INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

V.59

(11/2000)

SERIES V: DATA COMMUNICATION OVER THE TELEPHONE NETWORK

Transmission quality and maintenance

Managed objects for diagnostic information of public switched telephone network connected V-series modem DCEs

ITU-T Recommendation V.59

(Formerly CCITT Recommendation)

ITU-T V-SERIES RECOMMENDATIONS

DATA COMMUNICATION OVER THE TELEPHONE NETWORK

General	V.1–V.9
Interfaces and voiceband modems	V.10–V.34
Wideband modems	V.35–V.39
Error control	V.40–V.49
Transmission quality and maintenance	**V.50–V.59**
Simultaneous transmission of data and other signals	V.60–V.99
Interworking with other networks	V.100–V.199
Interface layer specifications for data communication	V.200–V.249
Control procedures	V.250–V.299
Modems on digital circuits	V.300–V.399

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

ITU-T Recommendation V.59

Managed objects for diagnostic information of public switched telephone network connected V-series modem DCEs

Summary

This Recommendation specifies a series of Modem Managed Objects (MMO) that can be used for modem diagnostics across standardized interfaces.

Source

ITU-T Recommendation V.59 was prepared by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 17 November 2000.

For the purposes of forward and backward compatibility, this Recommendation is assigned a version number 1.

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 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 V.59

Managed objects for diagnostic information of public switched telephone network connected V-series modem DCEs

1 Scope

This Recommendation specifies a series of Modem Managed Objects (MMO), which can be used to diagnose modem connections on the Public Switched Telephone Network for performance and statistics. This information is only valid from the beginning of the last connection until the start of the next connection. These MMOs are relevant to information exchanged across standardized interfaces.

This Recommendation does not define the DTE to DCE management process.

The set of managed objects includes:

a) Modem Identification;

b) Modem Capabilities;

c) Modem Selected Mode;

d) Modem Diagnostics;

e) Line Conditions;

f) Call Progress;

g) Protocol Performance;

h) DTE/DCE interface status.

NOTE – There is some similarity between the contents of this Recommendation and that of ITU-T V.58. ITU-T V.58 defines a management model for V-series DCEs and was designed within the scope of the Telecommunication Management (TMN) activity defining objects required for the management of Network Elements (NE). This Recommendation is intended to provide a standard way for untrained modem users to extract diagnostic information from modems which could be passed on to a third party e.g. Internet Service Provider (ISP). Where ITU-T V.58 is used as a reference and if the same attributes are used in this Recommendation, this is indicated.

Recommendation version

For the purposes of forward and backward compatibility, this Recommendation is assigned a version number that may be included as one of the diagnostic items.

NOTE – The reader is encouraged to check on the ITU-T website for any normative or informative amendments to this Recommendation.

Version: 1

Status: Approved November 2000

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; all 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.

– ITU-T H.324 (1998), *Terminal for low bit-rate multimedia communication*.

– ITU-T T.30 (1999), *Procedures for document facsimile transmission in the general switched telephone network.*

– ITU-T V.8 (2000), *Procedures for starting sessions of data transmission over the public switched telephone network.*

– ITU-T V.8 *bis (*2000), *Procedures for the identification and selection of common modes of operation between data circuit-terminating equipments (DCEs) and between data terminal equipments (DTEs) over the public switched telephone network and on leased point-to-point telephone-type circuits.*

– ITU-T V.14 (1993), *Transmission of start-stop characters over synchronous bearer channels.*

– ITU-T V.17 (1991), *A 2-wire modem for facsimile applications with rates up to 14 400 bit/s*.

– ITU-T V.18 (2000), *Operational and interworking requirements for DCEs operating in the text telephone mode*.

– ITU-T V.21 (1988), *300 bits per second duplex modem standardized for use in the general switched telephone network.*

– ITU-T V.22 *bis* (1988), *2400 bits per second duplex modem using the frequency division technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.*

– ITU-T V.23 (1988), *600/1200-baud modem standardized for use in the general switched telephone network.*

– ITU-T V.24 (2000), *List of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE).*

– ITU-T V.25 (1996), *Automatic answering equipment and general procedures for automatic calling equipment on the general switched telephone network including procedures for disabling of echo control devices for both manually and automatically established calls.*

– ITU-T V.27 *ter* (1988), *4800/2400 bits per second modem standardized for use in the general switched telephone network.*

– ITU-T V.29 (1988), *9600 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits*.

– ITU-T V.32 *bis* (1991), *A duplex modem operating at data signalling rates of up to 14 400 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits*.

– ITU-T V.34 (1998), *A modem operating at data signalling rates of up to 33 600 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits.*

– ITU-T V.42 (1996), *Error correcting procedures for DCEs using asynchronous-to-synchronous conversion.*

– ITU-T V.42 *bis* (1990), *Data compression procedures for data circuit-terminating equipment (DCE) using error correction procedures.*

– ITU-T V.43 (1998), *Data flow control.*

– ITU-T V.44 (2000), *Data compression procedures*.

– ITU-T V.58 (1994), *Management information model for V-series DCEs.*

– ITU-T V.61 (1996), *A simultaneous voice plus data modem, operating at a voice plus data signalling rate of 4800 bit/s, with optional automatic switching to data-only signalling rates of up to 14 400 bit/s, for use on the General Switched Telephone Network and on leased point-to-point 2-wire telephone type circuits*.

– ITU-T V.70 (1996), *Procedures for the simultaneous transmission of data and digitally encoded voice signals over the GSTN, or over 2-wire leased point-to-point telephone type circuits*.

– ITU-T V.80 (1996), *In-band DCE control and synchronous data modes for asynchronous DTE*.

– ITU-T V.90 (1998), *A digital modem and analogue modem pair for use on the Public Switched Telephone Network (PSTN) at data signalling rates of up to 56 000 bit/s downstream and up to 33 600 bit/s upstream*.

– ITU-T V.91 (1999), *A digital modem operating at data signalling rates of up to 64 000 bit/s for use on a 4-wire circuit switched connection and on leased point-to-point 4-wire digital circuits*.

– ITU-T V.92 (2000), *Enhancements to Recommendation V.90*.

– ITU-T V.250 (1999), *Serial asynchronous automatic dialling and control*.

– ITU-T V.253 (1998), *Control of voice-related functions in a DCE by an asynchronous DTE*.

– ITU-T X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

– ITU-T X.691 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

3 Definitions

This Recommendation defines the following terms.

3.1 exchanged sequence: Coded binary sequences exchanged between DCEs during their start up phases.

3.2 analogue PCM modem: The analogue PCM modem is that as defined in ITU-T V.90.

3.3 digital PCM modem: The digital PCM modem is that as defined in ITU-T V.90.

3.4 *IA5 STRING*: A simple type whose distinguished values are an ordered sequence of zero, one or more octets, each octet being an ordered sequence of eight bits. The valid contents of this type are the character set of International Alphabet Number 5 (IA5).

3.5 modem-on-hold: This term describes the procedure in ITU-T V.92 that allows for the temporary suspension of a modem connection in favour for a second incoming call without termination of the initial call.

3.6 robbed bit signalling: This term describes the occasional use of bits from PCM code words to convey signalling information for call control, using the least significant bit from every 6th codeword.

4 Abbreviations

This Recommendation uses the following abbreviations.

APCM	Analogue PCM modem
ASN.1	Abstract Syntax Notation One
ASVD	Analogue Simultaneous Voice and Data
DCE	Data Circuit-terminating Equipment (Modem)
DPCM	Digital PCM modem
DSVD	Digital Simultaneous Voice and Data
DTE	Data Terminal Equipment
ISP	Internet Service Provider
MMO	Modem Managed Object
NE	Network Element
PSTN	Public Switched Telephone Network
RBS	Robbed Bit Signalling
TMN	Telecommunications Management Network

5 Modem Managed Objects Structure and Definitions

Information items that are used for diagnosing PSTN modem connections are modelled as abstract data objects, and the view of these items is a managed object. Objects with similar attributes may be grouped into Object classes. An object is characterized by its object class and object instance, and may posses multiple attribute types and associated values. The terms "managed object class" and "managed object instance" apply specifically to objects that are being managed. An object class may be a subclass of another Class. A subclass inherits attribute types, packages and behaviour of the super class, in addition to possessing its own specific attributes and properties.

For the scope of this Recommendation the following classes are used.

High-Level Object	Highest-level object in data structure hierarchy.
Mid-Level Object	A subclass of objects, which make up a High-Level Object. These objects inherit all the attributes of the High-Level Object.
Low-Level Object	A subclass of objects, which make up a Mid-Level Object. These objects inherit all the attributes of the Mid-Level objects.

Object classes and attribute types are defined only for the purpose of communicating diagnostic messages between systems, and need not be related to the structure of data within those systems.

The presentation of the managed objects within this Recommendation complies with the specification of ASN.1 (see ITU-T X.680). The ASN.1 as defined in Annex A should be used in conjunction with the object definitions to provide a complete overview of the diagnostics.

5.1 Compliance Requirements

The Recommendation does not require DCE behaviour that is inconsistent with other V-series PSTN modem Recommendations, or with national regulatory requirements, and shall be interpreted accordingly.

In order to be compliant with this Recommendation an implementation must provide the set of objects defined as being mandatory. Note as a general rule objects that cannot be derived directly from exchanged sequences, but need extra processing from the DCE, are optional.

5.2 Model Overview

The intent of ITU-T V.59 is to diagnose the modem off-line. Such that the DTE will not take long to determine what is wrong, so that the connection will work the next time. Figure 1 illustrates the typical PSTN connection model that is considered by this Recommendation. DTE/DCE pairs will be present at both sides of the PSTN. The inter-working characteristics of the various terminals are beyond the scope of this Recommendation. The connection types may be modem peer to peer (or client to client) and stand alone modem to Central Site modem (client to Server). Due to the various interface requirements for these types of modem, the method and procedures for exchanging the diagnostic information is outside the scope of this Recommendation.

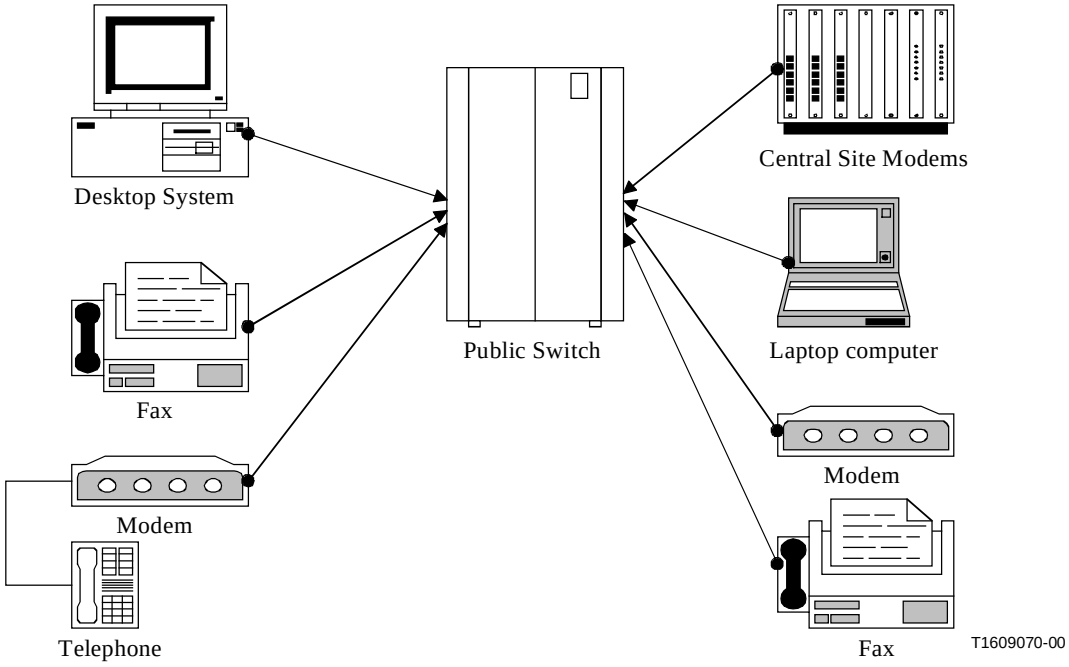

Figure 1/V.59 – Typical Network Model applicable to ITU-T V.59

5.2.1 Data Call States

Within a connection cycle, there are six Data Call states to consider. These states are shown in Figure 2 and described below.

