



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Q.822.1

Enmienda 1
(03/2003)

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN

Interfaz Q3

Servicio de gestión de la calidad de funcionamiento
de la RGT basado en arquitectura de intermediario
de petición de objeto común

**Enmienda 1: Gestión de la calidad de
funcionamiento del transporte genérico**

Recomendación UIT-T Q.822.1 (2001) – Enmienda 1

RECOMENDACIONES UIT-T DE LA SERIE Q
CONMUTACIÓN Y SEÑALIZACIÓN

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLOTACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 4	Q.120–Q.139
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 5	Q.140–Q.199
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 7	Q.700–Q.799
INTERFAZ Q3	Q.800–Q.849
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.º 1	Q.850–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1699
REQUISITOS Y PROTOCOLOS DE SEÑALIZACIÓN PARA IMT-2000	Q.1700–Q.1799
ESPECIFICACIONES DE LA SEÑALIZACIÓN RELACIONADA CON EL CONTROL DE LLAMADA INDEPENDIENTE DEL PORTADOR	Q.1900–Q.1999
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

Para más información, véase la Lista de Recomendaciones del UIT-T.

Recomendación UIT-T Q.822.1

Servicio de gestión de la calidad de funcionamiento de la RGT basado en arquitectura de intermediario de petición de objeto común

Enmienda 1

Gestión de la calidad de funcionamiento del transporte genérico

Resumen

En esta enmienda se describe una nueva clase de objetos que se puede usar para representar los datos de gestión de la calidad de funcionamiento (PM) de puertos o puntos de extremo físicos de las conexiones de transporte. Se prevé que esta nueva clase de objetos para la gestión de la calidad de funcionamiento del transporte genérico se pueda aplicar en diferentes tecnologías, arquitecturas y servicios. El modelo IDL (lenguaje de definición de interfaz) describe la nueva clase de objetos GenericTransportPmCD y todas sus interfaces asociadas.

Orígenes

La enmienda 1 a la Recomendación UIT-T Q.822.1 (2001), preparada por la Comisión de Estudio 4 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 29 de marzo de 2003.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2003

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

ÍNDICE

	Página
1 Alcance	1
2 Referencias	1
3 Introducción al modelo de información de datos para la gestión de la calidad de funcionamiento de transporte genérico.....	2
4 IDL del modelo de información	3
4.1 Declaraciones previas	4
4.2 Estructuras y definiciones de tipo Typedefs.....	4
4.3 Excepciones aplicables a los lotes condicionales.....	4
4.4 Interfaces – Granularidad fina	5
4.5 Interfaces – Fachada	13
4.6 Vinculación de nombre.....	16

Recomendación UIT-T Q.822.1

Servicio de gestión de la calidad de funcionamiento de la RGT basado en arquitectura de intermediario de petición de objeto común

Enmienda 1

Gestión de la calidad de funcionamiento del transporte genérico

1 Alcance

En esta enmienda se describe una nueva clase de objetos que se puede usar para representar los datos de gestión de la calidad de funcionamiento (PM, *performance management*) de puertos o puntos de extremo físicos de las conexiones de transporte. Se prevé que esta nueva clase de objetos para la gestión de la calidad de funcionamiento del transporte genérico se pueda aplicar en diferentes tecnologías, arquitecturas y servicios. El modelo IDL (lenguaje de definición de interfaz) describe la nueva clase de objetos GenericTransportPmCD y todas sus interfaces asociadas.

2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- [1] Recomendación UIT-T Q.816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común*.
- [2] Recomendación UIT-T Q.816.1 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común: Extensiones para el soporte de interfaces de granularidad gruesa*.
- [3] Recomendación UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común*.
- [4] Recomendación UIT-T X.780.1 (2001), *Directrices de la RGT para la definición de interfaces de objetos gestionados mediante arquitectura de intermediario de petición de objeto común de granularidad gruesa*.
- [5] Recomendación UIT-T M.3100 (1995), *Modelo genérico de información de red*, más enmienda 2 (2000).
- [6] Recomendación UIT-T Q.822.1 (2001), *Servicio de gestión de la calidad de funcionamiento de la RGT basado en arquitectura de intermediario de petición de objeto común*.
- [7] ANSI Standard T1.231-1997, *Digital Hierarchy – Layer 1 in-Service Digital Transmission Performance Monitoring*.

