INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.822.1

Amendment 1
(03/2003)

SERIES Q: SWITCHING AND SIGNALLING

Q3 interface

CORBA-based TMN performance management service

Amendment 1: Generic transport performance management

ITU-T Recommendation Q.822.1 (2001) – 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 SYSTEM No. 4	Q.120–Q.139
SPECIFICATIONS OF SIGNALLING SYSTEM No. 5	Q.140–Q.199
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–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.822.1

CORBA-based TMN performance management service

Amendment 1

Generic transport performance management

Summary

This amendment provides a new object class that can be used to represent Performance Management data for a physical port or endpoints of transport connections. This new Generic Transport PM object class is intended to be applicable across different technologies, architectures and services. The IDL model describes the new GenericTransportPmCD object class and all the interfaces associated with it.

Source

Amendment 1 to ITU-T Recommendation Q.822.1 (2001) was prepared by ITU-T Study Group 4 (2001-2004) and approved under the WTSA Resolution 1 procedure on 29 March 2003.

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.

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.822.1

CORBA-based TMN performance management service

Amendment 1

Generic transport performance management

1 Scope

This amendment provides a new object class that can be used to represent Performance Management data for a physical port or endpoints of transport connections. This new Generic Transport PM object class is intended to be applicable across different technologies, architectures and services. The IDL model describes the new GenericTransportPmCD object class and all the interfaces associated with it.

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 Q.816 (2001), *CORBA-based TMN services*.

[2] ITU-T Recommendation Q.816.1 (2001), *CORBA-based TMN services: Extensions to support coarse-grained interfaces*.

[3] ITU-T Recommendation X.780 (2001), *TMN guidelines for defining CORBA managed objects*.

[4] ITU-T Recommendation X.780.1 (2001), *TMN guidelines for defining coarse-grained CORBA managed object interfaces*.

[5] ITU-T Recommendation M.3100 (1995), *Generic network information model*, plus Amendment 2 (2000).

[6] ITU-T Recommendation Q.822.1 (2001), *CORBA-based TMN performance management service*.

[7] ANSI Standard T1.231-1997, *Digital Hierarchy – Layer 1 in-Service Digital Transmission Performance Monitoring*.

3 Overview of the Generic Transport PM Data information model

This amendment provides the definition of a new Generic Transport PM Data object class. This new object class is used for collection of performance data related to the GenericTransportTTP (the termination of a generic transport connection). GenericTransportPmCDValueType and GenericTransportPmHDValueType contain placeholders for current and history data values for counters within the physical transport protocol monitoring grouping.

The GenericTransportPmCD object class is subclassed from the CurrentData object class defined in ITU-T Recs Q.822 and Q.822.1. Inherited methods from CurrentData provide for retrieval of the SuspectFlag, retrieval of the ElapsedTime, activating or deactivating HistoryRetention, associating threshold data with a current data instance, and activating or deactivating the suppression of counters having all-zero counts. Additional inherited methods from Scanner provide for the setting of AdministrativeState, retrieval of OperationalState, and setting of Granularity Period. Similarly, GenericTransportPmHD is subclassed from the HistoryData.

For some digital signals, performance primitives in the incoming direction are reported to the far-end via special messages embedded within the signal format. Examples include: the Performance Report Message (PRM) in DS1 ESF, Far-End Block Error (FEBE) indicators in DS3 CC-bit applications, and Remote Error Indicators (REI) in SDH. With such a capability built into a transmission signal, reporting of transmission performance parameters observed at the far-end can be provided at the near-end. For this reason, the managed entity used to hold Generic Transport TTP performance management data contains placeholders for both near-end and far-end data.

The GenericTransportPmCD object attributes are arranged into six packages: near-end path, near-end line, near-end section, far-end path, far-end line, and far-end section. Only the near-end path package attributes are mandatory. Other packages may be used whenever they are deemed relevant to the underlying transport technology being modelled.

Clause 4 defines a set of CORBA IDL interfaces for the GenericTransportPmCD and GenericTransportPmHD objects. These interfaces are specified following the TMN CORBA framework and guidelines given in ITU-T Recs Q.816 and X.780 for fine-grained CORBA interface.

