerialNumSyntax,
                        DuplicateSerialNumber,
                        DuplicateUserLabel,
                        AccessDenied);"
```

c) Replace the following:

```
"// See 9.2.1.3 for the description of the behaviour of this operation

            void modifyNode (
                  in ManagedEntityIdType managedEntityId,
                  in SlotAssignmentSeqType newSlotAssignmentList,
                  in NameType newAlarmSeverityProfiles,
                  in NameSeqType newThresholdDataProfiles,
                  in ManagedEntityIdType port,  // OLT PON Port
                  in string newModelCode,
                  in UserLabelType newNEuserLabel,
                  in ExternalTimeType externalTime,
```

```
                    in AdministrationDomainType administrationDomain )
                    raises (UnknownManagedEntity,
                          UnknownNE,
                          InvalidSlotAssignmentList,
                          UnknownProfiles,
                          DuplicateUserLabel,
                          AccessDenied,
                          InvalidExternalTime,
                          ProfileSuspended);"
```

with:

```
" // See 9.2.1.3 for the description of the behaviour of this operation

              void modifyNode (
                    in ManagedEntityIdType managedEntityId,
                    in SlotAssignmentSeqType newSlotAssignmentList,
                    in NameSeqType newAlarmSeverityProfiles,
                    in NameSeqType newThresholdDataProfiles,
                    in ManagedEntityIdType port,  // OLT PON Port
                    in string newModelCode,
                    in UserLabelType newNEuserLabel,
                    in ExternalTimeType externalTime,
                    in RegistrationIdType registrationId,
                    in boolean sRInd,
                    in AdministrationDomainType administrationDomain )
                    raises (UnknownManagedEntity,
                          UnknownNE,
                          InvalidSlotAssignmentList,
                          UnknownProfiles,
                          DuplicateUserLabel,
                          AccessDenied,
                          InvalidExternalTime,
                          ProfileSuspended,
                          ParameterViolation);"
```

d) Add the following operations:

```
"// See 9.2.1.20 for the description of the behaviour of this operation

ManagedEntityIdType  buildTCONT (
                          in ManagedEntityIdType nEId,
                          in TCONTType  typeOFTCONT,
                          in BWType maxBW,
                          in BWType fixedBW,
                          in BWType guaranteedBW,
                          in UserLabelType userLabel)
                          raises (UnknownNE, AccessDenied, InsufficientPONBW,
                          UnsupportedTCONTType, DuplicateUserLabel);

// See 9.2.1.21 for the description of the behaviour of this operation

void  modifyTCONTParameters (
                          in ManagedEntityIdType tCONTId,
                          in BWType maxBW,
                          in BWType fixedBW,
                          in BWType guaranteedBW,
                          in UserLabelType userLabel)
                          raises (UnknownManagedEntity, AccessDenied,
                          InsufficientPONBW, DuplicateUserLabel);
```

```
// See 9.2.1.22 for the description of the behaviour of this operation

void deleteTCONT (
            in ManagedEntityIdType tCONTId)
            raises (UnknownManagedEntity, AccessDenied,
            RemainingSubnetworkConnections);"
```

12.2 Enhancements to clause C.3, "Q834Common.idl"

12.2.1 Structures and typedefs

a) Add the following typedefs and structures:

```
"typedef boolean SRIndType;

typedef string RegistrationIdType;

struct DBAFairnessPMHistoryDataType {
            long long recordId;
            ManagedEntityIdType monitoringPoint;
            GeneralizedTimeType periodEndTime;
            boolean suspectIntervalFlag;
            NameType thresholdDataId;
            float type2Variance; //value = -1 if type 2 is unsupported by NE
            float type3Variance; //value = -1 if type 3 is unsupported by NE
            float type4Variance; //value = -1 if type 4 is unsupported by NE
            float type5Variance; //value = -1 if type 5 is unsupported by NE
      };

struct TCONTTEPMHistoryDataType {
            long long recordId;
            ManagedEntityIdType monitoringPoint;
            GeneralizedTimeType periodEndTime;
            boolean suspectIntervalFlag;
            NameType thresholdDataId;
            float averageReceive_AssignRate;
            float maximumReceive_AssignRate;
            float minimumReceive_AssignRate;
      };


struct OLTResourceRecordType  {
            long long recordId;
            ManagedEntityIdType     oLTResourceId;
            ManagedEntityIdType     containingNEId;
            GeneralizedTimeType     collectionTimestamp;
            SlotAssignmentSeqType   slotAssignmentList;
      };

struct ATMPortResourceRecordType    {
            long long recordId;
            ManagedEntityIdType     portResourceId;
            ManagedEntityIdType     portId;
            GeneralizedTimeType     collectionTimestamp;
            unsigned long           maxVPConnectionCount;
            unsigned long           maxVCConnectionCount; //Zero if APON port
            unsigned long           reservedVPConnectionCount;
            unsigned long           reservedVCConnectionCount; //Zero if APON
                                    port
            unsigned long           assignedVPConnectionCount;
            unsigned long           assignedVCConnectionCount; //Zero if APON
                                    port
```