3 Introducción al modelo de información de datos para la gestión de la calidad de funcionamiento de transporte genérico

En esta enmienda se define una nueva clase de objetos de datos para la gestión de la calidad de funcionamiento de transporte genérico. Esta nueva clase de objetos se utiliza para recopilar datos de calidad de funcionamiento relacionados con la clase `GenericTransportTTP` (la terminación de una conexión de transporte genérico). Las clases `GenericTransportPmCDValueType` y `GenericTransportPmHDValueType` contienen marcadores de posición para los valores de datos actuales y anteriores que utilizan los contadores en el conjunto de supervisión del protocolo de transporte físico.

La clase de objeto `GenericTransportPmCD` es una subclase de la clase de objeto `CurrentData` definida en las Recomendaciones UIT-T Q.822 y Q.822.1. Los métodos heredados de `CurrentData` proporcionan los mecanismos necesarios para recuperar el `SuspectFlag`, recuperar el `ElapsedTime`, activar o desactivar `HistoryRetention`, asociar datos de umbral al actual ejemplar de datos, y activar o desactivar la supresión de contadores cuyo valor sea todo ceros. Otros métodos heredados de `Scanner` proporcionan lo necesario para configurar el `AdministrativeState`, recuperar el `OperationalState`, y configurar el periodo de granularidad. Asimismo, la clase `GenericTransportPmHD` es una subclase de `HistoryData`.

Para ciertas señales digitales, las primitivas de calidad de funcionamiento en el sentido entrante se comunican al extremo distante mediante mensajes especiales incorporados en el formato de la señal. Por ejemplo, el mensaje de notificación de calidad (PRM, *performance report message*) en ESF DS1, los indicadores de bloque con errores en el extremo distante (FEBE, *far-end block error*) en aplicaciones CC-bit DS3 e indicadores de errores distantes (REI, *remote error indicator*) en SDH. Con esa capacidad incorporada en la señal de transmisión, se puede informar en el extremo cercano de los parámetros de calidad de funcionamiento de la transmisión observados en el extremo distante. Por esta razón, la entidad gestionada que se utiliza para guardar los datos de gestión de la calidad de funcionamiento TTP del transporte genérico contiene marcadores de posiciones para el extremo cercano y el extremo distante.

Los atributos del objeto `GenericTransportPmCD` se clasifican en seis lotes: trayecto de extremo cercano, línea de extremo cercano, sección de extremo cercano, trayecto de extremo distante, línea de extremo distante y sección de extremo distante. Los únicos atributos obligatorios son los del lote trayecto de extremo cercano. Los demás lotes se pueden utilizar siempre que se consideren pertinentes para la tecnología de transporte subyacente que se está modelizando.

En la cláusula 4 se define el conjunto de interfaces IDL CORBA para los objetos `GenericTransportPmCD` y `GenericTransportPmHD`. Estas interfaces están especificados según el marco y las directrices CORBA RGT definidas en las Recomendaciones UIT-T Q.816 y X.780 para interfaces CORBA de granularidad fina.

Además de las interfaces de granularidad fina definidas en 4.4, se define también un conjunto de interfaces fachada en 4.5. Estas interfaces fachada se definen de conformidad con el marco y las directrices de granularidad gruesa definidas en las Recomendaciones UIT-T Q.816.1 y X.780.1 para el soporte de la interfaz CORBA de granularidad gruesa. Los nombres de estas interfaces fachada son los nombres de las correspondientes interfaces de granularidad fina con el sufijo "_F" (una línea de subrayado seguida de la letra en mayúsculas "F").