In addition to the fine-grained interfaces in 4.4, a companion set of Façade interfaces are defined in 4.5. These façade interfaces are defined according to the coarse-grained framework and guidelines given in ITU-T Recs Q.816.1 and X.780.1 for supporting coarse-grained CORBA interface. The name of these façade interfaces is the name of the corresponding fine-grained interface appended with "_F" (an underscore followed by a capital "F").

This amendment IDL is an integral part of ITU-T Rec. Q.822.1. This implies that all definitions (object classes, type, structure, ...) defined in ITU-T Rec. Q.822.1 are in the same IDL module and can be referenced without the module identifier.

The IDL in this amendment has been compiled successfully without syntax error. The compiler used claims CORBA 2.3 compliance, which includes value type and M4 macro capabilities.

Figures 1 and 2 show the inheritance, containment, and association relationships of the CORBA interfaces defined in this amendment. Note that façade interfaces follow the same inheritance hierarchy relationship as the corresponding fine-grained interfaces.

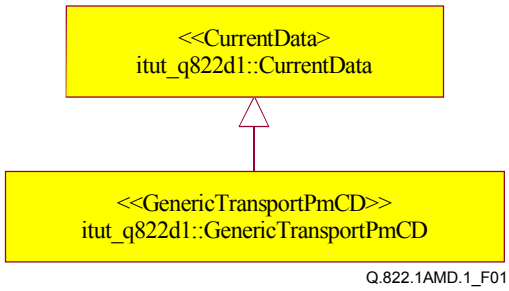

Figure 1/Q.822.1/Amd.1 – Inheritance relationship

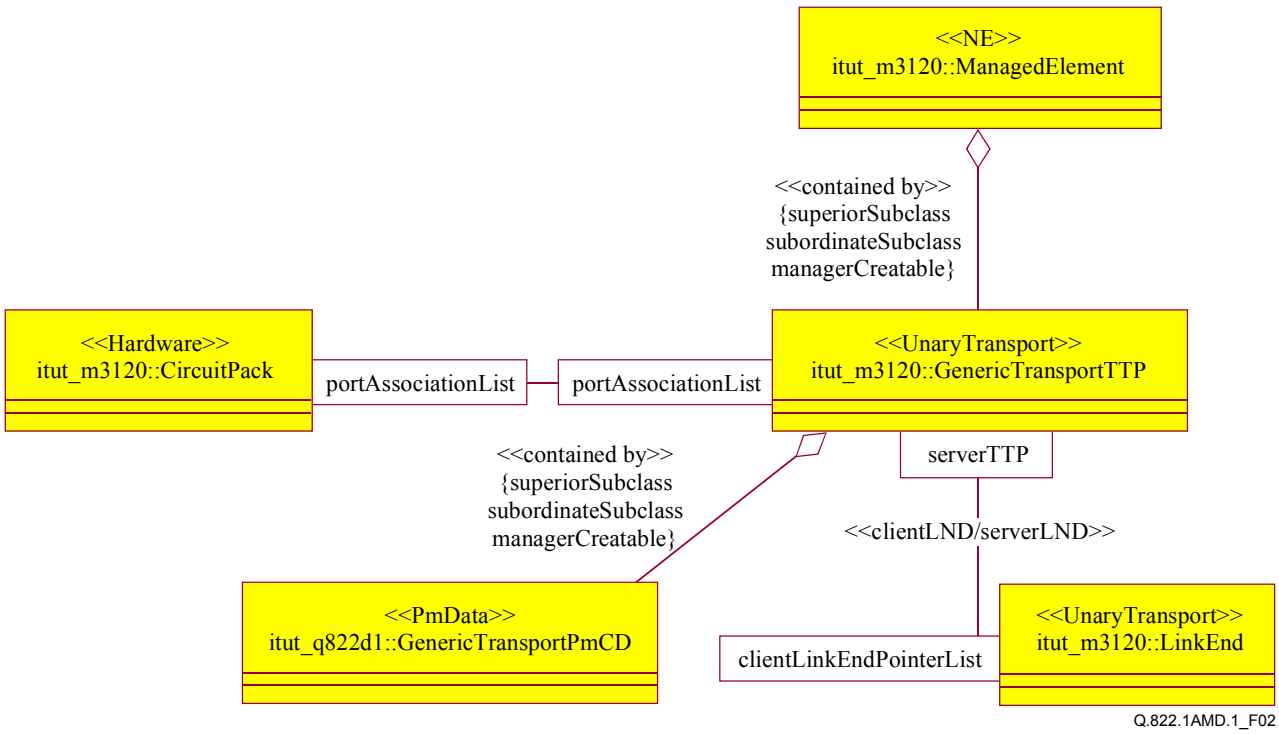

Figure 2/Q.822.1/Amd.1 – Containment and association relationships

4 Information model IDL

```
#ifndef _itut_q822d1_amd1_idl_
#define _itut_q822d1_amd1_idl_


#include <itut_q822d1.idl>


#pragma prefix "itu.int"


/**
This IDL code is intended to be stored in a file named "itut_q822d1_amd1.idl"
located in the search path used by IDL compilers on your system. The Q.822.1
main module (defined in ITU-T Rec. Q.822.1) is contained in a separate file
named
"itut_q822d1.idl"
*/




/**
This fragment is added to the module, itut_q822d1, which contains IDL definition
based on objects defined in ITU-T Rec. Q.822.
*/
module itut_q822d1
{


/**
```

4.1 Forward declarations
*/

```
/**
Interface forward declarations
*/
       interface GenericTransportPmCD;

/**
Valuetype forward declarations
*/
     valuetype GenericTransportPmCDValueType;
valuetype GenericTransportTTPHDValueType;


/**
Typedefs forward declarations
*/
     typedef MONameType GenericTransportPmCDNameType;


/**
```

4.2 Structures and typedefs

```
No new structures are defined in this amendment.
*/

/**
```

4.3 Exceptions for conditional packages
*/

```
       exception NOfarEndLinePMDataPackage {};
       exception NOfarEndPathPMDataPackage {};
       exception NOfarEndSectionPMDataPackage {};
       exception NOnearEndLinePMDataPackage {};
       exception NOnearEndSectionPMDataPackage {};



/**
```