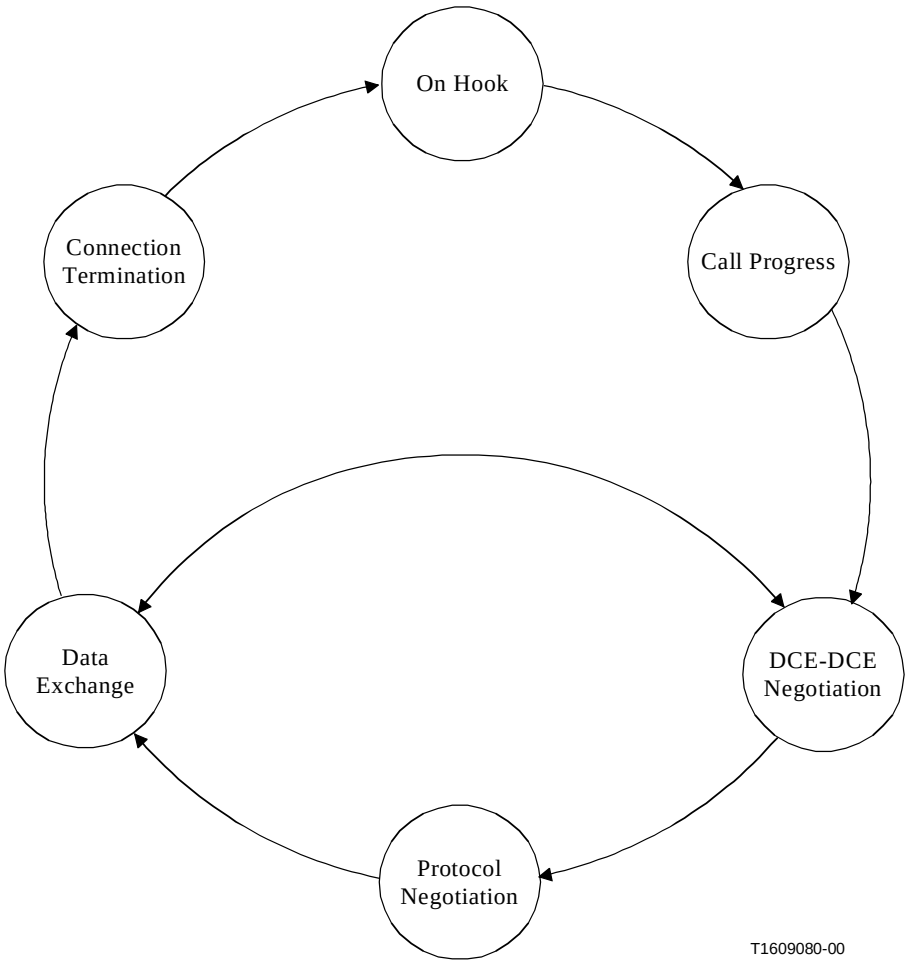

Figure 2/V.59 – Connection States diagram

On Hook

This is the idle state of the modem and is the point at which the diagnostic information can be extracted. The modems are waiting for either a request to initiate a connection, or are monitoring the network for a ringing signal to respond to a connection request.

Call Progress

In this state the progress of the PSTN set up is monitored. The call DCE may expect to see Dial tone or dial the desired number and proceed appropriately in accordance with network signalling tones. The answering DCE may detect the incoming call and initiate the transmission of Answer Tone.

DCE – DCE Negotiation

In this state the DCE will exchange capabilities, perform modulation selection, train-up the receivers and exchange modulation parameters.

Protocol Negotiation

Once a data channel is established, the DCE may optionally negotiate and exchange the various Protocol parameters. Examples of this are Error Control and Data Compression. Note that higher layer protocols (e.g. PPP, TCP/IP, etc.) are beyond the scope of this Recommendation.

Data Exchange

In this state the modems are exchanging user data. The DCE is still responsible for the integrity of the channel. This includes the possibility of transitioning to the DCE-DCE Negotiation State if retraining the modem or changing the data-signalling rate is necessary and is shown as a transition to the DCE-DCE Negotiation state in the figure above.

Call Termination

A call can be terminated either as a user request or erroneously. Error cases include problems with the data transport mechanism, the failure of the DCE to maintain signal integrity of the connection or due to a network interruption.

The model described above considers only the case where the local DTE requests and obtains the diagnostic information from its local DCE. Extracting diagnostic information from a remote DCE is for further study.

5.2.2 Scope and Volatility of Objects

The managed objects as defined and described in this Recommendation shall only be initialized or cleared by one of the following events:

– Power off

– Forced Reset

When the DCE goes off-hook in response to a request to either initiate an outgoing call or to answer an incoming call (automatic or manual).

NOTE – In the case that a DTE initiates a soft reset during a call (for example via the V.250 ATZ command) the modem shall record this event as being the cause for the call termination and shall not clear the objects.

6 Object Classes

The following clauses describe and define the various High-Level Objects and their sub-objects.

6.1 Object Identification

As described in clause 5 the objects have a distinct hierarchy. Each object is assigned an object name and a unique Tag-ID to aid in the identification of the object. Since storage efficiency is an important factor, the use of the optional Tag-ID may in certain circumstances aid this requirement. Tag-IDs are formatted as a four digit hexadecimal number. The two most significant digits represent the high-level object class and the least two significant digits indicate the mid and low-level object IDs. A summary of the Tag-Ids associated with each high level set of objects is provided above each class description.

The following is an example of equivalent V.90 diagnostics including both the object name and Tag-ID options:

iNFO1a{FE45;1130;ABCD;00FF;1E2C} or *0902{FE45;1130;ABCD;00FF;1E2C}*

rxSignalQuality{0.445;0.312} or *0950{0.445;0.312}*

Illustrates two V.90 diagnostics.

NOTE – Object names are used as the primary means of object identification within this Recommendation.

6.2 Data Type definitions

The data types used in this Recommendation are based upon those defined in ASN.1. For types defined as an IA5 STRING (International Alphabet No. 5), the following string types are used.

A text string consists of an object label followed by a set of delimited parameters. The delimiters are a pair of {} parentheses for the object and semi-colons are used to delimit the parameters within an object. In the definition of the objects contained in this Recommendation the String Formats are declared as IA5String: simpleText or IA5String: dataSequence. The following example illustrates a simpleText type and a dataSequence type respectively.

Example:

INFO1a{ FE45;1130;ABCD;00FF;1E2C}

rxSignalQuality{0.445;0.312}

The default maximum size for the IA5String data type is 40 octets, unless otherwise specified in the object description below.

6.2.1 String Type simpleText

The String type simpleText consists of the character set as defined in clause 10/X.680 and Table 2/X.680.

Example:

noiseEstimate{-55dBm}

txDataHistory{33600;31200;33600;28800}

6.2.2 String Type dataSequence

All bit stream sequence strings are represented as hexadecimal numbers. Start, Stop and fill bits are not included, nor are the CRC's if applicable. Sequences that have a natural parsing into N-bit words shall utilize semi-colons as a delimiter. For each parsed word the first bit in time of the sequence shall be the least significant bit of the hexadecimal representation.

Hexadecimal numbers shall be case insensitive.

For example: iNFO1a{FE45;1130;ABCD;00FF;1E2C}

6.2.3 BIT STRING

Where BIT STRING is used to identify an object a 0 indicates non-availability or disabled, a 1 indicates availability or enabled.

6.2.4 NULL Type

The Null type has a simple type value of NULL (refer to ITU-T X.680) and is used to indicate the absence of a value or sequence. If an object is requested but is not applicable to the DCE state or configuration, then a NULL shall be the returned.

Example:

farEchoEstimate{NULL}, could be a response to a query to a V.90 APCM.

6.2.5 Comment on the format of numbers

Integers are numbers consisting of one or more digits, which can be positive or negative. The first digit shall not be zero unless the number is a single digit.

NOTE – The "number" item is always mapped to an integer value by interpreting it in decimal notation.

For objects that use counters or numbers that exceed their specified limits, the returned value shall be set at the specified maximum value.

6.2.6 Object Lists

Object lists that represent a history of that object are formatted such that the oldest event is first and the most recent is the last of the lists.

6.3 High-Level Objects

The following is the set of managed objects that constitute the High-Level objects for this Recommendation.

Object Identifier	**Tag Id**
ModemIdentity	00xx
ModemCapability	01xx
ModeSelected	02xx
Reserved	03xx – 06xx
V8Diag	07xx
V8bisDiag	08xx
V90Diag	09xx
V91Diag	0Axx
V92ModDiag	0Bxx
V34Diag	0Cxx
V32Diag	0Dxx
V22Diag	0Exx
V23Diag	0Fxx
V21Diag	10xx
Reserved	11xx – 14xx
V34HdxDiag	15xx
V17Diag	16xx
V29faxDiag	17xx
V27tfaxDiag	18xx
Reserved	19xx – 22xx
V18Diag	23xx
V70Diag	24xx
V61Diag	25xx
NSMDiag	26xx
Reserved	27xx – 2Bxx
CnxDiag	2Cxx
LineConDiag	2Dxx
CallProgress:	2Exx
ErrorControl	2Fxx
Compression	30xx
DTEDCE	31xx
Reserved	32xx – FFxx

An object called V59Objects is defined exclusively for the completeness of the ASN.1 contained in Annex A and is the equivalent of a Union in C of all the Mid-Level objects.

6.4 Common Mid-Level Objects

The following clause defines a common set of mid-level objects. These objects are used throughout this Recommendation and have been placed in this clause for easy maintenance and reference. The Tag-ID's for the common set of objects have been allocated numbers in the range 40 to 5F hexadecimal. Unused numbers in this range are reserved for future use.

Common Object Attributes

Object: iNFO0Tx

Tag-ID: 41

Description: This object displays the INFO0 sequence transmitted by the modem.

Data type: IA5STRING*:*dataSequence.

Mandatory: Yes

Object: iNFO0Rx

Tag-ID: 42

Description: This object displays the INFO0 sequence received by the modem.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: precodeCoeffTx

Tag-ID: 43

Description: This object returns the Precoder coefficients as used in the transmitter. Note that the Precoder coefficient values have been separated from the their appropriate Exchanged Sequence. This is to facilitate simpler presentation for these objects. The coefficients are expressed as a delimited set of real and imaginary pairs in hexadecimal form as if extracted from a type 1 MP sequence. The pairs are separated by commas and the sets by a semicolon. The order of the sets is h(1), h(2) and h(3). The hexadecimal sequences represent the 16-bit coefficients.

Data type: IA5STRING*:*dataSequence

Example*:*

> PrecodeCoeffTx{real h(1), imaginary h(1); real h(2), imaginary h(2); real h(3), imaginary h(3)}

Mandatory: No

Object: precodeCoeffRx

Tag-ID: 44

Description: This object returns the receiver Precoder coefficients. See precodeCoeffTx above for the definition and format of the output for this object type.

Data type: IA5STRING*:*dataSequence

Mandatory: No

Object: initialTxPower

Tag-ID: 45

Description: This object provides the initial transmit power level as configured by the modem and contains the unit of power used. For example initialTxPower{-9dBm} or initialTxPower{-80dBrn}. This object is part of the TxPowerLevel Mid-Level object.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: txPowerDrop

Tag-ID: 46

Description: This object provides the amount of transmit power reduction if appropriate to the modulation mode as configured by the modem. If not used then the result is a NULL string. This object is part of the TxPowerLevel Mid-Level object.

Data type: IA5STRING:simpleText

Mandatory: No

Object: txPower

Tag-ID: 47

Description: This object provides an estimation of the transmit power level. This object is part of the TxPowerLevel Mid-Level object.

Data type: IA5STRING:simpleText

Mandatory: No

Object: txSymbolRate

Tag-ID: 48

Description: Delimited list of transmitter Symbol Rates used during the connection. The possible values for this object are defined by the codes shown in Table 1.

Data type: ENUMERATED.

Table 1/V.59 – Symbol Rate Codes

Symbol Rate (symbols/second)	Code
8000	0
3429	1
3000	2
3200	3
2800	4
2743	5
2400	6
1600	7
1200	8
600	9
NOTE – A rate of 8000 means that PCM encoding is being used in the transmitter.	

Mandatory: No

Object: rxSymbolRate

Tag-ID: 49

Description: Delimited list of receiver Symbol Rates used during the connection (same as transmitter). The possible values for this object are defined in Table 1.

NOTE – A rate of 8000 indicates that PCM encoding is being used in the receiver.

Data type: ENUMERATED.

Mandatory: No

Object: txCarrierFreq

Tag-ID: 4A

Description: Delimited list of Transmitter Carrier Frequencies used in the transmitter that correspond to the transmitter symbol rates used. The Codes are defined in Table 2.

Data type: ENUMERATED.