Esta enmienda al IDL forma parte integral de la Rec. UIT-T Q.822.1. Esto significa que todas las definiciones (clases de objetos, tipos, estructura, ...) definidas en la Rec. UIT-T Q.822.1 están en el mismo módulo IDL y se puede hacer referencia a ellas sin tener que especificar el identificador del módulo.

El IDL en esta enmienda se ha compilado correctamente y sin errores de sintaxis. El compilador utilizado era conforme con la especificación CORBA 2.3, que incluye capacidades de tipo de valor y macros M4.

En las figuras 1 y 2 se muestran las relaciones de herencia, contención y asociación de las interfaces CORBA definidas en esta enmienda. Obsérvese que las interfaces fachada tienen la misma relación jerárquica de herencia que las correspondientes interfaces de granularidad fina.

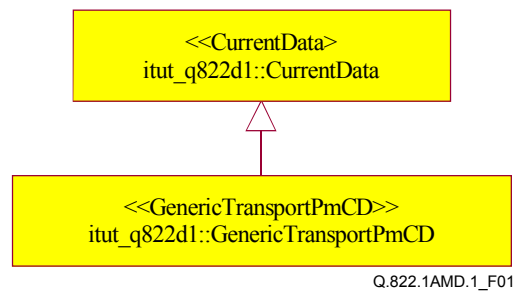


Figura 1/Q.822.1/Enm.1 – Relación de herencia

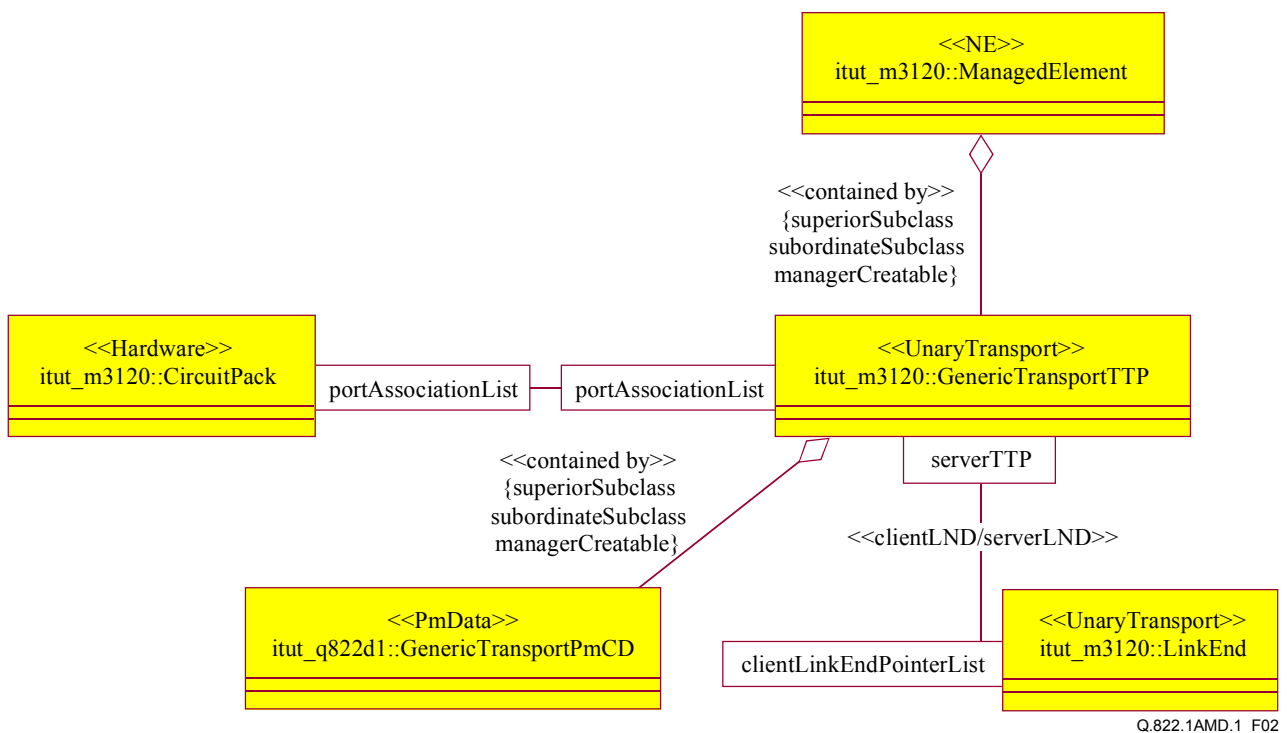


Figura 2/Q.822.1/Enm.1 – Relaciones de contención y asociación

4 IDL del modelo de información

```
#ifndef _itut_q822d1_amd1_idl_
#define _itut_q822d1_amd1_idl_

#include <itut_q822d1.idl>

#pragma prefix "itu.int"
```

```
/**
Este código IDL deberá almacenarse en un archivo denominado
"itut_q822d1_amd1.idl" situado en el directorio de búsqueda que utilizan los
compiladores IDL en su sistema. El módulo principal Q.822.1 (definido en la
Rec. UIT-T Q.822.1) estará en un archivo aparte denominado
"itut_q822d1.idl"
*/
```

```
/**
Este fragmento se añade al módulo itut_q822d1, que contiene la definición IDL
elaborada a partir de los objetos definidos en la Rec. UIT-T Q.822.
*/
module itut_q822d1
{
```

```
/**
```

4.1 Declaraciones previas

```
*/
```

```
/**
```

```
Declaraciones previas de la interfaz
```

```
*/
```

```
interface GenericTransportPmCD;
```

```
/**
```

```
Declaraciones previas Valuetype
```

```
*/
```

```
valuetype GenericTransportPmCDValueType;
```

```