```
            float       maxBW;
            float       reservedBW;
            float       assignedBW;
      }; //PON ports in BPON support ATM

struct TDMPhysicalPortResourceRecordType  {
            long long recordId;
            ManagedEntityIdType     portResourceId;
            ManagedEntityIdType     portId;
            GeneralizedTimeType     collectionTimestamp;
            unsigned long           maxTSCount;
            unsigned long           reservedTSCount;
            unsigned long           assignedTSCount;
      };

struct EthernetPhysicalPortResourceRecordType   {
            long long recordId;
            ManagedEntityIdType     portResourceId;
            ManagedEntityIdType     portId;
            GeneralizedTimeType     collectionTimestamp;
            unsigned long           maxVLANTagCount;
            unsigned long           reservedVLANTagCount;
            unsigned long           assignedVLANTagCount;
      };"
```

b) Replace the following:

```
"struct     ONTType     {
            NEFSANType  nEFSAN;
            ManagedEntityIdType     upstreamNEFSAN;
            };"
```

with:

```
"struct     ONTType     {
            NEFSANType  nEFSAN;
            ManagedEntityIdType     upstreamNEFSAN;
            SRIndType   sRInd;
            short  maxDataGrants;
            RegistrationIdType registrationId;
            };"
```

c) Replace the following:

```
"struct     ONUType     {
            NEFSANType  nEFSAN;
            ManagedEntityIdType     upstreamNEFSAN;
            ManagedEntityIdSeqType  subtendingNEFSANList;
            };"
```

with:

```
"struct     ONUType     {
            NEFSANType  nEFSAN;
            ManagedEntityIdType     upstreamNEFSAN;
            ManagedEntityIdSeqType  subtendingNEFSANList;
            SRIndType   sRInd;
            short maxDataGrants;
            RegistrationIdType registrationId;
      };"
```

d) Replace the following:

```
"struct     PlugInUnitFType   {
            ManagedEntityIdType     plugInUnitFId;
            ManagedEntityIdType     containingNEId;
            EquipmentHolderAddressType    containingSlotAddress;
            AdministrativeStateType administrativeState;
            AvailabilityStatusSetType     availabilityStatus;
            OperationalStateType    operationalState;
            string      equipmentCode;
            string      functionCode;
            string      supplierName;
            VersionType hardwareVersion;
            string      serialNumber;
            short       portCount;
            NameSeqType alarmSeverityAssignmentProfileNames;
            NameSeqType thresholdDataNames;
            UserLabelType     circuitPackUserLabel;
            ManagedEntityIdSeqType  supportedByManagedEntityList;
      };"
```

with:

```
      "struct     PlugInUnitFType   {
            ManagedEntityIdType     plugInUnitFId;
            ManagedEntityIdType     containingNEId;
            EquipmentHolderAddressType    containingSlotAddress;
            AdministrativeStateType administrativeState;
            AvailabilityStatusSetType     availabilityStatus;
            OperationalStateType    operationalState;
            string      equipmentCode;
            string      functionCode;
            string      supplierName;
            VersionType hardwareVersion;
            VersionSeqType     softwareVersions;
            string      serialNumber;
            short       portCount;
            NameSeqType alarmSeverityAssignmentProfileNames;
            NameSeqType thresholdDataNames;
            UserLabelType     circuitPackUserLabel;
            ManagedEntityIdSeqType  supportedByManagedEntityList;
      };"
```

12.2.2 Interface ProbableCause

In the ProbableCause interface, add the following definitions:

```
"const unsigned short LOSS_OF_MINISLOT = 35;
const unsigned short STATUS_REPORTING_HANDSHAKE_FAILURE = 36;"
```

12.2.3 Interface PMCategory

In the PMCategory interface, add the following definitions:

```
"const unsigned short DBA_FAIRNESS_PM = 24;
const unsigned short TCONT_TRAFFIC_PM = 25;"
```

12.2.4 Interface MonitoringParameter

In the MonitoringParameter interface, add the following definitions:

```
"const      string type2Var = "Type2Var";
const       string type3Var = "Type3Var";
const       string type4Var = "Type4Var";
const       string type5Var = "Type5Var";
const       string averageRec_AssignRate = "AverageRec_AssignRate";
const       string maximumRec_AssignRate = "MaximumRec_AssignRate";
const       string minimumRec_AssignRate = "MinimumRec_AssignRate";"
```

12.2.5 Interface RecordSetType

In the RecordSetType Interface, insert, in numerical order, the following definitions:

```
"const unsigned short DBAFAIRNESSPMHISTORYDATA = 19;
const unsigned short TCONTTEPMHISTORYDATA = 20;"
const unsigned short OLTRESOURCERECORD = 21;
const unsigned short ATMPORTRESOURCERECORD = 22;
const unsigned short TDMPHYSICALPORTRESOURCERECORD = 23;
const unsigned short ETHERNETPORTRESOURCERECORD = 24;"
```

12.3 Enhancements to clause C.6, "Q834Eventpublisher.idl"

12.3.1 Imports

Add the following import:

"typedef Q834Common::TrackingObjectIdType TrackingObjectIdType;"

12.3.2 ActivityCompletionEventSupplier

Add the new interface:

```
"interface ActivityCompletionEventSupplier : itut_x780::ManagedObject {

/*  Structured event fixed header mappings:
domain_type is set to "telecommunications",
type_name is set to "ActivityEvent", and
event_name is set to one of the following constant strings
provided below.
*/
            const string softwareDownload = "SoftwareDownload";
            const string softwareCommit = "SoftwareCommit";
            const string softwareActivation = "SoftwareActivation";

/* Additional items to be mapped in the filterable data section of the
structured event object are provided below.
*/
            const string eventTime = "EventTime";
            const string additionalText = "AdditionalText";
            const string notificationIdentifier = "NotificationIdentifier";
            const string correlatedNotifications = "CorrelatedNotifications";
            const string successIndication = "SuccessIndication";
            const string trackingObjectId = "TrackingObjectId";

/*
Mapping to filterable data within the structured event is provided below for a
software activity event.

            {
            {"EventTime", any (GeneralizedTimeType)},
```

```
            {"NotificationIdentifier", any (NotificationIdentifierType)},
            {"CorrelatedNotifications", any (CorrelatedNotificationType)},
            {"TrackingObjectId", any (TrackingObjectIdType)},
            {"SuccessIndication", boolean},
            {"AdditionalText", any (string)}
            }
*/

}; // interface ActivityCompletionEventSupplier"
```

12.4 Enhancements to clause C.15, "Q834Test.idl"

12.4.1 Structures and typedefs

Add the following typedefs and structures:

```
"typedef  float  HazardousPotentialType; //in volts

struct ForElectroMotiveForceType {
      float       acVoltageTipToGround;
      float       acVoltageRingToGround;
      float       dcVoltageTipToGround;
      float       dcVoltageRingToGround;
      }; //in volts

struct ResistiveFaultType  {
      float dcResistanceTipToRing;
      float dcResistanceTipToGround;
      float dcResistanceRingToGround;
      }; //in volts

struct ReceiverOffHookType {
      float dcResistance1TipToRing;
      float dcResistance2TipToRing;
      }; //in ohms

struct ReceiverOffHookType {
      float dcResistance1TipToRing;
      float dcResistance2TipToRing;
      }; //in ohms

struct PresenceOfRingerType {
      float acImpedenceTipToRing;
      float acImpedenceTipToGround;
      float acImpedenceRingToGround;
      }; //in ohms

struct NetworkTerm1DCSignatureType {
      float dcVoltage1TipToRing;
      float dcVoltage2TipToRing;
      };

struct   DropTestResultsType {
      short resultsMask;
      HazardousPotentialType hazardousPotential;
      ForElectroMotiveForceType foreignElectroMotiveForce;
      ResistiveFaultType      resistiveFault;
      ReceiverOffHookType     receiverOffHook;
      PresenceOfRingerType    ringer;
      boolean     networkTermination1dcSignatureTest;
      };
```

```
/* ResultsMask is an integer from 0 to 6, where 0 indicates all tests were
passed. A nonzero integer indicates that measurements for the failed test are
being returned, where 1 indicates failure on the hazardous potential test, 2
indicates failure on foreign electromotive force test, 3 indicates failure on
resistive fault test, 4 indicates failure on receiver off hook test, 5 indicates
failure on presence of ringer test, and 6 indicates failure on network
termination 1 DC signature test. */"
```

12.4.2 TestActionPerformer interface

a) Replace the following:

```
"// See 9.15.1.5 for the description of the behaviour of this operation

                  void modifyResourceSelfTestSchedule(
                        in TestTrackingObjectIdType testTrackingObjectId,
                        in UserLabelType newSchedulerName)
                        raises (AccessDenied,
                              UnknownTest,
                              UnknownScheduler,
                              InvalidScheduler);

// See 9.15.1.6 for the description of the behaviour of this operation

                  void cancelScheduledResourceSelfTest (
                        in TestTrackingObjectIdType testTrackingObjectId)
                        raises (AccessDenied,
                              UnknownTest);"
```

with:

```
"// See 9.15.1.5 for the description of the behaviour of this operation

                   void modifyTestSchedule (
                        in TestTrackingObjectIdType testTrackingObjectId,
                        in UserLabelType  newSchedulerName)
                        raises (UnknownTest,
                                          UnknownScheduler,
                                          InvalidScheduler,
                                          AccessDenied);

// See 9.15.1.6 for the description of the behaviour of this operation

                   void cancelScheduledTest (
                        in TestTrackingObjectIdType testTrackingObjectId)
                        raises (UnknownTest, UncontrolledTestInProgress,
                        AccessDenied);"
```

b) Add the following operations:

```
"// See 9.15.1.17 for the description of the behaviour of this operation

DropTestResultsType metallicDropTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId)
                  raises (AccessDenied,
                        CommFailure,
                        UnknownManagedEntity);
```

```
// See 9.15.1.18 for the description of the behaviour of this operation

TestTrackingObjectIdType scheduleMetallicDropTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId,
                  in UserLabelType schedulerName)
                  raises (AccessDenied,
                        UnknownManagedEntity,
                        UnknownScheduler,
                        InvalidScheduler);


// See 9.15.1.19 for the description of the behaviour of this operation

short mACLayerTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId)
                  raises (AccessDenied,
                        CommFailure,
                        UnknownManagedEntity);

// See 9.15.1.20 for the description of the behaviour of this operation

TestTrackingObjectIdType scheduleMACLayerTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId,
                  in UserLabelType schedulerName)
                  raises (AccessDenied,
                        UnknownManagedEntity,
                        UnknownScheduler,
                        InvalidScheduler);

// See 9.15.1.21 for the description of the behaviour of this operation

boolean drawDialToneBreakTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId)
                  raises (AccessDenied,
                        CommFailure,
                        UnknownManagedEntity);


// See 9.15.1.22 for the description of the behaviour of this operation

TestTrackingObjectIdType scheduleDrawDialToneBreakTest (
                  in UserIdType testRequestorId,
                  in ManagedEntityIdType port,
                  in ServiceInstanceIdType serviceInstanceId,
                  in UserLabelType schedulerName)
                  raises (AccessDenied,
                        UnknownManagedEntity,
                        UnknownScheduler,
                        InvalidScheduler);"
```

13 Enhancements to Annex D, "Example endpoint templates"

a) After Table D.2, UNI Port Endpoints, add the following paragraph:

"For management of service connection provisioning for BPON systems supporting dynamic bandwidth assignment, all the examples under the column heading "Parameter" in Table D.2 could be augmented to include the ManagedEntityIdType of the TCONT to which the service connection belongs. The TCONT designation would appear first. This is only required in the case that T-CONT provisioning is supported explicitly."

b) At the end of Annex D, Example Endpoint Templates, add the following:

"Table D.3 illustrates examples of the type of subnetwork connections that are formed through use of the operation provisionConnection specified in Q834ServiceProvisioning::ServiceProvisioner interface.

Table D.3/Q.834.4 – Subnetwork connection types

Service	Endpoint A	Endpoint Z	Subnetwork connection
DS1	TDM DS3	TDM DS1	ds1SubnetworkConnection
DS1	ATM DS3 or OCn	TDM DS1	vcSubnetworkConnection
DS1	TDM DS3	TDM DS3	ds1SubnetworkConnection
DS3	TDM DS3	TDM DS3	ds3SubnetworkConnection
DS3	ATM DS3 or OCn	TDM DS3	vcSubnetworkConnection
Voice	ATM DS3 or OCn	RJ-11	vcSubnetworkConnection
Voice	TDM DS1	RJ-11	ds0SubnetworkConnection
Bridged LAN	ATM DS3 or OCn	Ethernet	vcSubnetworkConnection
Bridged LAN	Ethernet	Ethernet	vcSubnetworkConnection

"

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure, Internet protocol aspects and Next Generation Networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2004