Table 2/V.59 – Carrier Frequency Codes

Carrier Frequency (Hertz)	Code
1200	0
1600	1
1646	2
1680	3
1700	4
1800	5
1829	6
1867	7
1920	8
1959	9
2000	10
2400	11

Mandatory: No

Object: rxCarrierFreq

Tag-ID: 4B

Description: Delimited list of Receiver Carrier Frequencies used in the transmitter that correspond to the receiver symbol rates used. The Codes used in this object is defined in Table 2.

Data type: ENUMERATED.

Mandatory: No

Object: txDataHistory

Tag-ID: 4C

Description: Delimited list of transmitter Primary Data rates used during connection.

Data type: IA5STRING*:*simpleText

Example*:*

TxDataHistory{45333;44000;33333;33600}

Mandatory: Yes

Object: rxDataHistory

Tag-ID: 4D

Description: Delimited list of receiver Primary Data rates used during connection (same as txDataHistory).

Data type: IA5STRING*:*simpleText

Mandatory: Yes

Object: rxLevelEstimate

Tag-ID: 4E

Description: Estimation of the Receive power level as measured by the modems input in specified units (e.g. rxLevelEstimate{-25dBm0}). (Note this may not be the same as that if measured at the line interface.)

Data type: IA5STRING:simpleText

Mandatory: No

Object: noiseEstimate

Tag-ID: 4F

Description: Estimation of the noise level measured in specified units. (e.g. noiseEstimate{-55dBm} or noiseEstimate{35dBrn}.)

Data type: IA5STRING:simpleText

Mandatory: No

Object: rxSignalQuality

Tag-ID: 50

Description: This is an indication of receiver signal quality measured at the start of the connection and if available at a point near to the termination of the connection. There are two options possible. The first is a text string representing the signal quality in fractional form, where 1.0 is the highest quality and 0.0 is the worst. (e.g. "0.312") Alternatively if defined as an Integer the number has proprietary meaning.

Example:

> The string "0.445;0.312" implies that the initial signal quality was normalized 0.445 but deteriorated at the end of the connection to be 0.312)

Data type: IA5STRING:simpleText

Mandatory: No

Object: nearEchoEstimate

Tag-ID: 51

Description: This is an estimate of the near end echo measured at the beginning of the connection and if available at the end of the connection. The units should be specified in the string. No indicated unit implies a custom numbering is being used.

Data type: IA5STRING:simpleText

Mandatory: No

Object: farEchoEstimate

Tag-ID: 52

Description: This is an estimate of the far end echo measured at the beginning of the connection and if available at the end of the connection. The units should be specified in the string. No indicated unit implies a custom numbering is being used.

Data type: IA5STRING:simpleText

Mandatory: No

Object: roundTripDelay

Tag-ID: 53

Description: This is an estimate of the round trip delay of the connection. It is displayed in a string with indicated units. For example "25ms", "0.025s" or "166T" are allowable. (T means symbol periods.)

Data type: IA5STRING:simpleText

Mandatory: No

Object: cP

Tag-ID: 54

Description: displays the last CP sequence.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: rBSPattern

Tag-ID: 55

Description: This object displays the Robbed Bit Signalling pattern. The observed pattern is presented as a packed 6-bit object with a "1" indicating the presence of a robbed bit and a "0" indicates the absence of a robbed bit. This object represents a 6-bit cyclic presentation and no synchronization with the observed robbed bit pattern and network frame alignment is assumed.

Data type: BIT STRING

Table 3/V.59 – Bit Position for RBS Pattern

Mnemonic	Bit Position	Description
robBit0	0	Bit 0 Robbed
robBit1	1	Bit 1 Robbed
robBit2	2	Bit 2 Robbed
robBit3	3	Bit 3 Robbed
robBit4	4	Bit 4 Robbed
robBit5	5	Bit 5 Robbed

Mandatory: No

Object: digitalPadLoss

Tag-ID: 56

Description: This object indicates the estimate of digital attenuation as measured in the downstream direction. The string indicates the amount of loss and the units used. The units can be in dB or in decimal fractional format.

Data type: Octal String*:* simpleText

Example*:*

digitalPadLoss{3dB}, or digitalPadLoss{0.5011}

Mandatory: No

6.5 Modem Identification Objects

This managed object contains all the generic information pertaining to its identification.

Modem Identification Objects

Object Identifier	Tag-ID
manufacturer	0001
model	0002
firmwareVersion	0003
diagnosticVersion	0004
multimediaMode	0005

Modem Identification Object Attributes

Object: manufacturer

Tag-ID: 01

Description: This object is a string providing information on the manufacturer of the modem.

Data type: IA5STRING*:*simpleText

Mandatory: Yes

Object: model

Tag-ID: 02

Description: This object provides information on the model or product, as an IA5 string.

Data type: IA5STRING*:*simpleText

Mandatory: Yes

Object: firmwareVersion

Tag-ID: 03

Description: This object further identifies the modem's firmware revisions as an IA5 string.

Data type: IA5STRING*:*simpleText

Mandatory: Yes

Object: diagnosticVersion

Tag-ID: 04

Description: This object defines the version of managed objects of diagnostics supported. The format is a single Integer. See 3.1.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: multimediaMode

Tag-ID: 05

Description: This object indicates the multimedia capability of the modem. The object is defined as a sequence of binary values indicating as being capable (binary 1) or incapable (binary 0). The definition of this binary sequence is shown in Table 4.

Data type: BIT STRING

Table 4/V.59 – Multimedia Capability indication

Mnemonic	Bit Position	Description
dataMode	0	DCE provides data mode
faxT30	1	DCE provides Facsimile transmission and reception per ITU-T T.30
faxT30C	2	DCE provides Facsimile transmission and reception per Annex C/T.30
fAXT30F	3	DCE provides Facsimile transmission and reception per Annex F/T.30
voiceV253	4	DCE provides voice function and control per ITU-T V.253
sVDV70	5	The DCE provides Simultaneous voice and Data functionality per ITU-T V.70
sVDV61	6	The DCE provides Simultaneous voice and Data functionality per ITU-T V.61
vidTelH.324	7	The DCE provides video telephony functionality per ITU-T H.324
v80other	8	The DCE provides other forms of V.80 type functionality
V18Text	9	The DCE provides V.18 text telephone type functionality

Example:

ModemIdentity{ModemMaker Inc; PCI Card;2.66;1.0;83}

Mandatory: No

6.6 Modem Capability

This object defines the modems capabilities. Each of the objects contained in this high-level object are defined under their own appropriate object class but all share the same Enumerated return code as defined in Table 5.

Table 5/V.59 – Mode Capability Codes

Mnemonic	Code	Description
notSupported	0	This mode of operation is not supported
default	1	This mode is supported and is the preferred/default mode of operation
enabled	2	This mode of operation is Supported and Enabled (may not be default)
disabled	3	This mode of operation is Disabled, although it is supported

The following is the list of modem capability objects.

Modem Capability Objects

Object Identifier	Tag-ID
modeV90	0900
modeV91	0A00
modeV92	0B00
modeV34	0C00
modeV32B	0D00
modeV22B	0E00
modeV23	0F00
modeV21	1000
modeV34H	1500
modeV17	1600
modeV29	1700
modeV27T	1800
modeV18	2300
otherModMode	0101
modeV42	2F00
modeV42B	3000
modeV44	300A
otherProtMode	0102

Modem Capability Object Attributes

Object: otherModMode

Tag-ID: 01

Description: This object allows the ability to determine if proprietary or other standard type modulation modes are supported in the modem.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: otherProtMode

Tag-ID: 02

Description: This object allows the ability to determine if proprietary or other protocol modes are supported in the modem.

Data type: IA5STRING:simpleText

Mandatory: No

6.7 Modem Selected Mode Objects

This object consists of the following mid-level definition.

Modem Selected Mode Objects

Object Identifier	Tag-ID
modulationHistory	0201

Modem Selected Mode Object Attributes

Object: modulationHistory

Tag-ID: 01

Description: This object indicates as a string the sequence of modulations the modem has operated for the duration of the past call. The format of this object is to use the ITU-T V-series Recommendation number in a delimited list. For the situation where a proprietary mode is part of the modulation history, the object may contain a short string as a descriptor.

Data type: IA5STRING:simpleText

Example:

ModulationHistory{V.90;V.34}

Mandatory: Yes

6.8 Modem Diagnostic Objects

The response to this diagnostic object depends upon the modulation being used in the previous connection. If however a modem initially connects in one modulation but during that connection falls back or changes to another modulation then both sets of diagnostics may be used.

NOTE – Objects that have been previously defined are marked by the symbol †.

6.8.1 V.8 Diagnostics

This set of objects are applicable for V.8.

V.8 Objects

Object Identifier	Tag-ID
Ci	0701
Cm	0702
Jm	0703
v8Result	0704

V.8 Object Attributes

Object: ci

Tag-ID: 01

Description: This object contains the CI sequence transmitted if calling or received if answering.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: cm

Tag-ID: 02

Description: This object contains the CM sequence transmitted if calling or received if answering.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: jm

Tag-ID: 03

Description: This object contains JM sequence transmitted if answering or received if calling.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: v8ResultCode

Tag-ID: 04

Description: This object provides a code that indicates the result of the V.8 negotiation.

Data type: ENUMERATED

Table 6/V.59 – V.8 Negotiation Result Codes

Mnemonic	Code	Description
v8ComNeg	0	Common mode negotiated
v8NoComNeg	1	No common mode negotiated
v8NoNegAuto	2	No V.8 negotiation, modem auto-moded

Mandatory: Yes

6.8.2 V.8 *bis* Diagnostics

The following clause defines the objects used for the diagnosis of V.8 *bis*.

V.8 *bis* Objects

Object Identifier	Tag-ID
v8bisTransaction	0801
mRtype	0802
mRSequence	0803
cRtype	0804
cRSequence	0805
cLSequence	0806
cLRSequence	0807
eSType	0808
eSSequence	0809
aCK	080A
nAK	080B

V.8 *bis* Object Attributes

Object: v8bisTransaction

Tag-ID: 01

Description: This object identifies the type of V.8 *bis* transaction used in the previous connection.

Data type: INTEGER value 1 to 13

Mandatory: Yes

Object: mRtype

Tag-ID: 02

Description: Indicates the type of MR sequence either transmitted or received.

Data type: ENUMERATED.

Table 7/V.59 – Definition for MR signal types

Mnemonic	Code	Description
none	0	No MR sequence transmitted or received
e	1	Sequence type is MRe
d	2	Sequence type is MRd

Mandatory: Yes

Object: mRSequence

Tag-ID: 03

Description: This object contains the MR sequences either transmitted or received.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: cRtype

Tag-ID: 04

Description: Indicates the type of CR sequence either transmitted or received.

Data type: ENUMERATED

Table 8/V.59 – Definition for CR signal types

Mnemonic	Code	Description
none	0	No CR sequence transmitted or received
e	1	Sequence type is Cre
d	2	Sequence type is CRd

Mandatory: Yes

Object: cRSequence

Tag-ID: 05

Description: This object contains the CR sequences either transmitted or received.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: cLSequence

Tag-ID: 06

Description: This object contains the CL sequences either transmitted or received.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: cLRSequence

Tag-ID: 07

Description: This object contains the CLR sequences either transmitted or received.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: eSType

Tag-ID: 08

Description: Indicates the type of ES sequence either transmitted or received.

Data type: ENUMERATED

Table 9/V.59 – Definition for ES signal types

Mnemonic	Code	Description
none	0	No ES sequence transmitted or received
i	1	Sequence type is ESi
r	2	Sequence type is ESr

Mandatory: Yes

Object: eSSequence

Tag-ID: 09

Description: This object contains the ES sequences either transmitted or received.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: aCK

Tag-ID: 0A

Description: Indicates the type of ACK message either transmitted or received.

Data type: ENUMERATED

Table 10/V.59 – Definition for ACK types

Mnemonic	Code	Description
none	0	No ACK sequence transmitted or received
aCK1	1	ACK1 message transmitted or received
aCK2	2	ACK2 message transmitted or received

Mandatory: Yes

Object: nAK

Tag-ID: 0B

Description: Indicates the type of NAK message either transmitted or received.

Data type: ENUMERATED

Table 11/V.59 – Definition for NAK types

Mnemonic	Code	Description
none	0	No NAK sequence transmitted or received
nAK1	1	NAK1 message transmitted or received
nAK2	2	NAK2 message transmitted or received
nAK2	3	NAK3 message transmitted or received
nAK2	4	NAK4 message transmitted or received

Mandatory: Yes

6.8.3 V.90 Diagnostics

The content of the V90Diag high-level object is described in this clause. Note that some of the objects have previously been defined in the set of common objects (6.4).