valuetype GenericTransportTTPHDValueType;
```

```
/**
```

```
Declaraciones previas Typedefs
```

```
*/
```

```
typedef MOnameType GenericTransportPmCDNameType;
```

```
/**
```

4.2 Estructuras y definiciones de tipo Typedefs

```
En esta enmienda no se definen nuevas estructuras
```

```
*/
```

```
/**
```

4.3 Excepciones aplicables a los lotes condicionales

```
*/
```

```
exception NOfarEndLinePMDDataPackage {};
```

```
exception NOfarEndPathPMDDataPackage {};
```

```
exception NOfarEndSectionPMDDataPackage {};
```

```
exception NOnearEndLinePMDDataPackage {};
```

```
exception NOnearEndSectionPMDDataPackage {};
```

```
/**
```

4.4 Interfaces – Granularidad fina

*/

/**

4.4.1 GenericTransportPmCD (Datos actuales de supervisión de la calidad de funcionamiento del transporte físico)

Este objeto recupera los atributos o los valores de contadores de datos actuales y anteriores en el conjunto de supervisión del protocolo de transporte físico.

Otros métodos heredados de Scanner proporcionan lo necesario para configurar el AdministrativeState, recuperar el OperationalState, y configurar el periodo de granularidad. Los métodos heredados de CurrentData proporcionan lo necesario para recuperar el SuspectFlag, el ElapsedTime, activar o desactivar HistoryRetention, asociar datos de umbral a las instancias de datos actuales y activar o desactivar la supresión de los contadores cuyo valor sea todo ceros.

Algunos de los métodos aplicables heredados de Scanner son:

administrativeStateGet, administrativeStateSet, operationalStateGet, granularityPeriodGet, y granularityPeriodSet.

Algunos de los métodos aplicables heredados de CurrentData son:

suspectIntervalFlagGet, suspectIntervalFlagDefaultSet, elapsedTimeGet, historyRetentionGet, historyRetentionSet, thresholdDataInstanceListGet, thresholdDataInstanceListSet, getMostRecent, getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet, maxSuppressedIntervalsSet.