4.4 Interfaces – Fine-grained
*/

```
/**
```

4.4.1 GenericTransportPmCD (Physical Transport Performance Monitoring Current Data)

```
This object retrieves attributes or current and history data counter values for
counters within the Physical Transport Protocol Monitoring grouping.

Additional inherited methods from Scanner provide for the setting of
AdministrativeState, retrieval of OperationalState, and setting of Granularity
Period. Inherited methods from CurrentData provide for retrieval of the
SuspectFlag, retrieval of the ElapsedTime, activating or deactivating
HistoryRetention, associating threshold data with a current data instance,
and activating or deactivating the suppression of counters having all-zero
counts.

Applicable methods from Scanner include: administrativeStateGet,
administrativeStateSet, operationalStateGet, granularityPeriodGet, and
granularityPeriodSet.
```

```
Applicable methods from CurrentData include: suspectIntervalFlagGet,
suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet,
historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet,
getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

This object should support the following notifications: objectCreation,
objectDeletion, attributeValueChange, and stateChange.

A Threshold Crossing Alert (qualityOfServiceAlarm) is used to notify the
management system when any of the performance parameters exceeds a pre-set
threshold described in the associated ThresholdData object.

A containment relationship exists between a GenericTransportTTP and its
associated GenericTransportPmCD object.
*/

valuetype GenericTransportPmCDValueType: truncatable
itut_q822d1::CurrentDataValueType
     {

/**
The codingViolationsPath parameter is used as a count of certain error events
occurring in the accumulation period. Sample error events include Frame
Synchronization and CRC errors for DS1 links, or P-bit and CP-bit parity errors
for DS3 links.
*/
               public unsigned long codingViolationsPath;
                       // GET
/**
The erroredSecondsPath parameter is a count of 1-second intervals containing
path errors. Sample path errors include CRC-6 errors (DS1), Severely-Errored
Frame defects (DS1/DS3), and P-bit Parity errors (DS3).
*/
               public unsigned long erroredSecondsPath;
                       // GET
/**
The severelyErroredSecondspath parameter is a count of 1-second intervals with X
or more path error events, or one or more Loss of Signal defect.
*/
               public unsigned long severelyErroredSecondsPath;
                       // GET
/**
The unavailableSecondsPath parameter is a count of 1-second intervals during
which the path is unavailable.
*/
               public unsigned long unavailableSecondsPath;
                       // GET
               public unsigned long failureCounterPath;
                       // GET
               public unsigned long farEndCodingViolationsPath;
                       // GET
                       // farEndPathPMDataPackage
                       // Conditional, present if the instance supports it
               public unsigned long farEndErroredSecondsPath;
                       // GET
                       // farEndPathPMDataPackage
                       // Conditional, present if the instance supports it
               public unsigned long farEndSeverelyErroredSecondsPath;
                       // GET
                       // farEndPathPMDataPackage
                       // Conditional, present if the instance supports it
```

```
                public unsigned long farEndUnavailableSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndFailureCounterPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
/**
The codingViolationsLine parameter is count of certain error events occurring in
the accumulation period. Sample error events include Bipolar Violations (BPVs)
and Excessive Zeros (EXZs) occurring over a DS1/DS3 link.
*/
                public unsigned long codingViolationsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
An erroredSecondsLine is a count of 1-second intervals in which one or more Line
Coding Violation error events were detected.
*/
                public unsigned long erroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
severelyErroredSecondsLine is a count of 1-second intervals with X or more BPVs
plus EXZs, or one or more Loss of Signal defect.
*/
                public unsigned long severelyErroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
The lossOfSignalLine parameter is a count of 1-second intervals containing one
or more Loss of Signal defects.
*/
                public unsigned long lossOfSignalLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long failureCounterLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndCodingViolationsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndSeverelyErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndlossOfSignalLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
```

```
        public unsigned long farEndFailureCounterLine;
                // GET
                // farEndLinePMDataPackage