V.90 Modulation Objects

Object Identifier	Tag-ID
modeV90	0900
iNFO0_Tx†	0941
iNFO0_Rx†	0942
iNFO1d	0901
iNFO1A	0902
mP	0903
cP	0954
cPt	0904

jAsequence	0905
jDsequence	0906
v90TxLevel†	0945, 0946, 0947: This object uses the TxPowerLevel definition
txSymbolRate†	0948
rxSymbolRate†	0949
txCarrierFreq†	094A
rxCarrierFreq†	094B
txDataHistory†	094C
rxDataHistory†	094D
rxLevelEstimate†	094E
noiseEstimate†	094F
rxSignalQuality†	0950
nearEchoEstimate†	0951
farEchoEstimate†	0952
roundTripDelay†	0953
rBSpattern	0955
digitalPadLoss	0956
codecLaw	0907

V.90 Modulation Object Attributes

Object: modeV90

Tag-ID: 00

Description: This object displays the modems V.90 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: iNFO1d

Tag-ID: 01

Description: This object displays the INFO1d sequence transmitted or received by the modem if in V.90 mode and depending if calling or answering.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: iNFO1a

Tag-ID: 02

Description: This object displays the INFO1a sequence transmitted or received by the modem if in V.90 mode.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: mP

Tag-ID: 03

Description: This object displays the MP sequence.

NOTE – Precoder coefficients for type 1 MP sequences are omitted.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: cPt

Tag-ID: 04

Description: Displays the last CPt sequence.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: jAsequence

Tag-ID: 05

Description: Displays the JA sequence.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: jDsequence

Tag-ID: 06

Description: Displays the JD sequence.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: CodecLaw

Tag-ID: 07

Description: This object is extracted from the CP sequence and describes whether the codec is A-law or μ-law. For the DPCM this object indicates the local (downstream) codec. For the APCM this object indicates the remote codec. For A-Law the object value is TRUE and for μ-law the value is FALSE.

Data type: BOOLEAN

Mandatory: No

6.8.4 V.91 Diagnostic Objects

The definitions for the V91Diag high-level object are described in this clause.

V.91 Modulation Objects

Object Identifier	Tag-ID
modeV91	0A00
iNFO0_TX†	0A41
iNFO0_RX†	0A42
cP†	0A54
v91TxPowerLevel†	0A45, 0A46, 0A47: This object uses the TxPowerLevel definition
controlChannel	0A01
transparentMode	0A02
txDataHistory†	0A4D
rxDataHistory†	0A4E
noiseEstimate†	0A50
rxSignalQuality†	0A51
rBSpattern†	0A55
digitalPadLoss†	0A56
localCodecLaw	0A04
remoteCodecLaw	0A05
frameSlipsDetected	0A03

V.91 Modulation Object Attributes

Object: modeV91

Tag-ID: 00

Description: This object displays the modems V.91 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: controlChannel

Tag-ID: 01

Description: This object indicates if the optional control channel is enabled (TRUE) or disabled (FALSE).

Data type: BOOLEAN

Mandatory: Yes

Object: transparentMode

Tag-ID: 02

Description: This object indicates the selected mode of operation was transparent mode (TRUE) or Encoded mode (FALSE).

Data type: BOOLEAN

Mandatory: Yes

Object: frameSlipsDetected

Tag-ID: 03

Description: This object counts the number of Frame Slips detected during the connection.

Data type: Integer (0 to 256)

Mandatory: No

Object: localCodecLaw

Tag-ID: 04

Description: This object indicates the codec companding law used by the local transmitter. A-law is represented by a TRUE value and μ-law by a FALSE value.

Data type: BOOLEAN

Mandatory: No

Object: remoteCodecLaw

Tag-ID: 05

Description: This object indicates the codec companding law used by the remote transmitter. A-law is represented by a TRUE value and μ-law by a FALSE value.

Data type: BOOLEAN

Mandatory: No

6.8.5 V.92 Diagnostics

The V.92 diagnostics consist of a single High Level Object, that defines the mid-level objects for V.92 modulation, the short Phase 1 Procedures and Call Waiting and Modem-on-Hold.

V92 Modulation Objects

Identificateur d'objet	**Identificateur d'étiquette**
modeV92	0B00
iNFO0_TX†	0B41
iNFO0_RX†	0B42
iNFO1d	0B01
iNFO1a-US	0B02
iNFO1a-SP2	0B03
jA	0B04
jD	0B05
jDP	0B06
cP	0B5
v92TxLevel†	0B45, 0B46, 0B47: Cet objet utilise la définition de TxPowerLevel
v92SymbolRate†	0B48
txDataHistory†	0B4C

rxDataHistory†	0B4D
rxLevelEstimate†	0B4E
noiseEstimate†	0B4F
rxSignalQuality†	0B50
nearEchoEstimate†	0B51
farEchoEstimate†	0B52
roundTripDelay†	0B53
rBSpattern†	0B55
digitalPadLoss†	0B56
localCodecLaw†	0B07
remoteCodecLaw†	0B08
sP1Enable	0B10
lastCallSP1	0B11
qC1a	0B12
qC1d	0B13
qC2a	0B14
qC2d	0B15
qCA1a	0B16
qCA1d	0B17
qCA2a	0B18
qCA2d	0B19
cWDEnable	0B20
mOHEnable	0B21
lastMOHaction	0B22
mohTimeout	0B23
lastMOHduration	0B24
mHreq	0B25
mHack	0B26
mHnak	0B27
mHcld	0B28
mHcda	0B29
mHfrr	0B2A

V.92 Modulation Object Attributes

Object: modeV92

Tag-ID: 00

Description: This object displays the modems V.92 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: iNFO1d

Tag-ID: 01

Description: This object displays the INFO1d sequence transmitted or received by the modem when in V.92 mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: iNFO1a-US

Tag-ID: 02

Description: This object displays the INFO1a sequence transmitted or received by the modem when in V.92 mode if PCM encoding has been selected for upstream, depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: iNFO1a-SP2

Tag-ID: 03

Description: This object displays the INFO1a sequence transmitted or received by the modem when Short Phase 2 is used for V.92 mode, depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: jA

Tag-ID: 04

Description: This object displays the JA sequence transmitted or received by the modem when in V.92 mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: jD

Tag-ID: 05

Description: This object displays the JD sequence transmitted or received by the modem when in V.92 mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: jDP

Tag-ID: 06

Description: This object displays the JD_p sequence transmitted or received by the modem when in V.92 mode.

Data type*:* IA5 STRING*:*dataSequence

Mandatory: Yes

Object: localCodecLaw

Tag-ID: 07

Description: This object indicates the codec companding law used by the local transmitter. A-law is represented by a TRUE value and μ-law by a FALSE value.

Data type: BOOLEAN

Mandatory: No

Object: remoteCodecLaw

Tag-ID: 08

Description: This object indicates the codec companding law used by the remote transmitter. A-law is represented by a TRUE value and μ-law by a FALSE value.

Data type: BOOLEAN

Mandatory: No

V.92 Short Phase 1 Object Attributes

Object: sP1Enable

Tag-ID: 10

Description: This object indicates that when the modem is V.92 capable, whether the short phase 1 Procedures are enabled.

Data type: BOOLEAN (True is enabled, False is disabled)

Mandatory: Yes

Object: lastCallSP1

Tag-ID: 11

Description: This object indicates if the previous connection was made using V.92 short phase 1 Procedures.

Data type: BOOLEAN (True is enabled, False is disabled)

Mandatory: Yes

Object: qC1a

Tag-ID: 12

Description: This object displays the QC1A sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qC1d

Tag-ID: 13

Description: This object displays the JD sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qC2a

Tag-ID: 14

Description: This object displays the QC2A sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qC2d

Tag-ID: 15

Description: This object displays the QC2d sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qCA1a

Tag-ID: 16

Description: This object displays the QCA1a sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qCA1d

Tag-ID: 17

Description: This object displays the QCA1d sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qCA2a

Tag-ID: 18

Description: This object displays the QCA2a sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: qCA2d

Tag-ID: 19

Description: This object displays the JD sequence transmitted or received by the modem if V.92 Short phase 1 Mode is enabled mode and depending if APCM or DPCM.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

V.92 Call Waiting and Modem-on-Hold Object Attributes

Object: cWDEnable

Tag-ID: 20

Description: This object indicates whether the Call-Waiting Detect function is enabled.

Data type: BOOLEAN (True is enabled, False is disabled)

Mandatory: Yes

Object: mOHEnable

Tag-ID: 21

Description: This object indicates that if the modem is V.92 capable, whether the Modem-on-Hold Procedures are enabled.

Data type: BOOLEAN (True is enabled, False is disabled)

Mandatory: Yes

Object: lastMOHaction

Tag-ID: 22

Description: This object indicates by a code that if the modem is V.92 capable and Modem-on-Hold is enabled, what the result of the last Modem-on-Hold transaction was.

Data type: ENUMERATED

Table 12/V.59 – Previous Connection Result Codes for V.92 Modem-on-hold

Mnemonic	Code Value	Description
mohaccepted	0	The two V.92 capable modems have signalled and accepted the Modem-on-Hold request
mohdeclined	1	The Modem receiving the interruption event declines to go on-hold and tries to resume the data connection
mohdenied	2	The Modem requesting the on-hold state has been denied by remote modem
mohclrdn	3	The modems have negotiated a cleardown in response to the interrupt event and or on-hold request

Mandatory: Yes

Object: mOHTimeout

Tag-ID: 23

Description: This object indicates the amount of time allocated by the DPCM for the duration of the modem-on-hold.

Data type: IA5STRING*:*SimpleText

Mandatory: Yes

Object: lastMOHduration

Tag-ID: 24

Description: This object indicates the duration of the last on-hold occurrence.

Data type: IA5STRING*:*SimpleText

Mandatory: Yes

Object: mHreq

Tag-ID: 25

Description: This object displays the MHreq sequence transmitted or received by the modem if V.92 MOH Mode is enabled mode.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: MHack

Tag-ID: 26

Description: This object displays the MHack sequence transmitted or received by the modem if V.92 MOH Mode is enabled mode.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: mHnak

Tag-ID: 27

Description: This object displays the MHnak sequence transmitted or received by the modem if V.92 MOH Mode is enabled mode.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: mHcld

Tag-ID: 28

Description: This object displays the MHcld sequence transmitted or received by the modem if V.92 MOH Mode is enabled mode.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: mHcda

Tag-ID: 29

Description: This object displays the MHcda sequence transmitted or received by the modem if V.92 MOH Mode is enabled mode.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: mHfrr

Tag-ID: 2A

Description: This object displays the MHFrr sequence transmitted or received by the modem if V.92 MOH Mode is enabled mode.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

6.8.6 V.34 Duplex Diagnostics

The following are the definitions for the V.34 diagnostic objects.

V.34 Modulation Objects

Object Identifier	Tag-ID
ModeV34	0C00
iNFO0Tx†	0C41
iNFO0Rx†	0C42
iNFO1c	0C01
iNFO1a	0C02
mPTx	0C03
mPRx	0C04

precodeCoeff_Tx†	0C43
precodeCoeff_Rx†	0C44
v34TxLevel†	0C45,0C46,0C47: Object uses the TxPowerLevel mid-level object
txSymbolRate†	0C48
rxSymbolRate†	0C49
txCarrierFreq†	0C4A
rxCarrierFreq†	0C4B
txDataHistory†	0C4C
rxDataHistory†	0C4D
rxLevelEstimate†	0C4E
noiseEstimate†	0C4F
rxSignalQuality†	0C50
nearEchoEstimate†	0C51
farEchoEstimate†	0C52
roundTripDelay†	0C53

V.34 Modulation Object Attributes

Object: modeV34

Tag-ID: 00

Description: This object displays the modems V.34 duplex operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: iNFO1c

Tag-ID: 01

Description: This object displays the INFO1c sequence either transmitted or received by the modem.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: iNFO1a

Tag-ID: 02

Description: This object displays the INFO1a sequence either transmitted or received by the modem.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: mPTx

Tag-ID: 03

Description: This object displays the MP sequence transmitted.

NOTE 1 – Precoder coefficients for type 1 MP sequences are not included.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: mPRx

Tag-ID: 04

Description: This object displays the MP sequence received.

NOTE 2 – Precoder coefficients for type 1 MP sequences are not included.

Data type: IA5STRING:dataSequence

Mandatory: Yes

6.8.7 V.32 *bis* Diagnostics

The objects defined below are applicable to modulations V.32 and V.32 *bis*.

V.32 *bis* Modulation Objects

Object Identifier	Tag-ID
ModeV32B	0D00
v32TxLevel†	0D45,0D46,0D47: This object uses the TxPowerLevel mid-level object
v32txRateSeq	0D01
v32rxRateSeq	0D02
txDataHistory†	0D4C
rxDataHistory†	0D4D
rxLevelEstimate†	0D4E
noiseEstimate†	0D4F
rxSignalQuality†	0D50
nearEchoEstimate†	0D51
farEchoEstimate†	0D52
roundTripDelay†	0D53

V.32 *bis* Modulation Object Attributes

Object: modeV32B

Tag-ID: 00

Description: This object displays the modems V.32 or V.32 *bis* operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: v32TxRateSeq

Tag-ID: 01

Description: Rate sequence transmitted.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

Object: v32RxRateSeq

Tag-ID: 02

Description: Rate sequence received.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

6.8.8 V.22 *bis* Diagnostics

The objects defined below are applicable to modulations V.22 and V.22 *bis*.

V.22 *bis* Modulation Objects

Object Identifier	Tag-ID
modeV22B	0E00
dataRateHistory†	0E4C
v22TxLevel	0E45,0E46,0E47: This object uses the TxPowerLevel mid-level object
v22RateSeq	0E01
rxLevelEstimate†	0E4E
noiseEstimate†	0E4F
rxSignalQuality†	0E50

V.22 *bis* Modulation Object Attributes

Object: modeV22B

Tag-ID: 00

Description: This object displays the modems V.22 or V.22 *bis* operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: v22RateSeq

Tag-ID: 01

Description: This object describes the result of the two-bit rate sequence used in V.22 *bis* modulation.

Data type: IA5STRING*:*dataSequence

Mandatory: Yes

6.8.9 V.23 Diagnostics

The objects defined below are applicable to modulation type V.23.

V.23 Modulation Objects

Object Identifier	Tag-ID
ModeV23	0F00
v23TxLevel†	0F45,0F46,0F47: This object uses the TxPowerLevel mid-level object
duplex	0F01
transmitDataRate†	0F02
receiveDataRate†	0F03
rxLevelEstimate†	0F4E

V.23 Modulation Object Attributes

Object: modeV23

Tag-ID: 00

Description: This object displays the modems V.23 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: duplex

Tag-ID: 01

Description: This object indicated the duplex mode of operation.

Data type: BOOLEAN

Mandatory: Yes

Object: transmitDataRate

Tag-ID: 02

Description: This object indicates the data signalling rate of the local transmitter.

Data type: ENUMERATED

Table 13/V.59 – V.23 data rate codes

Mnemonic	Code Value	V.23 data signalling rate
dR1200	0	1200 bit/s
dR600	1	600 bit/s
dR75	2	75 bit/s

Mandatory: Yes

Object: receiveDataRate

Tag-ID: 03

Description: This object indicates the data signalling rate of the local transmitter.

Data type: ENUMERATED. See transmitDataRate object definition for the enumerated codes used.

Mandatory: Yes

6.8.10 V.21 Diagnostics

The object defined below is applicable to modulation V.21.

V.21 Modulation Objects

Object Identifier	Tag-ID
ModeV21	1000
v21TxLevel†	1045,1046,1047: This object uses the TxPowerLevel mid-level object
rxLevelEstimate†	104E

V.21 Modulation Object Attributes

Object: modeV21

Tag-ID: 00

Description: This object displays the modems V.21 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

6.8.11 V.34 half-duplex facsimile Diagnostics

The following are the objects define for use in V.34 half-duplex operation (clause 12/V.34).

V.34 HDX Modulation Objects

Object Identifier	Tag-ID
ModeV34H	1500
iNFO0_Tx†	1541
iNFO0_Rx†	1542
iNFOh	1501
mPh	1502
precodeCoeff†	1543
v34TxLevel†	1545,1546,1547: This object uses the TxPowerLevel mid-level object.
symbolRate†	1548
carrierFreq†	154A
dataHistory†	154C
rxLevelEstimate†	154E
noiseEstimate†	154F
rxSignalQuality†	1550
cCrate	1503

V.34 HDX Modulation Object Attributes

Object: modeV34H

Tag-ID: 00

Description: This object displays the modems V.34 half-duplex operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: iNFOh

Tag-ID: 01

Description: This object displays the INFO sequence either transmitted by the Source or received by the Recipient.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: mPh

Tag-ID: 02

Description: This object displays the MP sequence either transmitted by the Source or received by the Recipient.

NOTE – Precoder coefficients for type 1 MP sequences are not included.

Data type: IA5STRING:dataSequence

Mandatory: Yes

Object: cCrate

Tag-ID: 03

Description: This objects indicates the Control Channel Data rate used.

Data type: ENUMERATED

Table 14/V.59 – V.34 HDX Control Channel Data Rate Codes

Mnemonic	Code Value	Description
t12R12	0	The Control Channel is configured for 1200 bit/s operation in both transmit and receive directions
t24R24	1	The Control Channel is configured for 2400 bit/s operation in both transmit and receive directions
t12R24	2	The Control Channel has asymmetric rates, 1200 bit/s in the transmitter and 2400 bit/s in the receiver
t24R12	3	The Control Channel has asymmetric rates, 2400 bit/s in the transmitter and 1200 bit/s in the receiver

Mandatory: Yes

6.8.12 V.17 facsimile Diagnostics

The object defined below is applicable to modulation V.17.

V.17 Modulation Objects

Object Identifier	Tag-ID
modeV17	1600
V17TxLevel	1645,1646,1647: This object uses the TxPowerLevel mid-level object
rxLevelEstimate†	164E

V.17 Modulation Object Attributes

Object: modeV17

Tag-ID: 00

Description: This object displays the modems V.17 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

6.8.13 V.29 facsimile Diagnostics

The object defined below is applicable to modulation V.29.

V.29 Modulation Objects

Object Identifier	Tag-ID
modeV29	1700
v29TxLevel	1745,1746,1747: This object uses the TxPowerLevel mid-level object
rxLevelEstimate†	174E

V.29 Modulation Object Attributes

Object: modeV29

Tag-ID: 00

Description: This object displays the modems V.29 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

6.8.14 V.27 *ter* facsimile Diagnostics

The object defined below is applicable to modulation V.27 *ter.*

V.27 *ter* Modulation Objects

Object Identifier	Tag-ID
modeV27T	1800
v27TxLevel	1845,1846,1847: This object uses the TxPowerLevel mid-level object.
rxLevelEstimate†	184E

V.27 *ter* Modulation Object Attributes

Object: modeV27T

Tag-ID: 00

Description: This object displays the modems V.27 *ter* operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

6.8.15 V.18 Diagnostics

The objects defined below are applicable to text telephone modem V.18.

V.18 Modulation Objects

Object Identifier	Tag-ID
modeV18	2300
v18used	2301
v18TxLevel†	2345,2346,2347: This object uses the TxPowerLevel mid-level object
rxLevelEstimate†	234E

V.18 Modulation Object Attributes

Object: modeV18

Tag-ID: 00

Description: This object displays the modems V.18 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: v18used

Tag-ID: 01

Description: This object indicated the mode of operation achieved for V.18 in the last call. See ITU-T V.18 for the definitions of V.18 modes.

Data type: ENUMERATED

Table 15/V.59 – V.18 modulation mode codes

Mnemonic	Code Value	V.18 mode
None	0	No mode achieved
v18-V21Hi	1	native V.18 mode transmitting on the low channel for text only or text and voice alternating
v18-V21Lo	2	native V.18 mode transmitting on the low channel for text only or text and voice alternating
v18-V61C	3	native V.18 mode for text and voice simultaneously, transmitting in the caller's channel
v18-V61A	4	native V.18 mode for text and voice simultaneously, transmitting in the answering part's channel
v21Hi	5	legacy V.21 mode transmitting on the high channel
v21Lo	6	legacy V.21 mode transmitting on the low channel
dTMF	7	DTMF text telephone mode
eDT	8	EDT ("European Deaf Telephone")

Table 15/V.59 – V.18 modulation mode codes *(concluded)*

Mnemonic	Code Value	V.18 mode
baudot45	9	Baudot, 45.45 bits/s
baudot47	10	Baudot, undetermined bit rate
baudot50	11	Baudot 50 bits/s
b23Hi	12	V.23 modulation and Minitel coding transmitting on the high channel
v23Lo	13	V.23 modulation and Minitel coding transmitting on the low channel
bellHi	14	Bell 103, transmitting on the high channel
bellLo	15	Bell 103, transmitting on the low channel

Mandatory: Yes

6.8.16 V.70 DSVD Diagnostics

This class of objects is for further study.

6.8.17 V.61 ASVD Diagnostics

This class of objects is for further study.

6.8.18 Non-Standard Modem Diagnostics

This clause defines a class of customized object classes that may be used to provide diagnostic capabilities of proprietary modem modes.

Non-Standard Diagnostic Objects

Object Identifier	Tag-ID
proprietaryMode	2601
fieldData	2602

Non-Standard Diagnostic Object Attributes

Object: proprietaryMode

Tag-ID: 01

Description: This object contains the name of the proprietary mode or modes, whose diagnostics are contained in this customized object class.

Data type: IA5STRING:simpleText

Mandatory: No

Object: fieldData

Tag-ID: 02

Description: This object contains the delimited set of user-defined diagnostic objects.

Data type: IA5STRING:simpleText. The maximum size of this IA5 String is 256 octets.

Mandatory: No

Example: This example illustrates how this object class could be used. The example is a fictitious modem whose mode is called SuperTurbo+. The custom diagnostic objects are contained within the delimiter parentheses and is defined by the manufacturer.

Example:

proprietaryMode{SuperTurbo+}

FieldData{2300;FE6B;+1;0.34}

6.9 Connection Diagnostic

The following objects describe the statistics of a previous connection.

Connection Diagnostic Objects

Object Identifier	Tag-ID
txCarrier	2C01
rxCarrier	2C02
carrierLosses	2C03
renegRequests	2C04
renegSuccesses	2C05
retrainRequests	2C06
retrainSuccesses	2C07
callWaiting	2C08
duration	2C09
caller	2C0A
reverseRoles	2C0B

Connection Diagnostic Object Attributes

Object: txCarrier

Tag-ID: 01

Description: This object is a text string that is used in the same manner as defined in 6.4.1/V.250 and Table 13/V.250.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: rxCarrier

Tag-ID: 02

Description: This object performs the same function as described in txCarrier above and it uses the same definition.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: carrierLosses

Tag-ID: 03

Description: This is the number of times carrier was lost and then restored.

Data type: INTEGER value 0 to 255.

Mandatory: Yes

Object: renegRequests

Tag-ID: 04

Description: This is the number of times a rate renegotiation was requested by either end.

Data type: INTEGER value 0 to 255

Mandatory: Yes

Object: renegSuccesses

Tag-ID: 05

Description: This is the number of times a rate renegotiation was successfully completed.

Data type: INTEGER value 0 to 255

Mandatory: Yes

Object: retrainRequests

Tag-ID: 06

Description: Similar to renegRequests.

Data type: INTEGER value 0 to 255

Mandatory: Yes

Object: retrainSuccesses

Tag-ID: 07

Description: Is similar to renegSuccesses.

Data type: INTEGER value 0 to 255

Mandatory: Yes

Object: callWaiting

Tag-ID: 08

Description: Is a count of the number of call waiting events that occurred in the previous connection.

Data type: INTEGER value 0 to 255

Mandatory: No

Object: duration

Tag-ID: 09

Description: Is the duration of the call, with units specified (e.g. 00H15M27S). If the duration of the call exceeds the timing capability of the DCE, this overflow is indicated by the string ">T" where T is the maximum time capable of being measured by the DCE.

Data type: IA5STRING:simpleText

Mandatory: No

Object: caller

Tag-ID: 0A

Description: Is TRUE if the modem originated the previous call, FALSE otherwise.

Data type: BOOLEAN

Mandatory: Yes

Object: reverseRoles

Tag-ID: 0B

Description: This object returns TRUE if the modem reversed the actual roles of caller and answerer. An originating modem behaving as an answerer sets Caller to TRUE and ReverseRoles to TRUE.

Data type: Boolean

Mandatory: No

6.10 Line Conditions

This set of objects indicates the conditions of the PSTN channel or line. If any of these objects are provided, there is no constraint upon when the parameters are to be measured within the connection.

Line Condition Objects

Object Identifier	Tag-ID
lineProbeValues	2D01
phaseJitter	2D02
ampJitter	2D03
nLD	2D04

Line Condition Object Attributes

Object: lineProbeValues

Tag-ID: 01

Description: This object is a sequence of numbers representing the results of any line probe made during the connection. The values can be represented either as Real Magnitudes or as a Complex pair. Commas separate each item. For the complex the format is X ± JY. The order of the values in the list are the lowest frequency first and the highest frequency last.

Example:

lineProbeValues{FF03,03C0,…….} or lineProbeValues{-0.05+J0.4,0.33-J0.001,…..}

Data Type: IA5STRING:simpleText

Mandatory: No

Object: phaseJitter

Tag-ID: 02

Description: This object provides the modems estimate of phase jitter. The format consists of D@F where D represents the magnitude of the jitter and F is the Frequency of the jitter. Units should be provided if calibrated.

Example:

phaseJitter{3.2d@60H} or phaseJitter{0020@40}.

Data Type: IA5STRING:simpleText

Mandatory: No

Object: ampJitter

Tag-ID: 03

Description: This object provides an estimate of Amplitude Jitter if measured by the modem. The Format of the response is a single number with units if calibrated.

Example:

ampJitter{0.5dB} or ampJitter{00BC}.

Data Type: IA5STRING:simpleText

Mandatory: No

Object: nLD

Tag-ID: 04

Description: This parameter provides the modems estimate of Non-Linear Distortion if measures. The format of the object is a single number with units if calibrated.

Example:

nLD{2.1dB} or alternatively nDL{004C}.

Data Type: IA5STRING:simpleText

Mandatory: No

6.11 Call Progress

The intent of this class of objects is to provide details of the nature of the previous call made.

Call Progress Objects

Object Identifier	Tag-ID
callProgressResult	2E01
callTerminate	2E02

Call Progress Object Attributes

Object: callProgressResult

Tag-ID: 01

Description: This object indicates the result of the previous call and is represented by a single code.

Data type: ENUMERATED

Table 16/V.59 – Previous Call Progress Result Code

Mnemonic	Code	Description
NoPrevCall	0	No previous call recorded
NoDialTone	1	Network Dial tone not detected
NoRingBack	2	Network Ring back not detected
ReOrderTone	3	Network Reorder Tone detected
BusyTone	4	Network Busy Tone detected
VoiceDetected	5	Voice detected
ANSDetected	6	V.25 Answer tone detected
ANSamDetected	7	V.8 AM modulated Answer tone detected
v8bisDetected	8	V.8 *bis* detected
TxpDetected	9	Text telephone terminal detected
UnKnownSig	20	Unrecognized signal detected
Connected	40	DCE connected

Mandatory: Yes

Object: callTerminate

Tag-ID: 02

Description: This object indicates the reason for the termination of the last call made and is defined by a code.

Data type: ENUMERATED

Table 17/V.59 – Call Termination Result Codes

Mnemonic	Code	Description
causeUnidentified	0	Cause of call failure was unknown
dCEpowerLoss	20	If able to record this event, the DCE indicates that it was a power failure that caused the call failure to occur
equipmentFailure	21	If able to record this event, the DCE indicates that the cause of call termination was due to internal equipment failure
inactivityTimerExpired	25	Call failure caused due to an excess period of inactivity
cct108isOffInhibitsDial	31	V.24 circuit 108 is off, preventing dialling
cct108turnedOff	32	V.24 circuit 108 off
noNumberProvided	40	The call failed due to there being no number to dial
blacklistedNumber	41	Modem attempted to auto-dial a blacklisted number
callAttemptsLimitExceeded	42	Modem has exceeded the number of auto-dial attempts
extensionDeviceOffHook	43	If able to record this event, the DCE indicates that the cause of call termination was due to an extension device going off-hook
callSetupFailTimerExpired	44	Call set up timer has expired
incomingCallDetected	45	Outgoing failed due to an incoming call
loopCurrentInterrupted	46	If able to record this event, the DCE indicates that the cause of call termination was due to an interruption in the loop current

Table 17/V.59 – Call Termination Result Codes *(concluded)*

Mnemonic	Code	Description
noDialTone	47	Outgoing call failed due there being no dial tone detected
voiceDetected	48	Connection failed to voice being detected
reorderTone	49	Call failed due to the detection of the reorder tone
sitTone	50	Call failed due to the detection of the special information tone
engagedTone	51	Call progress failed due to the detection of the Engaged tone
longSpaceDisconnect	52	Modem disconnected using the long space criteria
carrierLost	53	Modem disconnected due to loss of carrier
trainingFailed	54	Modem disconnected due to failure in modem training
noModulationinCommon	56	Modems failed to connect due to there being no common modulation mode
retrainAttemptCountExceeded	64	Connection terminated due to retrain count exceeded
gstnCleardownReceived	65	Connection terminated with correct CLEARDOWN exchange
faxDetected	66	Connection terminated due to detecting facsimile terminal
protocolError	80	Connection terminated due to excess protocol errors
user Disconnect	90	Connection terminated due to user request
onHoldTimerExpired	100	Connection terminated because on-hold timer expired
onHoldRemoteDisc	101	Connection terminated because remote modem disconnected while on hold

Mandatory: Yes

6.12 Protocol Performance

The intent of this class of objects is to provide details of the protocols used during the last connection. Two mid-level objects make up this class of diagnostics. These are the Error Control and the Compression Diagnostics.

Error Control Objects

Object Identifier	Tag-ID
ModeV42	2F00
protocolNegotiation	2F01
featureNegotiation	2F02
txFrameSize	2F03
rxFrameSize	2F04
txWindow	2F05
rxWindow	2F06
linkTimeout	2F07
framesSentAck	2F08
framesRetransmitted	2F09
framesSentAck	2F0A
framesReceivedDiscard	2F0B
txErrors	2F0C
rxErrors	2F0D
txThroughput	2F0E
rxThroughput	2F0F

Error Control Object Attributes

Object: modeV42

Tag-ID: 00

Description: This object displays the modems V.42 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: protocolNegotiation

Tag-ID: 01

Description: This object returns a code indicating which error control protocol was negotiated in the previous connection.

Data type: ENUMERATED

Table 18/V.59 – Code values for Error Control Protocol

Mnemonic	Code	Description
Disabled	0	No error control protocol used
Lapm	1	LAPM used
v42annexA	2	V42/Annex A used

Mandatory: Yes

*Object: v42*featureNegotiation

Tag-ID: 02

Description: This object provides the V.42 features as a sequence of bit flags packed into a codeword.

Data type: BIT STRING

Table 19/V.59 – Code values for V.42 Error Control Feature Negotiation

Mnemonic	Bit	Description
SingleSREJ	0	Single selective reject procedure enabled
MultipleSREJ	1	Multiple selective reject procedure enabled
HDLCBalMode	2	Normal error-control procedure enabled
TestFrameSup	3	Indicates support of the TEST frame
fcs16	4	16-bit frame check sequence (FCS) enabled
fcs32	5	32-bit frame check sequence (FCS) enabled
v42phaseDetused	6	V.42 detection phase used
odpAdpbypassed	7	ODP/ADP procedure bypassed by other mechanism (e.g. V.92)
v42fallbackDisc	8	Disconnect if V.42 negotiation fails
v42fallbackBuf	9	Use V.14 buffered mode if V.42 negotiation fails
v42fallbackUnBuf	10	Use V.14 un-buffered mode if V.42 negotiation fails

Mandatory: Yes

Object: txFrameSize
Tag-ID: 03
Description: This object indicates the negotiated transmitter frame size in octets.
Data type: INTEGER value of 1 to 65536
Mandatory: Yes

Object: rxFrameSize
Tag-ID: 04
Description: This object indicates the negotiated receive frame size in octets.
Data type: INTEGER value of 1 to 65536
Mandatory: Yes

Object: txWindow
Tag-ID: 05
Description: This object is the size of the transmit window in octets.
Data type: INTEGER value of 1 to 65536
Mandatory: Yes

Object: rxWindow
Tag-ID: 06
Description: This object is the size of the receive window in octets
Data type: INTEGER value of 1 to 65536
Mandatory: Yes

Object: linkTimeout
Tag-ID: 07
Description: This is the number of timeouts incurred by the end of the connection.
Data type: INTEGER value of 0 to 511
Mandatory: No

Object: framesSentAck
Tag-ID: 08
Description: This object is similar to that used in 3.5/V.58 for V.42 and is the number of frames containing user data transmitted by the DCE for which acknowledgement was received.
Data type: INTEGER value of 0 to 65535
Mandatory: No

Object: framesRetransmitted

Tag-ID: 09

Description: This object is similar to that used in 3.5/V.58 for V.42 and is the number of frames containing user data retransmitted by the DCE.

Data type: INTEGER value of 0 to 65535

Mandatory: No

Object: framesReceivedAck

Tag-ID: 0A

Description: This object is similar to that used in 3.5/V.58 for V.42 and is the number of frames containing user data received by the DCE for which acknowledgement was sent.

Data type: INTEGER value of 0 to 65535

Mandatory: No

Object: framesReceivedDiscard

Tag-ID: 0B

Description: This object is similar to that used in 3.5/V.58 for V.42 and is the number of frames discarded.

Data type: INTEGER value of 0 to 65535

Mandatory: No

Object: txErrors

Tag-ID: 0C

Description: Is the number of REJ, SREJ or similar frames received from the remote end.

Data type: INTEGER value from 0 to 65535

Mandatory: No

Object: rxErrors

Tag-ID: 0D

Description: Is the number of errored frames received from the remote end. This includes bad FCS, unknown address, out of sequence, etc.

Data type: INTEGER value from 0 to 65535

Mandatory: No

Object: txThroughput

Tag-ID: 0E

Description: Is the average throughput measured at the input to the error control entity, rounded to the nearest Integer, and expressed in bits per second (bit/s).

Data type: INTEGER value from 0 to 16777215

Mandatory: No

Object: rxThroughput

Tag-ID: 0F

Description: Is similar to TxThroughput.

Data type: INTEGER value from 0 to 16777215

Mandatory: No

Compression Objects

modeV42B	**3000**
compressionNegotiationResult	3001
v42bisCompressionActive	3002
v42bisDictionarySize	3003
v42bisStringLength	3004
modeV44	300A
v44CompressionActive	300B
v44EncoderDictionarySize	300C
v44EncoderHistorySize	300D
v44EncoderStringLength	300E
v44DecoderDictionarySize	300F
v44DecoderHistorySize	3010
v44DecoderStringLength	3011

Compression Object Attributes

Object: modeV42B

Tag-ID: 00

Description: This object displays the modems V.42 *bis* operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: compressionNegotiationResult

Tag-ID: 01

Description: Code indicating resultant negotiated Compression used.

Data type: ENUMERATED

Table 20/V.59 – Codes for the Compression Negotiation Result

Mnemonic	Code	Description
None	0	No compression negotiated
v42bisOnly	1	Proposed V.42 *bis*/response V.42 *bis*
v42bisBoth	2	Proposed V.42 *bis* or V.44/response V.42 *bis*
v44Both	3	Proposed V.42 *bis* or V.44/response V.44
v44Only	4	Proposed V.44/response V.44
Other	5	Compression of an non-defined type used

Mandatory: Yes

Object: v42bisCompressionActive

Tag-ID: 02

Description: Contains the value of the V.42 *bis* data compression request parameter (P0) established for the last call. If ITU-T V.42 *bis* is not enabled for the current or past call, then the value will be set to None.

Data type: ENUMERATED

Table 21/V.59 – Codes for V.42 *bis* Compression request parameter (P0)

Mnemonic	Code	Description
none	0	V.42 *bis* not selected
initOnly	1	V.42 *bis* requested in initiator to responder direction only
respOnly	2	V.42 *bis* requested in responder to initiator direction only
both	3	V.42 *bis* requested in both directions

Mandatory: Yes

Object: v42bisDictionarySize

Tag-ID: 03

Description: Number of dictionary code words used in the previous call.

Data type: INTEGER value from 512 to 65536

Mandatory: No

Object: v42bisStringLength

Tag-ID: 04

Description: Is the maximum string length negotiated, in octets for the previous call.

Data type: Integer value from 6 to 250

Mandatory: No

Object: modeV44

Tag-ID: 0A

Description: This object displays the modems V.44 operating mode capability.

Data type: ENUMERATED (see Table 5 for Result codes)

Mandatory: Yes

Object: v44CompressionActive

Tag-ID: 0B

Description: Contains the value of the V.44 data compression request parameter (P0) established for the last call. If ITU-T V.44 is not enabled for the current or past call, then the value will be set to None.

Data type: ENUMERATED

Table 22/V.59 – Codes for V.44 Compression request parameter (P0)

Mnemonic	Code	Description
none	0	V.44 not selected
txOnly	1	V.44 requested in transmit direction only
rxOnly	2	V.44 requested in receive direction only
both	3	V.44 requested in both transmit and receive directions

Mandatory: Yes

Object: v44EncoderDictionarySize

Tag-ID: 0C

Description: Number of dictionary code words used by the encoder in the previous call.

Data type: INTEGER value from 256 to 65536.

Mandatory: No

Object: v44EncoderHistorySize

Tag-ID: 0D

Description: Length of the history buffer used by the encoder in the previous call.

Data type: INTEGER value from 512 to 1677215.

Mandatory: No

Object: v44EncoderStringLength

Tag-ID: 0E

Description: Is the maximum string length negotiated, in octets, used by the encoder for the previous call.

Data type: Integer value from 32 to 255.

Mandatory: No

Object: v44DecoderDictionarySize

Tag-ID: 0F

Description: Number of dictionary code words used by the decoder in the previous call.

Data type: INTEGER value from 256 to 65536.

Mandatory: No

Object: v44DecoderHistorySize

Tag-ID: 10

Description: Length of the history buffer used by the decoder in the previous call.

Data type: INTEGER value from 512 to 1677215.

Mandatory: No

Object: v44DecoderStringLength

Tag-ID: 11

Description: Is the maximum string length negotiated, in octets, used by the decoder for the previous call.

Data type: Integer value from 32 to 255.

Mandatory: No

6.13 DTE/DCE interface Diagnostics

The objective of this class of objects is to provide diagnostics on the DTE to DCE interface.

DTE/DCE Interface Diagnostic Objects

Object Identifier	Tag-ID
txFlowControl	3101
rxFlowControl	3102
protocol	3103
txSpeed	3104
rxSpeed	3105
txThroughput	3106
rxThroughput	3107
txErrors	3108
rxErrors	3109

DTE/DCE Interface Diagnostic Object Attributes

Object: txFlowControl

Tag-ID: 01

Description: A string indicating type of flow control used by the transmitter of the DCE. For example "XON/XOFF", "105/133" or "None" are all valid.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: rxFlowControl

Tag-ID: 02

Description: A string indicating type of flow control used by the Receiver of the DCE. Similar values as used by the previous object are valid.

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: protocol

Tag-ID: 03

Description: Is a text string describing the mode in which data are transferred between the DTE and the DCE (e.g. 8N1, 7E2, Autosync, X.32, V.80 framed, etc.).

Data type: IA5STRING:simpleText

Mandatory: Yes

Object: txSpeed

Tag-ID: 04

Description: Is the data rate from DTE to DCE in bits per second.

Data type: INTEGER value from 50 to 1677215.

Mandatory: Yes

Object: rxSpeed

Tag-ID: 05

Description: Is the data rate from DCE to DTE in bits per second.

Data type: INTEGER value from 50 to 16777215

Mandatory: Yes

Object: txThroughput

Tag-ID: 06

Description: Is the average data rate from DTE to DCE, in bits per second.

Data type: INTEGER value from 50 to 16777215

Mandatory: No

Object: rxThroughput

Tag-ID: 07

Description: Is the average data rate from DCE to DTE, in bits per second.

Data type: INTEGER value from 50 to 16777215

Mandatory: No

Object: txErrors

Tag-ID: 08

Description: Is the number of errored characters received from the DTE (e.g. overruns, framing errors, etc.).

Data type: INTEGER value from 0 to 65535

Mandatory: No

Object: rxErrors

Tag-ID: 09

Description: Is the number of characters known not to have been sent to the DTE.

Data type: INTEGER value from 0 to 65535

Mandatory: No

ANNEX A

ASN.1 Notation

This annex provides the ASN.1 notation in data abstract form.

```
V59 DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

V59String ::=IA5String(SIZE(1..40))

V59Objects ::= CHOICE
{
        modemIdentity           ModemIdentity,
        modeCapability          ModeCapability,
        modeSelected            ModeSelected,
         v8Diag                 V8Diag,
         v8bisDiag              V8bisDiag,
        v90Diag                 V90Diag,
        v91Diag                 V91Diag,
        v92ModDiag              V92ModDiag,
        v34Diag                 V34Diag,
        v32Diag                 V32Diag,
        v22Diag                 V22Diag,
        v23Diag                 V23Diag,
        v21Diag                 V21Diag,
        v34HdxDiag              V34HdxDiag,
        v17Diag                 V17Diag,
        v29faxDiag              V29faxDiag,
        v27tfaxDiag             V27tfaxDiag,
        v18Diag                 V18Diag,
        v70Diag                 V70Diag,
        v61Diag                 V61Diag,
        nSMDiag                 NSMDiag,
        cnxDiag                 CnxDiag,
        lineConDiag             LineConDiag,
        callProgress            CallProgress,
        protocol                Protocol,
```

```
        errorControl            ErrorControl,
        compression             Compression,
        dteDce                  DTEDCE,
        ...
}

ModemIdentity ::=SEQUENCE OF SEQUENCE
{
        manufacturer            V59String,
        model                   V59String,
        firmwareVersion         V59String,
        diagnosticVersion       V59String,
        multimediaMode          BIT STRING
        {
                dataMode        (1),
                faxT30C         (2),
                fAXT30F         (3),
                voiceV253       (4),
                sVDV70          (5),
                sVDV61          (6),
                vidTelH324      (7),
                v80other        (8),
                v18Text         (9)
        } OPTIONAL,
        ...
}

ModeCapability::= SEQUENCE OF SEQUENCE
{
        modulationMode    SEQUENCE
        {
                modeV90         Capabilities,
                modeV91         Capabilities,
                modeV92         Capabilities,
                modeV34         Capabilities,
                modeV32B        Capabilities,
                modeV22B        Capabilities,
                modeV23         Capabilities,
                modeV21         Capabilities,
                modeV34H        Capabilities,
                modeV17         Capabilities,
                modeV29         Capabilities,
                modeV27T        Capabilities,
                modeV18         Capabilities,
                otherModMode    V59String OPTIONAL,
                ...
        },
        protocolmode        SEQUENCE
        {
                modeV42         Capabilities,
                modeV42B        Capabilities,
                modeV44         Capabilities,
                otherProtMode   V59String OPTIONAL,
                ...
        },
        ...

}

ModeSelected::= SEQUENCE
{
        modulationHistory       V59String
}
```

```
V8Diag::= SEQUENCE OF SEQUENCE
{
        ci                      V59String,
        cm                      V59String,
        jm                      V59String,
        v8Result        ENUMERATED
        {
                v8ComNeg        (0),
                v8NoComNeg      (1),
                v8NoNegAuto     (2)
        }
}

V8bisDiag::= SEQUENCE OF SEQUENCE
{
        v8bisTransaction        INTEGER (1..13),
        mR SEQUENCE
        {
                mRtype          V8bisSignalType,
                mRSequence      V59String
        },
        cR SEQUENCE
        {
                cRtype          V8bisSignalType,
                cRSequence      V59String
        },

        cL SEQUENCE
        {
                cLSequence      V59String
        },

        cLR SEQUENCE
        {
                cLRSequence     V59String
        },

        eS SEQUENCE
        {
                eSType CHOICE
                {
                        nONE    INTEGER (0),
                        i       INTEGER (1),
                        r       INTEGER (2)
                },
                eSSequence      V59String
        },

        ackNak SEQUENCE
        {
                aCK CHOICE
                {
                        nONE    INTEGER (0),
                        aCK1    INTEGER (1),
                        aCK2    INTEGER (2)
                },
                nAK CHOICE
                {
                        none    INTEGER (0),
                        nAK1    INTEGER (1),
                        nAK2    INTEGER (2),
                        nAK3    INTEGER (3),
```

```
                nAK4        INTEGER (4)
            }
        }
}

V90Diag::= SEQUENCE OF SEQUENCE
{
        modeV90             Capabilities,
        iNFO0Tx             V59String,
        iNFO0Rx             V59String,
        iNFO1A              V59String,
        iNFO1d              V59String,
        mP                  V59String,
        cP                  V59String,
        cPt                 V59String,
        jAsequence          V59String,
        jDsequence          V59String,
        v90TxLevel          TxPowerLevel,
        txSymbolRate        SymbolRate OPTIONAL,
        rxSymbolRate        SymbolRate OPTIONAL,
        txCarrierFreq       CarrierFreq OPTIONAL,
        rxCarrierFreq       CarrierFreq OPTIONAL,
        txDataHistory       V59String,
        rxDataHistory       V59String,
        rxLevelEstimate     V59String OPTIONAL,
        noiseEstimate       V59String OPTIONAL,
        rxSignalQuality     V59String OPTIONAL,
        nearEchoEstimate    V59String OPTIONAL,
        farEchoEstimate     V59String OPTIONAL,
        roundTripDelay      V59String OPTIONAL,
        rBSpattern          RobBitPatt OPTIONAL,
        digitalPadLoss      V59String OPTIONAL,
        codecLaw            BOOLEAN OPTIONAL,
        ...
}

V91Diag::= SEQUENCE OF SEQUENCE
{
        modeV91             Capabilities,
        iNFO0TX             V59String,
        iNFO0RX             V59String,
        cP                  V59String,
        controlChannel      BOOLEAN,
        v91TxPowerLevel     TxPowerLevel,
        transparentMode     BOOLEAN,
        txDataHistory       V59String,
        rxDataHistory       V59String,
        noiseEstimate       V59String OPTIONAL,
        rxSignalQuality     V59String OPTIONAL,
        rBSpattern          RobBitPatt OPTIONAL,
        digitalPadLoss      V59String OPTIONAL,
        localCodecLaw       BOOLEAN OPTIONAL,
        remoteCodecLaw      BOOLEAN OPTIONAL,
        frameSlipsDetected  INTEGER (0..256) OPTIONAL,
        ...
}

V92ModDiag::= SEQUENCE OF SEQUENCE
{
        modeV92             Capabilities,
        iNFO0d              V59String,
        iNFO0a              V59String,
        iNFO1d              V59String,
```

```
        iNFO1a-US               V59String,
        iNFO1a-SP2              V59String,
        jA                      V59String,
        jD                      V59String,
        jDP                     V59String,
        cP                      V59String,
        v92TxLevel              TxPowerLevel,
        v92SymbolRate           SymbolRate OPTIONAL,
        txDataHistory           V59String,
        rxDataHistory           V59String,
        rxLevelEstimate         V59String OPTIONAL,
        noiseEstimate           V59String OPTIONAL,
        rxSignalQuality         V59String OPTIONAL,
        nearEchoEstimate        V59String OPTIONAL,
        farEchoEstimate         V59String OPTIONAL,
        roundTripDelay          V59String OPTIONAL,
        rBSpattern              RobBitPatt OPTIONAL,
        digitalPadLoss          V59String OPTIONAL,
        localCodecLaw           BOOLEAN OPTIONAL,
        remoteCodecLaw          BOOLEAN OPTIONAL,
        sP1Enable               BOOLEAN,
        lastCallSP1             BOOLEAN,
        qC1a                    IA5String(SIZE(1..10)),
        qC1d                    IA5String(SIZE(1..10)),
        qC2a                    IA5String(SIZE(1..2)),
        qC2d                    IA5String(SIZE(1..2)),
        qCA1a                   IA5String(SIZE(1..10)),
        qCA1d                   IA5String(SIZE(1..10)),
        qCA2a                   IA5String(SIZE(1..2)),
        qCA2d                   IA5String(SIZE(1..2)),
        cWDEnable               BOOLEAN,
        mOHEnable               BOOLEAN,
        lastMOHaction           ENUMERATED
        {
                mohaccepted     (0),
                mohdeclined     (1),
                mohdenied       (2),
                mohclrdn        (3),
                ...
        },
        mohTimeout              V59String,
        lastMOHduration         V59String,
        mHreq                   IA5String(SIZE(1..5)),
        mHack                   IA5String(SIZE(1..5)),
        mHnak                   IA5String(SIZE(1..5)),
        mHcld                   IA5String(SIZE(1..5)),
        mHcda                   IA5String(SIZE(1..5)),
        mHfrr                   IA5String(SIZE(1..5)),
        ...
}

V34Diag::= SEQUENCE OF SEQUENCE
{
        modeV34                 Capabilities,
        iNFO0Tx                 V59String,
        iNFO0Rx                 V59String,
        iNFO1c                  V59String,
        iNFO1a                  V59String,
        mPTx                    V59String,
        mPRx                    V59String,
        precodeCoeffTx          V59String OPTIONAL,
        precodeCoeffRx          V59String OPTIONAL,
        v34TxLevel              TxPowerLevel,
```

```
        txSymbolRate            SymbolRate OPTIONAL,
        rxSymbolRate            SymbolRate OPTIONAL,
        txCarrierFreq           CarrierFreq OPTIONAL,
        rxCarrierFreq           CarrierFreq OPTIONAL,
        txDataHistory           V59String,
        rxDataHistory           V59String,
        rxLevelEstimate         V59String OPTIONAL,
        noiseEstimate           V59String OPTIONAL,
        rxSignalQuality         V59String OPTIONAL,
        nearEchoEstimate        V59String OPTIONAL,
        farEchoEstimate         V59String OPTIONAL,
        roundTripDelay          V59String OPTIONAL,
        ...
}

V32Diag::= SEQUENCE OF SEQUENCE
{
        modeV32B                Capabilities,
        v32TxLevel              TxPowerLevel,
        v32txRateSeq            V59String,
        v32rxRateSeq            V59String,
        txDataHistory           V59String,
        rxDataHistory           V59String,
        rxLevelEstimate         V59String OPTIONAL,
        noiseEstimate           V59String OPTIONAL,
        rxSignalQuality         V59String OPTIONAL,
        nearEchoEstimate        V59String OPTIONAL,
        farEchoEstimate         V59String OPTIONAL,
        roundTripDelay          V59String OPTIONAL,
        ...
}

V22Diag::= SEQUENCE OF SEQUENCE
{
        modeV22B                Capabilities,
        dataRateHistory         V59String,
        v22TxLevel              TxPowerLevel,
        v22RateSeq              V59String,
        rxLevelEstimate         V59String OPTIONAL,
        noiseEstimate           V59String OPTIONAL,
        rxSignalQuality         V59String OPTIONAL,
        ...
}

V23Diag::= SEQUENCE OF SEQUENCE
{
        modeV23                 Capabilities,
        v23TxLevel              TxPowerLevel,
        duplex                  BOOLEAN,
        transmitDataRate        V23Drate,
        receiveDataRate         V23Drate,
        rxLevelEstimate         V59String OPTIONAL,
        ...
}

V21Diag::= SEQUENCE OF SEQUENCE
{
        modeV21                 Capabilities,
        v21TxLevel              TxPowerLevel,
        rxLevelEstimate         V59String OPTIONAL
}
```

```
V34HdxDiag::= SEQUENCE OF SEQUENCE
{
        modeV34H                Capabilities,
        iNFO0Tx                 V59String,
        iNFO0Rx                 V59String,
        iNFOh                   V59String,
        mPh                     V59String,
        precodeCoeffTx          V59String OPTIONAL,
        v34TxLevel              TxPowerLevel,
        symbolRate              V59String OPTIONAL,
        carrierFreq             V59String OPTIONAL,
        dataHistory             V59String OPTIONAL,
        rxLevelEstimate         V59String OPTIONAL,
        noiseEstimate           V59String OPTIONAL,
        rxSignalQuality         V59String OPTIONAL,
        cCrate                  ENUMERATED
        {
                t12R12          (0),
                t24R24          (1),
                t12R24          (2),
                t24R12          (3)
        },
        ...
}

V17Diag::= SEQUENCE OF SEQUENCE
{
        modeV17                 Capabilities,
        v17TxLevel              TxPowerLevel,
        rxLevelEstimate         V59String OPTIONAL,
        ...
}

V29faxDiag::= SEQUENCE OF SEQUENCE
{
        modeV29                 Capabilities,
        v29TxLevel              TxPowerLevel,
        rxLevelEstimate         V59String OPTIONAL,
        ...
}

V27tfaxDiag::= SEQUENCE OF SEQUENCE
{
        modeV27T                Capabilities,
        v27TxLevel              TxPowerLevel,
        rxLevelEstimate         V59String OPTIONAL,
        ...
}

V18Diag::= SEQUENCE OF SEQUENCE
{
        modeV18                 Capabilities,
        v18used                 ENUMERATED
        {
                none            (0),
                v18-V21Hi       (1),
                v18-V21Lo       (2),
                v18-V61C        (3),
                v18-V61A        (4),
                v21Hi           (5),
                v21Lo           (6),
                dTMF            (7),
                eDT             (8),
```

```
                baudot45        (9),
                baudot47        (10),
                baudot50        (11),
                v23Hi           (12),
                v23Lo           (13),
                bellHi          (14),
                bellLo          (15)
        },
        v18TxLevel              TxPowerLevel,
        rxLevelEstimate         V59String OPTIONAL,
        ...
}

V70Diag::= SEQUENCE OF SEQUENCE
{
        ...
}


V61Diag::= SEQUENCE OF SEQUENCE
{
        ...
}

NSMDiag::= SEQUENCE OF SEQUENCE
{
        proprietaryMode         V59String OPTIONAL,
        fieldData               IA5String(SIZE(1..256)) OPTIONAL,
        ...
}

CnxDiag::= SEQUENCE OF SEQUENCE
{
        txCarrier               V59String,
        rxCarrier               V59String,
        carrierLosses           INTEGER (0..256),
        renegRequests           INTEGER (0..256),
        renegSuccesses          INTEGER (0..256),
        retrainRequests         INTEGER (0..256),
        retrainSuccesses        INTEGER (0..256),
        callWaiting             INTEGER (0..256) OPTIONAL,
        duration                V59String OPTIONAL,
        caller                  BOOLEAN,
        reverseRoles            BOOLEAN OPTIONAL,
        ...
}

LineConDiag::= SEQUENCE OF SEQUENCE
{
        lineProbeValues         V59String OPTIONAL,
        phaseJitter             V59String OPTIONAL,
        ampJitter               V59String OPTIONAL,
        nLD                     V59String OPTIONAL,
        ...
}

CallProgress::= SEQUENCE OF SEQUENCE
{
        callProgressResult      CallProgressRes,
        callTerminate           CallTerminateCode,
        ...
}
```

```
Protocol::= SEQUENCE OF SEQUENCE
{
        errorControlStat        ErrorControl,
        compStat                Compression,
        ...
}

ErrorControl::= SEQUENCE OF SEQUENCE
{
        protocolNegotiation     ENUMERATED
        {
                disabled        (0),
                lapm            (1),
                v42annexA       (2),
                ...

        },
        v42featureNegotiation   BIT STRING
        {
                singleSREJ      (0),
                multipleSREJ    (1),
                hDLCBalMode     (2),
                testFrameSup    (3),
                fcs16           (4),
                fcs32           (5),
                v42phaseDetused (6),
                odpAdpbypassed  (7),
                v42fallbackDisc (8),
                v42fallbackBuf  (9),
                v42fallbackUnBuf (10)
        },
        txFrameSize             INTEGER (1..65535),
        rxFrameSize             INTEGER (1..65535),
        txWindow                INTEGER (1..65535),
        rxWindow                INTEGER (1..65535),
        linkTimeout             INTEGER (0..511) OPTIONAL,
        ecStatistics            SEQUENCE
        {
                framesSentAck   INTEGER(0..65535) OPTIONAL,
                framesRetransmitted     INTEGER(0..65535) OPTIONAL,
                framesReceivedAck       INTEGER(0..65535) OPTIONAL,
                framesReceivedDiscard   INTEGER(0..65535) OPTIONAL
        },
        txErrors                INTEGER(0..65535) OPTIONAL,
        rxErrors                INTEGER(0..65535) OPTIONAL,
        txThroughput            INTEGER(0..32767) OPTIONAL,
        rxThroughput            INTEGER(0..32767) OPTIONAL,
        ...
}

Compression::= SEQUENCE OF SEQUENCE
{
        compressionNegotiationResult    ENUMERATED
        {
                none            (0),
                v42bisOnly      (1),
                v42bisBoth      (2),
                v44Both         (3),
                v44Only         (4),
                other           (5)
        },
        modeV42B                Capabilities
```

```
        v42bisCompressionActive ENUMERATED
        {
                none                    (0),
                initOnly                (1),
                respOnly                (2),
                both                    (3)
        },
        v42bisDictionarySize    INTEGER (512..65535) OPTIONAL,
        v42bisStringLength      INTEGER (6..250) OPTIONAL,
        modeV44                 Capabilities,
        v44CompressionActive    ENUMERATED
        {
                none                    (0),
                txOnly                  (1),
                rxOnly                  (2),
                both                    (3)
        },
        v44EncoderDictionarySize INTEGER (256..65535) OPTIONAL,
        v44EncoderHistorySize   INTEGER (512.. 1677215) OPTIONAL,
        v44EncoderStringLength  INTEGER (32..255) OPTIONAL,
        v44DecoderDictionarySizeINTEGER (256..65535) OPTIONAL,
        v44DecoderHistorySize   INTEGER (512.. 1677215) OPTIONAL,
        v44DecoderStringLength  INTEGER (32..255) OPTIONAL,
        ...
}

DTEDCE::= SEQUENCE OF SEQUENCE
{
        txFlowControl           V59String,
        rxFlowControl           V59String,
        protocol                V59String,
        txSpeed                 INTEGER (50..1677215),
        rxSpeed                 INTEGER (50..1677215),
        txThroughput            INTEGER (50..1677215),
        rxThroughput            INTEGER (50..1677215),
        txErrors                INTEGER (0.. 65535),
        rxErrors                INTEGER (0.. 65535),
        ...
}

Capabilities                    ::=CHOICE
{
        notSupported            INTEGER (0),
        default                 INTEGER (1),
        enabled                 INTEGER (2),
        disabled                INTEGER (3)
}

V8bisSignalType    ::=CHOICE
{
        nONE                    INTEGER (0),
        e                       INTEGER (1),
        d                       INTEGER (2)
}

RobBitPatt          ::=BIT STRING
{
        robBit0                 (0),
        robBit1                 (1),
        robBit2                 (2),
        robBit3                 (3),
        robBit4                 (4),
        robBit5                 (5)
}
```

```
SymbolRate          ::= CHOICE
{
        sR8000                  INTEGER (0),
        sR3429                  INTEGER (1),
        sR3200                  INTEGER (2),
        sR3000                  INTEGER (3),
        sR2800                  INTEGER (4),
        sR2743                  INTEGER (5),
        sR2400                  INTEGER (6),
        sR1600                  INTEGER (7),
        sR1200                  INTEGER (8),
        sR600                   INTEGER (9),
        ...
}

CarrierFreq         ::= CHOICE
{
        cF1200                  INTEGER (0),
        cF1600                  INTEGER (1),
        cF1646                  INTEGER (2),
        cF1680                  INTEGER (3),
        cF1700                  INTEGER (4),
        cF1800                  INTEGER (5),
        cF1829                  INTEGER (6),
        cF1867                  INTEGER (7),
        cF1920                  INTEGER (8),
        cF1959                  INTEGER (9),
        cF2000                  INTEGER (10),
        cF2400                  INTEGER (11),
        ...
}

TxPowerLevel::= SEQUENCE OF SEQUENCE
{
        initialTxPower          V59String,
        txPowerDrop             V59String OPTIONAL,
        txPower                 V59String OPTIONAL,
        ...
}

ONOFF                   ::= CHOICE
{
        dISABLED                INTEGER (0),
        eNABLED                 INTEGER (1)
}

V23Drate            ::= CHOICE
{
        dR1200                  INTEGER (0),
        dR600                   INTEGER (1),
        dR75                    INTEGER (2)
}

CallProgressRes             ::= CHOICE
{
        noPrevCall              INTEGER (0),
        noDialTone              INTEGER (1),
        noRingBack              INTEGER (2),
        reOrderTone             INTEGER (3),
        busyTone                INTEGER (4),
        voiceDetected           INTEGER (5),
        aNSdetected             INTEGER (6),
```

```
        aNSamDetected           INTEGER (7),
        v8bisDetected           INTEGER (8),
        txpDetected             INTEGER (9),
        unKnownSig              INTEGER (20),
        connected               INTEGER (40),
        ...
}

CallTerminateCode       ::= CHOICE
{
        causeUnidentified       INTEGER (0),
        dCEpowerLoss            INTEGER (20),
        equipmentFailure        INTEGER (21),
        inactivityTimerExpired  INTEGER (25),
        cct108isOffInhibitsDial  INTEGER (31),
        cct108turnedOff         INTEGER (32),
        noNumberProvided        INTEGER (40),
        blacklistedNumber       INTEGER (41),
        callAttemptsLimitExceeded INTEGER (42),
        extensionDeviceOffHook  INTEGER (43),
        callSetupFailTimerExpired INTEGER (44),
        incomingCallDetected    INTEGER (45),
        loopCurrentInterrupted  INTEGER (46),
        noDialTone              INTEGER (47),
        voiceDetected           INTEGER (48),
        reorderTone             INTEGER (49),
        sitTone                 INTEGER (50),
        engagedTone             INTEGER (51),
        longSpaceDisconnect     INTEGER (52),
        carrierLost             INTEGER (60),
        trainingFailed          INTEGER (61),
        noModulationinCommon    INTEGER (62),
        retrainFailed           INTEGER (63),
        retrainAttemptCountExceeded INTEGER(64),
        gstnCleardownReceived   INTEGER (65),
        faxDetected             INTEGER (66),
        protocolError           INTEGER (80),
        userDisconnect          INTEGER (90),
        onHoldTimerExpired      INTEGER (100),
        onHoldRemoteDisc        INTEGER (101),
        ...
}

END
```

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	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

Geneva, 2001