Este objeto debería soportar las notificaciones siguientes: objectCreation, objectDeletion, attributeValueChange, y stateChange.

Se utiliza una alerta de atravesamiento de umbral (qualityOfServiceAlarm) para notificar al sistema de gestión cuando alguno de los parámetros de calidad de funcionamiento sobrepasa el valor de umbral predeterminado descrito en el objeto ThresholdData asociado.

Existe una relación de contención entre el GenericTransportTTP y su objeto GenericTransportPmCD asociado.

*/

```
valuetype GenericTransportPmCDValueType: truncatable
itut_q822d1::CurrentDataValueType
{
```

/**

El parámetro codingViolationsPath se utiliza como un contador de ciertos eventos de error que se producen en el periodo de acumulación. Los eventos de errores de muestra incluyen errores de sincronización de trama y errores CRC de enlace DS1, o errores de paridad P-bit y CP-bit de los enlaces DS3.

*/

```
    public unsigned long codingViolationsPath;
        // GET
```

/**

El parámetro erroredSecondsPath es un contador de intervalos de 1 segundo que contiene los errores del trayecto. Los errores del trayecto de muestra incluyen errores CRC-6 (DS1), defectos de trama con muchos errores (DS1/DS3), y errores de paridad P-bit (DS3).

*/

```
    public unsigned long erroredSecondsPath;
        // GET
```

/**

El parámetro severelyErroredSecondsPath es un contador de intervalos de 1 segundo con X o más eventos de error de trayecto, o algún defecto de pérdida de señal.

```
*/  
        public unsigned long severelyErroredSecondsPath;  
            // GET
```

/**

El parámetro unavailableSecondsPath es un contador de intervalos de 1 segundo durante los periodos en que el trayecto no está disponible.

```
*/  
        public unsigned long unavailableSecondsPath;  
            // GET  
        public unsigned long failureCounterPath;  
            // GET  
        public unsigned long farEndCodingViolationsPath;  
            // GET  
            // farEndPathPMDDataPackage  
            // Conditional, present if the instance supports it  
        public unsigned long farEndErroredSecondsPath;  
            // GET  
            // farEndPathPMDDataPackage  
            // Conditional, present if the instance supports it  
        public unsigned long farEndSeverelyErroredSecondsPath;  
            // GET  
            // farEndPathPMDDataPackage  
            // Conditional, present if the instance supports it  
        public unsigned long farEndUnavailableSecondsPath;  
            // GET  
            // farEndPathPMDDataPackage  
            // Conditional, present if the instance supports it  
        public unsigned long farEndFailureCounterPath;  
            // GET  
            // farEndPathPMDDataPackage  
            // Conditional, present if the instance supports it
```

/**

El parámetro codingViolationsLine es un contador de ciertos eventos de error que se producen en el periodo de acumulación. Los eventos de error de muestra incluyen las violaciones bipolares (BPV) y los ceros excesivos (EXZ) que se producen en un enlace DS1/DS3.

```
*/  
        public unsigned long codingViolationsLine;  
            // GET  
            // nearEndLinePMDDataPackage  
            // Conditional, present if the instance supports it
```

/**

erroredSecondsLine es un contador de intervalos de 1 segundo en los cuales se detectó algún evento de error de violación de la codificación de línea.

```
*/  
        public unsigned long erroredSecondsLine;  
            // GET  
            // nearEndLinePMDDataPackage  
            // Conditional, present if the instance supports it
```

/**

severelyErroredSecondsLine es un contador de intervalos de 1 segundo con X o más BPV y EXZ, o algún defecto de pérdida de señal.

```
*/  
        public unsigned long severelyErroredSecondsLine;  
            // GET  
            // nearEndLinePMDDataPackage  
            // Conditional, present if the instance supports it
```