                // Conditional, present if the instance supports it
        public unsigned long codingViolationsSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long erroredSecondsSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long severelyErroredSecondsSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long lossOfSignalSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long failureCounterSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndCodingViolationsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndErroredSecondsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndSeverelyErroredSecondsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndlossOfSignalSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndFailureCounterSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

    }; // valuetype GenericTransportPmCDValueType



valuetype GenericTransportTTPHDValueType: truncatable
itut_q822d1::HistoryDataValueType
    {
```

```
/**
The codingViolationsPath parameter is used as a count of certain error events
occurring in the accumulation period. Sample error events include Frame
Synchronization and CRC errors for DS1 links, or P-bit and CP-bit parity errors
for DS3 links.
*/
              public unsigned long codingViolationsPath;
                      // GET
/**
The erroredSecondsPath parameter is a count of 1-second intervals containing
path errors. Sample path errors include CRC-6 errors (DS1), Severely-Errored
Frame defects (DS1/DS3), and P-bit Parity errors (DS3).
*/
              public unsigned long erroredSecondsPath;
                      // GET
/**
The severelyErroredSecondspath parameter is a count of 1-second intervals with X
or more path error events, or one or more Loss of Signal defect.
*/
              public unsigned long severelyErroredSecondsPath;
                      // GET
/**
The unavailableSecondsPath parameter is a count of 1-second intervals during
which the path is unavailable.
*/
              public unsigned long unavailableSecondsPath;
                      // GET
              public unsigned long failureCounterPath;
                      // GET
              public unsigned long farEndCodingViolationsPath;
                      // GET
                      // farEndPathPMDataPackage
                      // Conditional, present if the instance supports it
              public unsigned long farEndErroredSecondsPath;
                      // GET
                      // farEndPathPMDataPackage
                      // Conditional, present if the instance supports it
              public unsigned long farEndSeverelyErroredSecondsPath;
                      // GET
                      // farEndPathPMDataPackage
                      // Conditional, present if the instance supports it
              public unsigned long farEndUnavailableSecondsPath;
                      // GET
                      // farEndPathPMDataPackage
                      // Conditional, present if the instance supports it
              public unsigned long farEndFailureCounterPath;
                      // GET
                      // farEndPathPMDataPackage
                      // Conditional, present if the instance supports it
/**
The codingViolationsLine parameter is count of certain error events occurring in
the accumulation period. Sample error events include Bipolar Violations (BPVs)
and Excessive Zeros (EXZs) occurring over a DS1/DS3 link.
*/
              public unsigned long codingViolationsLine;
                      // GET
                      // nearEndLinePMDataPackage
                      // Conditional, present if the instance supports it
/**
An erroredSecondsLine is a count of 1-second intervals in which one or more Line
Coding Violation error events were detected.
```

```
*/
                public unsigned long erroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
severelyErroredSecondsLine is a count of 1-second intervals with X or more BPVs
plus EXZs, or one or more Loss of Signal defect.
*/
                public unsigned long severelyErroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
The lossOfSignalLine parameter is a count of 1-second intervals containing one
or more Loss of Signal defects.
*/
                public unsigned long lossOfSignalLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long failureCounterLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndCodingViolationsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndSeverelyErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndlossOfSignalLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndFailureCounterLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long codingViolationsSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long erroredSecondsSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long severelyErroredSecondsSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long lossOfSignalSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections
```

```
        public unsigned long failureCounterSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndCodingViolationsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndErroredSecondsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndSeverelyErroredSecondsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndlossOfSignalSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndFailureCounterSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections


};  // valuetype GenericTransportTTPHDValueType

interface GenericTransportPmCD: itut_q822d1::CurrentData
{

        unsigned long codingViolationsPathGet()
                raises (itut_x780::ApplicationError);

        unsigned long erroredSecondsPathGet()
                raises (itut_x780::ApplicationError);

        unsigned long severelyErroredSecondsPathGet()
                raises (itut_x780::ApplicationError);

        unsigned long unavailableSecondsPathGet()
                raises (itut_x780::ApplicationError);

        unsigned long failureCounterPathGet()
                raises (itut_x780::ApplicationError);

        unsigned long farEndCodingViolationsPathGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndPathPMDataPackage);

        unsigned long farEndErroredSecondsPathGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndPathPMDataPackage);

        unsigned long farEndSeverelyErroredSecondsPathGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndPathPMDataPackage);
```

```
unsigned long farEndUnavailableSecondsPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long farEndFailureCounterPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long codingViolationsLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long erroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long severelyErroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long lossOfSignalLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long failureCounterLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long farEndCodingViolationsLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);

unsigned long farEndErroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);

unsigned long farEndSeverelyErroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);

unsigned long farEndlossOfSignalLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);

unsigned long farEndFailureCounterLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);

unsigned long codingViolationsSectionGet()
        raises (itut_x780::ApplicationError,
                NOnearEndSectionPMDataPackage);

unsigned long erroredSecondsSectionGet()
        raises (itut_x780::ApplicationError,
                NOnearEndSectionPMDataPackage);

unsigned long severelyErroredSecondsSectionGet()
        raises (itut_x780::ApplicationError,
                NOnearEndSectionPMDataPackage);

unsigned long lossOfSignalSectionGet()
        raises (itut_x780::ApplicationError,
                NOnearEndSectionPMDataPackage);
```

```
            unsigned long failureCounterSectionGet()
                    raises (itut_x780::ApplicationError,
                            NOnearEndSectionPMDataPackage);

            unsigned long farEndCodingViolationsSectionGet()
                    raises (itut_x780::ApplicationError,
                            NOfarEndSectionPMDataPackage);

            unsigned long farEndErroredSecondsSectionGet()
                    raises (itut_x780::ApplicationError,
                            NOfarEndSectionPMDataPackage);

            unsigned long farEndSeverelyErroredSecondsSectionGet()
                    raises (itut_x780::ApplicationError,
                            NOfarEndSectionPMDataPackage);

            unsigned long farEndlossOfSignalSectionGet()
                    raises (itut_x780::ApplicationError,
                            NOfarEndSectionPMDataPackage);

            unsigned long farEndFailureCounterSectionGet()
                    raises (itut_x780::ApplicationError,
                            NOfarEndSectionPMDataPackage);

    }; // interface GenericTransportPmCD



    interface GenericTransportPmCDFactory: itut_x780::ManagedObjectFactory
    {
         itut_x780::ManagedObject create
               (in NameBindingType nameBinding,
                     // module name containing Name Binding info.
               in MONameType superior,
                     // Name of containing object.
               in string reqID,
                     // Requested ID value for name, will be
                     // empty if auto-naming is to be used.
               out MONameType name,
                     // Entire name of newly created object.
               in StringSetType packageNameList,
                     // List of packages requested.

               in short historyRetention,
               in AdministrativeStateType administrativeState,
               in TimePeriodType granularityPeriod,
               in MONameSetType thresholdDataInstanceList,
               in short maxSuppressedIntervals,
               in GeneralizedTimeType periodSynchronizationTime)
               raises (itut_x780::ApplicationError,
                     itut_x780::CreateError);

    }; // interface GenericTransportPmCDFactory

/**
```

4.5 Interfaces – Façade

The behaviour of the façade interfaces are identical to the corresponding fine-grained interfaces. Therefore, comments are not included in the façade interfaces. Readers are referred to the fine-grained interface in 4.4 for the behaviour of the façade interface.

```
This clause can be omitted from IDL if a management system only supports
fine-grained interface.
*/


/**
```

4.5.1 GenericTransportPmCD_F

```
*/
     interface GenericTransportPmCD_F: itut_q822d1::CurrentData_F
     {

               unsigned long codingViolationsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError);

               unsigned long erroredSecondsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError);

               unsigned long severelyErroredSecondsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError);

               unsigned long unavailableSecondsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError);

               unsigned long failureCounterPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError);

               unsigned long farEndCodingViolationsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError,
                                NOfarEndPathPMDataPackage);

               unsigned long farEndErroredSecondsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError,
                                NOfarEndPathPMDataPackage);

               unsigned long farEndSeverelyErroredSecondsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError,
                                NOfarEndPathPMDataPackage);

               unsigned long farEndUnavailableSecondsPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError,
                                NOfarEndPathPMDataPackage);

               unsigned long farEndFailureCounterPathGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError,
                                NOfarEndPathPMDataPackage);

               unsigned long codingViolationsLineGet
                         (in MONameType name)
                         raises (itut_x780::ApplicationError,
                                NOnearEndLinePMDataPackage);
```

```
unsigned long erroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long severelyErroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long lossOfSignalLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long failureCounterLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long farEndCodingViolationsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long farEndErroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long farEndSeverelyErroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long farEndlossOfSignalLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long farEndFailureCounterLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long codingViolationsSectionGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndSectionPMDataPackage);

unsigned long erroredSecondsSectionGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndSectionPMDataPackage);

unsigned long severelyErroredSecondsSectionGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndSectionPMDataPackage);

unsigned long lossOfSignalSectionGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndSectionPMDataPackage);
```

```
                unsigned long failureCounterSectionGet
                          (in MONameType name)
                          raises (itut_x780::ApplicationError,
                                 NOnearEndSectionPMDataPackage);

                unsigned long farEndCodingViolationsSectionGet
                          (in MONameType name)
                          raises (itut_x780::ApplicationError,
                                 NOfarEndSectionPMDataPackage);

                unsigned long farEndErroredSecondsSectionGet
                          (in MONameType name)
                          raises (itut_x780::ApplicationError,
                                 NOfarEndSectionPMDataPackage);

                unsigned long farEndSeverelyErroredSecondsSectionGet
                          (in MONameType name)
                          raises (itut_x780::ApplicationError,
                                 NOfarEndSectionPMDataPackage);

                unsigned long farEndlossOfSignalSectionGet
                          (in MONameType name)
                          raises (itut_x780::ApplicationError,
                                 NOfarEndSectionPMDataPackage);

                unsigned long farEndFailureCounterSectionGet
                          (in MONameType name)
                          raises (itut_x780::ApplicationError,
                                 NOfarEndSectionPMDataPackage);

     }; // interface GenericTransportPmCD_F
/**
```

4.6 Name binding

```
The following module contains name binding information.
*/
     module NameBinding
     {

/**
```

4.6.1 GenericTransportPmCD

```
*/
               module GenericTransportPmCD_GenericTransportTTP
               {
                     const string   superiorClass =
                               "itut_m3120:: GenericTransportTTP";
                     const boolean  superiorSubclassesAllowed = TRUE;
                     const string   subordinateClass =
                               "itut_m3120:: GenericTransportPmCD";
                     const boolean  subordinateSubclassesAllowed = TRUE;
                     const boolean  managerCreatesAllowed = TRUE;
                     const DeletePolicyType deletePolicy =
                               itut_x780::deleteOnlyIfNoContainedObjects;
                     const string   kind = "GenericTransportPmCD";
               }; // module GenericTransportPmCD_GenericTransportTTP

     }; // module NameBinding

}; // module itut_q822d1

#endif // _itut_q822d1_amd1_idl_
```

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 and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2003