/**

El parámetro lossOfSignalLine es un contador de intervalos de 1 segundo que contiene algún defecto de pérdida de señal.

```
*/
```

```

public unsigned long lossOfSignalLine;
    // GET
    // nearEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long failureCounterLine;
    // GET
    // nearEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndCodingViolationsLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndErroredSecondsLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndSeverelyErroredSecondsLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndlossOfSignalLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndFailureCounterLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long codingViolationsSection;
    // GET
    // nearEndSectionPMDDataPackage
    // Conditional, present if the instance supports sections

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

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

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

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

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

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

```

```

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

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

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

}; // valuetype GenericTransportPmCDValueType

```

```

valuetype GenericTransportTTPHDValueType: truncatable
itut_q822d1::HistoryDataValueType
{

```

```
/**
```

El parámetro codingViolationsPath se utiliza como un contador de ciertos eventos de error que se producen en el periodo de acumulación. Los eventos de error de muestra incluyen errores de sincronización de trama y errores CRC de enlaces DS1, o errores de paridad P-bit y CP-bit de enlaces DS3.

```
*/
```

```

public unsigned long codingViolationsPath;
    // GET

```

```
/**
```

El parámetro erroredSecondsPath es un contador de intervalos de 1 segundo que contienen errores de trayecto. Los errores del trayecto de muestra incluyen errores CRC-6 (DS1), defectos de tramas con muchos errores (DS1/DS3), y errores de paridad P-bit (DS3).

```
*/
```

```

public unsigned long erroredSecondsPath;
    // GET

```

```
/**
```

El parámetro severelyErroredSecondspath es un contador de intervalos de 1 segundo con X o más eventos de error de trayecto o algún defecto de pérdida de señal.

```
*/
```

```

public unsigned long severelyErroredSecondsPath;
    // GET

```

```
/**
```

El parámetro unavailableSecondsPath es un contador de intervalos de 1 segundo durante el cual el trayecto no está disponible.

```
*/
```

```

public unsigned long unavailableSecondsPath;
    // GET
public unsigned long failureCounterPath;
    // GET
public unsigned long farEndCodingViolationsPath;
    // GET
    // farEndPathPMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndErroredSecondsPath;
    // GET
    // farEndPathPMDDataPackage
    // Conditional, present if the instance supports it

```

```

public unsigned long farEndSeverelyErroredSecondsPath;
    // GET
    // farEndPathPMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndUnavailableSecondsPath;
    // GET
    // farEndPathPMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndFailureCounterPath;
    // GET
    // farEndPathPMDDataPackage
    // Conditional, present if the instance supports it
/**
El parámetro codingViolationsLine es un contador de ciertos eventos de error que
se producen en el periodo de acumulación. Los eventos de error de muestra
incluyen violaciones bipolares (BPV) y ceros excesivos (EXZ) que se producen en
un enlace DS1/DS3.
*/
    public unsigned long codingViolationsLine;
        // GET
        // nearEndLinePMDDataPackage
        // Conditional, present if the instance supports it
/**
erroredSecondsLine es un contador de intervalos de un segundo en el cual se ha
detectado algún evento de error de violación de codificación de línea.
*/
    public unsigned long erroredSecondsLine;
        // GET
        // nearEndLinePMDDataPackage
        // Conditional, present if the instance supports it
/**
severelyErroredSecondsLine es un contador de intervalos de un segundo con X o
más BPV y EXZ, o algún defecto de pérdida de señal.
*/
    public unsigned long severelyErroredSecondsLine;
        // GET
        // nearEndLinePMDDataPackage
        // Conditional, present if the instance supports it
/**
El parámetro lossOfSignalLine es un contador de intervalos de un segundo que
contienen algún defecto de pérdida de señal.
*/
    public unsigned long lossOfSignalLine;
        // GET
        // nearEndLinePMDDataPackage
        // Conditional, present if the instance supports it
public unsigned long failureCounterLine;
    // GET
    // nearEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndCodingViolationsLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndErroredSecondsLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndSeverelyErroredSecondsLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it

```

```

public unsigned long farEndlossOfSignalLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long farEndFailureCounterLine;
    // GET
    // farEndLinePMDDataPackage
    // Conditional, present if the instance supports it
public unsigned long codingViolationsSection;
    // GET
    // nearEndSectionPMDDataPackage
    // Conditional, present if the instance supports sections

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

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

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

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

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

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

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

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

public unsigned long farEndFailureCounterSection;
    // GET
    // farEndSectionPMDDataPackage
    // 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,
            NOfarEndPathPMDDataPackage);

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

    unsigned long farEndSeverelyErroredSecondsPathGet()
        raises (itut_x780::ApplicationError,
            NOfarEndPathPMDDataPackage);

    unsigned long farEndUnavailableSecondsPathGet()
        raises (itut_x780::ApplicationError,
            NOfarEndPathPMDDataPackage);

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

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

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

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

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

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

    unsigned long farEndCodingViolationsLineGet()
        raises (itut_x780::ApplicationError,
            NOfarEndLinePMDDataPackage);
}

```

```

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

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

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

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

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

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

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

unsigned long lossOfSignalSectionGet()
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

unsigned long failureCounterSectionGet()
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

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

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

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

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

unsigned long farEndFailureCounterSectionGet()
    raises (itut_x780::ApplicationError,
           NOfarEndSectionPMDDataPackage);
}; // 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 – Fachada

El funcionamiento de las interfaces fachada es idéntico al de las correspondientes interfaces de granularidad fina. Por consiguiente, en las interfaces fachada no se incluyen comentarios. Se aconseja que los lectores se remitan a la interfaz de granularidad fina 4.4 para obtener más información sobre el funcionamiento de la interfaz fachada.

Esta cláusula se puede omitir del IDL cuando los sistemas de gestión sólo soportan interfaces de granularidad fina.

*/

/**

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,
            NOfarEndPathPMDDataPackage);

unsigned long farEndErroredSecondsPathGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOfarEndPathPMDDataPackage);

unsigned long farEndSeverelyErroredSecondsPathGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOfarEndPathPMDDataPackage);

unsigned long farEndUnavailableSecondsPathGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOfarEndPathPMDDataPackage);

unsigned long farEndFailureCounterPathGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOfarEndPathPMDDataPackage);

unsigned long codingViolationsLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOnearEndLinePMDDataPackage);

unsigned long erroredSecondsLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOnearEndLinePMDDataPackage);

unsigned long severelyErroredSecondsLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOnearEndLinePMDDataPackage);

unsigned long lossOfSignalLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOnearEndLinePMDDataPackage);

unsigned long failureCounterLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOnearEndLinePMDDataPackage);

unsigned long farEndCodingViolationsLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOfarEndLinePMDDataPackage);

unsigned long farEndErroredSecondsLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
            NOfarEndLinePMDDataPackage);

```

```

unsigned long farEndSeverelyErroredSecondsLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndLinePMDDataPackage);

unsigned long farEndlossOfSignalLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndLinePMDDataPackage);

unsigned long farEndFailureCounterLineGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndLinePMDDataPackage);

unsigned long codingViolationsSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

unsigned long erroredSecondsSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

unsigned long severelyErroredSecondsSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

unsigned long lossOfSignalSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

unsigned long failureCounterSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOnearEndSectionPMDDataPackage);

unsigned long farEndCodingViolationsSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndSectionPMDDataPackage);

unsigned long farEndErroredSecondsSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndSectionPMDDataPackage);

unsigned long farEndSeverelyErroredSecondsSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndSectionPMDDataPackage);

unsigned long farEndlossOfSignalSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndSectionPMDDataPackage);

unsigned long farEndFailureCounterSectionGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOfarEndSectionPMDDataPackage);

```

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

4.6 Vinculación de nombre

El siguiente módulo contiene información sobre la vinculación de nombre.

```
*/
    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 DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación