



INTERNATIONAL TELECOMMUNICATION UNION

**TELECOMMUNICATION  
STANDARDIZATION SECTOR**

STUDY PERIOD 2001 - 2004

**COM 16-R 7-E  
December 2000  
Original: English**

---

Question(s): D, F, G/16; 1, 2, 3, 4 and 5/16

Texte disponible seulement en  
Text available only in  
Texto disponible solamente en } **E**

### **STUDY GROUP 16 – REPORT R 7**

SOURCE\*: Study Group 16 (Geneva meeting, 13-17 November 2000)

TITLE: Implementor's Guide for ITU-T Recommendation H.248

---

---

\* **Contact:** TSB

Tel: +41 22 730 5860

Fax: +41 22 730 5853

Email: [tsbsg16@itu.int](mailto:tsbsg16@itu.int)

**Attention:** This is not a publication made available to the public, but an **internal ITU-T Document** intended only for use by the Member States of the ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of the ITU-T.

## **Implementor's Guide for ITU-T Recommendation H.248**

### **Contact Information**

ITU-T Study Group 16 /  
Question 14 Rapporteur

Glen Freundlich  
Avaya Communications  
1300 W. 120<sup>th</sup> Avenue  
Westminster, Colorado 80234  
USA

Tel: +1 303 538 2899  
Fax: +1 303 538 3907  
E-mail: [ggf@avaya.com](mailto:ggf@avaya.com)

Implementors' Guide ITU-T  
Recommendation H.248  
Editor

Christian Groves  
Ericsson Australia  
37/360 Elizabeth St  
Melbourne, Victoria 3000  
Australia

Tel: +61 3 9301 6116  
Fax: +61 3 9301 1499  
E-mail: [Christian.Groves@ericsson.com](mailto:Christian.Groves@ericsson.com)

## **1 Introduction**

This document is a compilation of reported defects identified with the 2000 decided edition of ITU-T Recommendation H.248. It must be read in conjunction with the Recommendation to serve as an additional authoritative source of information for implementors'. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.248 Recommendations.

## **2 Scope**

This guide resolves defects in the following categories:

- editorial errors
- technical errors, such as omissions and inconsistencies
- ambiguities

In addition, the Implementors' Guide may include explanatory text found necessary as a result of interpretation difficulties apparent from the defect reports.

This Guide will not address proposed additions, deletions, or modifications to the Recommendations that are not strictly related to implementation difficulties in the above categories. Proposals for new features should be made in through contributions to the ITU-T.

## **3 Defect Resolution Procedure**

Upon discovering technical defects with any components of the H.248 Recommendation, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q14/16 Rapporteur. The template for a defect report is located at the end of the Guide. Contact information for these parties is included at the front of the document. Return contact information should also be supplied so a dialogue can be established to resolve the matter and an appropriate reply to the defect report can be conveyed. This defect resolution process is open to anyone interested in H.248 Recommendation. Formal membership in the ITU is not required to participate in this process.

## **4 References**

ITU-T Recommendation H.248 (2000), *Media Gateway Control Protocol*

## **5 Nomenclature**

In addition to traditional revision marks, the following marks and symbols are used to indicate to the reader how changes to the text of a Recommendation should be applied:

<b>Symbol</b>	<b>Description</b>
<hr/> <b>[Begin Correction]</b> <hr/>	Identifies the start of revision marked text based on extractions from the published Recommendations affected by the correction being described.
<hr/> <b>[End Correction]</b> <hr/>	Identifies the end of revision marked text based on extractions from the published Recommendations affected by the correction being described.
...	Indicates that the portion of the Recommendation between the text appearing before and after this symbol has remained unaffected by the correction being described and has been omitted for brevity.
--- <i>SPECIAL INSTRUCTIONS</i> --- {instructions}	Indicates a set of special editing instructions to be followed.

## 6 Technical and Editorial Corrections to ITU-T Recommendation H.248 (2000)

### 6.1 Correction in bibliographic reference

<b>Description:</b>	Section 2.1/H.248 contains a bibliographic reference to Q.765. The Q.765 series consists of a number of recommendations. The correct reference is Q.765.5, Application transport mechanism – Bearer independent call control (BICC), instead of the entire series.
---------------------	--

---

**[Begin Addition]**

---

#### 2.1 Normative references

...

~~ITU-T Recommendation Q.765, "Signalling System No. 7 – Application transport mechanism".~~

ITU-T Recommendation Q.765.5, "Application transport mechanism – Bearer independent call control (BICC)".

...

---

**[End Addition]**

---

The reference to Q.765 in section C.1/H.248 should be corrected to:

---

**[Begin Correction]**

---

ACodec	1006	Octet String	Audio Codec Type:
--------	------	--------------	-------------------

			Reference: ITU-T Rec. Q.765.5. — <del>Application transport mechanism</del> Non-ITU codecs are defined with the appropriate standards organisation under a defined Organizational identifier
--	--	--	--

---

*[End Correction]*

## 6.2 Valid parameters to Add, Modify and Move commands

<b>Description:</b>	The EventBufferDescriptor parameter was inadvertently omitted as a valid parameter to Add, Modify and Move commands in sections 7.2.1, 7.2.2, 7.2.4/H.248.
---------------------	--

---

*[Begin Addition]*

### 7.2.1 Add

The Add command adds a Termination to a Context.

TerminationID

[, MediaDescriptor]

[, ModemDescriptor]

[, MuxDescriptor]

[, EventsDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor}

[, ObservedEventsDescriptor]

[, EventBufferDescriptor]

[, StatisticsDescriptor]

[, PackagesDescriptor]

Add( TerminationID

    [, MediaDescriptor]

    [, ModemDescriptor]

    [, MuxDescriptor]

    [, EventsDescriptor]

    [, EventBufferDescriptor]

    [, SignalsDescriptor]

    [, DigitMapDescriptor]

    [, AuditDescriptor]

)

---

*[End Addition]*

---

*[Begin Addition]*

...

The EventsDescriptor parameter is optional. If present, it provides the list of events that should be detected on the Termination.

The EventBufferDescriptor parameter is optional. If present, it provides the list of events that the MG is requested to detect and buffer when EventBufferControl equals LockStep.

...

---

*[End Addition]*

---

*[Begin Addition]*

---

### 7.2.2 Modify

The Modify command modifies the properties of a Termination.

TerminationID

[, MediaDescriptor]

[, ModemDescriptor]

[, MuxDescriptor]

[, EventsDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor}

[, ObservedEventsDescriptor]

[, EventBufferDescriptor]

[, StatisticsDescriptor]

[, PackagesDescriptor]

Modify( TerminationID

[, MediaDescriptor]

[, ModemDescriptor]

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

)

---

*[End Addition]*

---

*[Begin Addition]*

---

### 7.2.4 Move

...

TerminationID

[, MediaDescriptor]

[, ModemDescriptor]

[, MuxDescriptor]

[, EventsDescriptor]

```
[, SignalsDescriptor]  
[, DigitMapDescriptor}  
[, ObservedEventsDescriptor]  
[, EventBufferDescriptor]  
[, StatisticsDescriptor]  
[, PackagesDescriptor]  
  Move( TerminationID  
        [, MediaDescriptor]  
        [, ModemDescriptor]  
        [, MuxDescriptor]  
        [, EventsDescriptor]  
        [, EventBufferDescriptor]  
        [, SignalsDescriptor]  
        [, DigitMapDescriptor]  
        [, AuditDescriptor]  
  )
```

---

*[End Addition]*

### 6.3 Cold Start

<b>Description:</b>	Section 11.2/H.248 contains a leftover from old terminology: Transaction Accept instead of Transaction Reply.
---------------------	---

---

*[Begin Correction]*

#### 11.2 Cold Start

A MG is pre-provisioned by a management mechanism outside the scope of this protocol with a Primary and (optionally) an ordered list of Secondary MGCs. Upon a cold start of the MG, it will issue a ServiceChange command with a "Restart" method, on the Root Termination to its primary MGC. If the MGC accepts the MG, it will send a Transaction ~~Accept~~ Reply, with the ServiceChangeMgcId set to itself.

...

---

*[End Correction]*

### 6.4 Digit map syntax

<b>Description:</b>	Annex A.3/H.248 provides a copy of the syntax of digit maps, stating that the definition given in Annex B/H.248 takes precedence in case of discrepancies. A discrepancy occurs in the production rule for digitStringList, and it is proposed to correct this discrepancy: the quoted forward slash should be replaced by a quoted vertical bar.
---------------------	---

*[Begin Correction]*

---

### A.3 Digit maps and path names

...  
digitStringList = digitString \*(LWSP "✓|" LWSP digitString)

*[End Correction]*

---

## 6.5 Omission in specification of text encoding

<b>Description:</b>	The specification of the text encoding of H.248 messages currently allows multiple occurrences of the same servicechange parameter, while the intention is that every such parameter should occur only once. The proposed resolution is to add a comment to the ABNF indicating this restriction.
---------------------	---

*[Begin Correction]*

---

### B.2 ABNF specification

...  
; each parameter at-most-once  
serviceChangeParm = (serviceChangeMethod / serviceChangeReason /  
serviceChangeDelay / serviceChangeAddress /  
serviceChangeProfile / extension / TimeStamp /  
serviceChangeMgcId / serviceChangeVersion )

*[End Correction]*

---

## 6.6 Ambiguity in text encoding

<b>Description:</b>	The text encoding as specified in Annex B/H.248 contains an ambiguity because the token "EB" was inadvertently used twice, in the production rules for EmbedToken and EventBufferToken. It is proposed to change the short tokens in the production rules for EmbedToken and EmergencyToken, and leave the production rule for EventBufferToken unchanged.
---------------------	--

*[Begin Correction]*

---

### B.2 ABNF specification

...  
EmbedToken = ("Embed" / "EMEB")  
EmergencyToken = ("Emergency" / "EGEM")

*[End Correction]*

---



## 6.7 Use of ServiceChange for MG registration

<b>Description:</b>	There is an inconsistency between section 7.2.8 and section 11.2 on when the ServiceChangeMgcID is used.
---------------------	--

*[Begin Correction]*

---

### 11.2 Cold Start

...

A MG is pre-provisioned by a management mechanism outside the scope of this protocol with a Primary and (optionally) an ordered list of Secondary MGCs. Upon a cold start of the MG, it will issue a ServiceChange command with a "Restart" method, on the Root Termination to its primary MGC. If the MGC accepts the MG's registration, ~~it will send a Transaction Accept, with the ServiceChangeMgcId set to itself. If the MG receives an ServiceChangeMgcId not equal to the MGC it contacted, it sends a ServiceChange to the MGC specified in the ServiceChangeMgcId. it sends a Transaction Reply not including a ServiceChangeMgcId parameter. If the MGC does not accept the MG's registration, it sends a Transaction Reply, providing the address of an alternate MGC to be contacted by including a ServiceChangeMgcId parameter.~~

If the MG receives a Transaction Reply that includes a ServiceChangeMgcId parameter, it sends a ServiceChange to the MGC specified in the ServiceChangeMgcId. It continues this process until it gets a controlling MGC to accept its registration, or it fails to get a reply. Upon failure to obtain a reply, either from the Primary MGC, or a designated successor, the MG tries its pre-provisioned Secondary MGCs, in order. If the MG is unable to establish a control relationship with any MGC, it shall wait a random amount of time as described in section 9.2 and then start contacting its primary, and if necessary, its secondary MGCs again.

It is possible that the reply to a ServiceChange with Restart will be lost, and a command will be received by the MG prior to the receipt of the ServiceChange response. The MG shall issue error 505 – Command Received before Restart Response.

*[End Correction]*

---

## 6.8 Echo cancellation parameters

<b>Description:</b>	<p>Appendix E.13.1/H.248 contains a property to turn echo cancellation off or on. In addition, C.1 and C.9 contain codepoints dealing with echo cancellation. The codepoints are</p> <ul style="list-style-type: none"><li>• Echocanc (100B), with allowed values Off, G.165 and G.168;</li><li>• ECHOCI (9021), with allowed values Off, incoming echo canceler on, outgoing echo canceler on, and incoming and outgoing echo canceler on.</li></ul> <p>The codepoints in Annex C/H.248 are for use with binary encoding only, while packages define properties for use with both text and binary encodings. In addition it is expected that SG 11 will complete work on their SPNE package, allowing more advanced control of echo cancellers than the basic control offered by the TDM circuit package of Annex E.13.1.</p>
---------------------	--

	Therefore it is proposed that the codepoints of Annex C dealing with echo cancellation be deprecated, and that the entries in the tables in C.1 and C.9 be listed as Reserved.
--	--

*[Begin Correction]*

---

### C.1 General Media Attributes

	...			<del>Echo Canceller: Off(0), G.165(1), G.168(2)</del> <u>Not Used. See H.248 E.13 for an example of possible Echo Control properties.</u>
--	-----	--	--	--

*[End Correction]*

*[Begin Correction]*

---

### C.9 Bearer Capabilities

	...			<del>Echo Control Information  echo canceler off (0), incoming echo canceler on (1), outgoing echo canceler on (2), incoming and outgoing echo canceler on (3)</del> <u>Not Used. See H.248 E.13 for an example of possible Echo Control properties.</u>
--	-----	--	--	---

*[End Correction]*

## 6.10 Topology Triples in ABNF

<b>Description:</b>	In the ABNF (Annex B), the term TopologyDescriptor allows the specification of only one triple. The ASN.1 permits a sequence of such triples.
---------------------	---

*[Begin Correction]*

---

### B.2 ABNF Specification

```

topologyDescriptor = TopologyToken LBRKT topologyTriple
                    *(COMMA topologyTriple) RBRKT

topologyTriple = terminationA COMMA terminationB COMMA topologyDirection
RBRKT

```

---

*[End Correction]*

## 6.11 Local Control for Annex C

<b>Description:</b>	Currently the introduction of Annex C specifies that native tags are applicable to Local and Remote descriptors. This introduction should also say that native tags are applicable to the Local Control descriptor as the ASN1 encoding makes use of native tags in the Local Control descriptor.
---------------------	---

---

*[Begin Correction]*

# ANNEX C TAGS FOR MEDIA STREAM PROPERTIES (NORMATIVE)

---

Parameters for Local descriptors, and Remote and Local Control descriptors are specified as tag-value pairs if binary encoding is used for the protocol. This annex contains the property names (PropertyID), the tags (Property Tag), type of the property (Type) and the values (Value). Values presented in the Value field when the field contains references shall be regarded as "information". The reference contains the normative values. If a value field does not contain a reference then the values in that field can be considered as "normative".

---

*[End Correction]*

## 6.12 Echo Cancellation Default

<b>Description:</b>	As the Echo Cancellation properties in Annex C have been deprecated in 6.8 of this implementors' guide the default of the Echo Cancellation property should be provisioned to allow for a wider change of applications.
---------------------	---

---

*[Begin Correction]*

### E.13.1 Properties

Echo Cancellation

PropertyID: ec (0x0008)

~~By default, the telephony gateways always perform echo cancellation. However, it is necessary, for some calls, to turn off these operations.~~

Type: boolean

Possible Values:

"on" (when the echo cancellation is requested) and

"off" (when it is turned off.)

The default is "~~on~~provisioned".

Defined In: LocalControlDescriptor

Characteristics: read/write

---

*[End Correction]*

#### 6.14 Error in text on interim AH header

<b>Description:</b>	The UDP destination port should be encoded as 20 hex digits, representing 10 bytes (4 source, 4 dest, 2 port).
---------------------	--

---

*[Begin Correction]*

#### 10.2 Interim AH Scheme

...

As an interim solution, an optional AH header is defined within the H.248 protocol header. The header fields are exactly those of the SPI, SEQUENCE NUMBER and DATA fields as defined in [RFC2402]. The semantics of the header fields are the same as the "transport mode" of [RFC2402], except for the calculation of the Integrity Check value (ICV). In IPsec, the ICV is calculated over the entire IP packet including the IP header. This prevents spoofing of the IP addresses. To retain the same functionality, the ICV calculation should be performed across the entire transaction prepended by a synthesized IP header consisting of a 32 bit source IP address, a 32 bit destination address and a 16 bit UDP destination port encoded as ~~40~~20 hex digits. When the interim AH mechanism is employed when TCP is the transport Layer, the UDP Port above becomes the TCP port, and all other operations are the same.

---

*[End Correction]*

#### 6.15 Termination Subtract from NULL context

<b>Description:</b>	A subtraction of a termination from a NULL context is not allowed however this is not clear in the recommendation. This should be stated.
---------------------	---

---

*[Begin Correction]*

#### 7.2.3 Subtract

...

ALL may be used as the ContextID as well as the TerminationId in a Subtract, which would have the effect of deleting all contexts, deleting all ephemeral terminations, and returning all physical terminations to Null context. Subtract of termination from the NULL context is not allowed.

---

*[End Correction]*

### 6.16 Missing M= Line in Annex SDP

<b>Description:</b>	Section C.11 SDP Equivalent lists various SDP encoding lines. However the Media Line (m=) is missing from this table. The Media line should occur in this table.
---------------------	--

---

*[Begin Addition]*

#### C.11 SDP Equivalent

...

<u>SDP M</u>	<u>B00F</u>	<u>STRING</u>	<u>Media name and transport address</u> <u>Reference: IETF RFC2327</u>
--------------	-------------	---------------	---

---

*[End Correction]*

### 6.17 Missing Optional on Keepactive Flag

<b>Description:</b>	In section 7.1.9 EventsDescriptor is states " Each event in the descriptor contains the Event name, an optional streamID, an optional KeepActive flag, and optional parameters." Clearly the KeepActive flag is meant to be optional however in the ASN.1 this flag is mandatory. The ASN.1 should be updated indicating OPTIONAL.
---------------------	--

---

*[Begin Correction]*

#### A.2 ASN.1 syntax specification

...

```
RequestedActions ::= SEQUENCE
{
    keepActive          BOOLEAN OPTIONAL,
    eventDM             EventDM OPTIONAL,
    secondEvent        SecondEventsDescriptor OPTIONAL,
    signalsDescriptor  SignalsDescriptor OPTIONAL,
    ...
}
```

...

```
SecondRequestedActions ::= SEQUENCE
{
    keepActive          BOOLEAN OPTIONAL,
    eventDM             EventDM OPTIONAL,
    signalsDescriptor  SignalsDescriptor OPTIONAL,
```

|                    }                    ...

---

*[End Correction]*

### 6.18 Syntax Problem in Appendix A

<b>Description:</b>	According to the definition of digitMapRange : digitMapRange = ("x" / LWSP "[" LWSP digitLetter LWSP "]" LWSP) "x" must be followed by "[" .
---------------------	--

---

*[Begin Correction]*

#### B.2 ABNF specification

| digitMapRange                    = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

---

*[End Correction]*

### 6.19 Retaining Descriptors on MOVE

<b>Description:</b>	When a MOVE command is issued on a termination the descriptors currently residing on that termination are retained. This is current ambiguous in the recommendation text.
---------------------	---

---

*[Begin Correction]*

#### Section 7.2.4 Move

The remaining descriptors are processed as in the Modify Command. The Move command does not affect the properties of the Termination on which it operates, except those properties explicitly modified by descriptors included in the Move command. The AuditDescriptor with the Statistics option, for example, would return statistics on the Termination just prior to the Move. Possible descriptors returned from Move are the same as for Add.

---

*[End Correction]*

### 6.20 Clarification on extending packages

<b>Description:</b>	An extended package shall not redefine or overload an identifier defined in the original package. For example: if package "aa" has a signal "ab", then if package "bb" extends aa it cannot define signal "ab". This is also valid for not redefining an id in "earlier" packages, when multiple levels of extension occur.
---------------------	---

Several packages in H.248 Annex E have made this error. Corrections are below.
--

---

*[Begin Correction]*

### Section 12.1.1 Package

A package may extend an existing package. The version of the original package must be specified. When a package extends another package it shall only add additional Properties, Events, Signals, Statistics and new possible values for an existing parameter described in the original package. An extended package shall not redefine or overload an identifiername defined in the original package and packages it may have extended (multiple levels of extension).

Hence, if package B version 1 extends package A version 1, version 2 of B will not be able to extend the A version 2 if A version 2 defines a name already in B version 1.

---

*[End Correction]*

---

*[Begin Correction]*

### Section E.6.2 Events

...

#### DigitMap Completion Event

EventID: ce, 0x0004~~1~~

Generated when a digit map completes as described in section 7.1.14.

---

*[End Correction]*

---

*[Begin Correction]*

### Section E.7.3

Signal Name	Signal ID/tone id
Dial Tone	dt (0x0030)
Ringing Tone	rt (0x0031)
Busy Tone	bt (0x0032)
Congestion Tone	ct (0x0033)

Special Information Tone	sit(0x0034)
Warning Tone	wt (0x0035)
Payphone Recognition Tone	prt (0x0036)
Call Waiting Tone	cw (0x0037)
Caller Waiting Tone	cr (0x0038)

---

*[End Correction]*

### 6.21 Context = ALL in Transaction Reply

<b>Description:</b>	The ASN1 and ABNF allows for the return of Context ID = ALL. This is used in the Wildcard response case. However the text in 8.2.2 Transaction Reply states that only a Specific or NULL is valid for the Context ID. This should be updated to allow ALL.
---------------------	--

---

*[Begin Correction]*

### 8.2.2 TransactionReply

...

The ContextID parameter must specify a value to pertain to all Responses for the action. The ContextID may be specific, all or null.

...

---

*[End Correction]*

### 6.22 Number of Events on a Termination

<b>Description:</b>	The text in H.248 Sect 9.1 says that "On a given Termination, there should normally be at most one outstanding Notify command at any time." This is only a consideration when using UDP. A termination typically can realise multiple events on a terminations.
---------------------	---

---

*[Begin Correction]*

### 9.1 Ordering of Commands

...



3. On a given Termination, there should normally be at most one outstanding Notify command at any time.

...

---

*[End Correction]*

## 6.23 ABNF Token for Signals and Events overlap with packages

<b>Description:</b>	<p>In the ABNF it is possible to define new tokens for Signal and Event information elements. Ie.</p> <pre>sigParameter = sigStream / sigSignalType / sigDuration               / sigOther / notifyCompletion / KeepActiveToken</pre> <p>The package identity can be contained in sigOther. The problem lies in the fact that in the future that the introduction of a new Token for a signal Parameter may cause overlap with an existing package identity. This would lead to an ambiguous interpretation. This problem relates to both signals and events.</p> <p>The solution below limits the any new sigParameter or eventParameter Tokens to start with a certain prefix. An update is also made to the package definition rules saying that packages identities cannot start with this prefix.</p>
---------------------	--

---

*[Begin Correction]*

## B.2 ABNF specification

```
RestChar          = ";" / "[" / "]" / ... / "{" / "}" / ":" / "," / "#" /  
                  "<" / ">" / "="
```

; New Tokens added to sigParameter must take the format of SPA\*  
; \* may be of any form ie. SPAM  
; New Tokens added to eventParameter must take the form of EPA\*  
; \* may be of any form ie. EPAD

```
AddToken          = ("Add" / "A")
```

---

*[End Correction]*

---

*[Begin Correction]*

## 12.2 Guidelines to defining Properties, Statistics and Parameters to Events and Signals

Parameter Name: only descriptive

ParameterID: Is an identifier. The textual ParameterID of parameters to Events and Signals shall not start with "EPA" and "SPA", respectively.-

---

*[End Correction]*

#### 6.24 Error Code for Number of Terminations in a Context exceeded

<b>Description:</b>	H.248 provides a means of setting the maximum number of terminations in a context. However no mechanism is provided to allow an error when the maximum number of terminations is requested to be exceeded.
---------------------	--

---

*[Begin Correction]*

---

#### 7.3 Command Error Codes

...

433 - TerminationID is already in a Context

434 - Max number of Terminations in a Context exceeded.

440 - Unsupported or unknown Package

---

*[End Correction]*

---

#### 6.25 Optional Statistics Parameter Value

<b>Description:</b>	<p>In section 7.2.6 for Audit capability its stated that "StatisticsDescriptor returns the names of the statistics being kept on the termination."</p> <p>But the ABNF grammar states the descriptor as "statisticsDescriptor = StatsToken LBRKT statisticsParameter *(COMMA statisticsParameter ) RBRKT</p> <p>;at-most-once per item</p> <p>statisticsParameter = pkgdName EQUAL VALUE"</p> <p>which doesn't allow specification of stats parameter always with value. If only names needs to be sent from MG, the value field needs to be made option.</p>
---------------------	---

---

*[Begin Correction]*

---

#### Section B.2 – ABNF Specification

...

statisticsParameterr = pkgd Name [EQUAL VALUE]

---

*[End Correction]*

---

## 6.26 Event Permanency

<b>Description:</b>	The current Event Descriptor text is unclear on whether or not a event continues to trigger notifications after the first event is detected. The intention is that the event shall do this.
---------------------	---

*[Begin Correction]*

---

### Section 7.1.9 Event Descriptor

...

When an event is processed against the contents of an active Events descriptor and found to be present in that descriptor ("recognized"), the default action of the MG is to send a Notify command to the MG. Notification may be deferred if the event is absorbed into the current dial string of an active digit map (see section 7.1.14). Any other action is for further study. Moreover, event recognition may cause currently active signals to stop, or may cause the current Events and/or Signals descriptor to be replaced, as described at the end of this section. Unless the events descriptor is replaced by another events descriptor, it remains active after an event has been recognized.

...

*[End Correction]*

---

## 6.27 Wildcard Response Alignment between ASN1 and ABNF

<b>Description:</b>	Chapter 6.2.2 allows wildcard response on all commands. It mentions that it is useful for audit. The ASN1 allows you to request a wildcard response on all commands [w-].  The ABNF only has it at:  auditRequest = ["W-"] (AuditValueToken / AuditCapToken ) EQUAL TerminationID LBRKT auditDescriptor RBRKT
---------------------	---

*[Begin Correction]*

---

### Section B.2 ABNF Specification

commandRequestList= ["O-"] ["W-"] commandRequest \*  
(COMMA ["O-"] ["W-"] commandRequest)

...

subtractRequest = ~~["W-"]~~ SubtractToken EQUAL TerminationID  
[ LBRKT auditDescriptor RBRKT]

...

```
auditRequest = [{"W"}(AuditValueToken / AuditCapToken ) EQUAL  
TerminationID LBRKT auditDescriptor RBRKT  
...
```

---

[End Correction]

## 6.28 MTP addressing for non ITU variants

<b>Description:</b>	The MTP MID needed to be updated to allow for American and Japanese variants as it only currently allows ITU defined MTP addresses to be used.
---------------------	--

---

[Begin Correction]

### Section A.2 ASN.1 Syntax

```
MID ::= CHOICE  
{  
    ip4Address          IP4Address,  
    ip6Address          IP6Address,  
    domainName         DomainName,  
    deviceName         PathName,  
    mtpAddress         OCTET STRING(SIZE(2..4)),  
    -- Addressing structure of mtpAddress:  
    --      15          0  
    --      | PC      | NI |  
    --      24 - 14 bits  2 bits  
    -- Note: 14 bits is defined for international use.  
    -- Two national options exist where the point code is 16 or 24 bits.  
    -- To octet align the mtpAddress the MSBs shall be encoded as 0s.  
    ...  
}
```

### Section B.2 ABNF Syntax

```
...  
; Addressing structure of mtpAddress:  
; 25 - 15          0  
; | PC      | NI |  
; 24 - 14 bits  2 bits  
; Note: 14 bits is defined for international use.  
; Two national options exist where the point code is 16 or 24 bits.  
; To octet align the mtpAddress the MSBs shall be encoded as 0s.  
; An octet shall be represented by 2 hex digits.  
; An mtp address is two octets long  
mtpAddress = MTPToken LBRKT 4*8 (HEXDIG) octetString RBRKT
```

---

[End Correction]

## 6.29 Audit Descriptor and Subtract and Statistics

<b>Description:</b>	The protocol document mentions: 7.1.1 Specifying Parameters
---------------------	--

	<p>...</p> <p>"A missing Audit descriptor is equivalent to an empty Audit Descriptor." and also</p> <p>7.1.15 Statistics Descriptor</p> <p>...</p> <p>"By default, statistics are reported when the Termination is Subtracted from the Context. This behavior can be overridden by including an empty AuditDescriptor in the Subtract command."</p> <p>According to this text, if Subtract command does not have an AuditDescriptor it would mean that there is an empty audit descriptor and no statistics would be reported. And, if the MGC needs termination statistics, it must send AuditDescriptor with Statistics token in the Subtract command. But this seems to change the definition of "By default".</p>
--	---

---

*[Begin Correction]*

### Section 7.1.1 Specifying Parameters

...

If a required descriptor other than the Audit descriptor is unspecified (i.e., entirely absent) from a command, the previous values set in that descriptor for that termination, if any, are retained. In commands other than Subtract, a missing Audit descriptor is equivalent to an empty Audit Descriptor. The behavior of the MG with respect to unspecified parameters within a descriptor varies with the descriptor concerned, as indicated in succeeding sections. Whenever a parameter is underspecified or overspecified, the descriptor containing the value chosen by the responder is included as output from the command.

---

*[End Correction]*

### 6.30 Signal Lists

<b>Description:</b>	<p>There are several inconsistency in the way the signal has been documented they are:</p> <p>Section 7.1.11 states that Signal Lists have type. This is incorrect.</p> <p>Section E.1.2 Doesn't not allow for the specification of which list instance of a signal contained in several lists should generate a notify completion.</p>
---------------------	---

---

*[Begin Correction]*

### Section 7.1.11 Signal Descriptor

A sequential signal list consists of a signal list identifier and, a sequence of signals to be played

~~sequentially, and a signal type. Only the trailing element of the sequence of signals in a sequential signal list may be an on/off signal. If the trailing element of the sequence is an on/off signal, the signal type of the sequential signal list shall be on/off as well. If the sequence of signals in a sequential signal list contains signals of type timeout and the trailing element is not of type on/off, the type of the sequential signal list SHALL be set to timeout. The duration of a sequential signal list with type timeout is the sum of the durations of the signals it contains. If the sequence of signals in a sequential signal list contains only signals of type brief, the type of the sequential signal list SHALL be set to brief. A signal list is treated as a single signal of the specified type when played out.~~

---

*[End Correction]*

---

*[Begin Correction]*

---

## Section E.1.2 Events

...

### Termination Method

ParameterID: Meth (0x0002)

Indicates the means by which the signal terminated.

Type: enumeration

Possible values:

"TO" (0x0001) Duration expired

"EV" (0x0002) Interrupted by event

"SD" (0x0003) Halted by new Signals Descriptor

"NC" (0x0004) Not completed, other cause

### Signal List ID

ParameterID: SLID (0x0003)

Indicates to which signal list a signal belongs. The SignalList ID is only returned in cases where the signal resides in a signal list.

Type: integer

Possible values: Any integer

---

*[End Correction]*

---

## 6.31 Topology

<b>Description:</b>	Topology specifications are cumulative over the life of the context. This is ambiguous in the text.
---------------------	---

---

*[Begin Correction]*

### Section 7.1.18 Topology Descriptor

...

The CHOOSE wildcard in a topology descriptor matches the TerminationID that the MG assigns in the first Add command that uses a CHOOSE wildcard in the same action. An existing Termination that matches *T1* or *T2* in the Context to which a Termination is added, is connected to the newly added Termination as specified by the topology descriptor. ~~The default association when a termination is not mentioned in the Topology descriptor is bothway (if T3 is added to a context with T1 and T2 with topology (T3,T1,one-way) it will be connected bothway to T2).~~ If a termination is not mentioned within a topology descriptor, any topology associated with it remains unchanged. If, however, a new termination is added into a context its association with the other terminations within the context defaults to bothway, unless a topology descriptor is given to change this (eg. if T3 is added to a context with T1 and T2 with topology (T3,T1,one-way) it will be connected bothway to T2).

---

*[End Correction]*

## 6.32 Value optionality in Packages

<b>Description:</b>	When supporting packages you must support all properties, signals, events and statistics. It is unclear in the specification on whether or not you must support all values of properties and parameter. The intention is that a subset of values may be supported.
---------------------	--

---

*[Begin Correction]*

### 6.2.3 Packages

...

Properties, Events, Signals and Statistics defined in Packages, as well as parameters to them, are referenced by identifiers (Ids). Identifiers are scoped. For each package, PropertyIds, EventIds, SignalIds, StatisticsIds and ParameterIds have unique name spaces and the same identifier may be used in each of them. Two PropertyIds in different packages may also have the same identifier, etc.

To support a particular package the MG must support all Properties, Signals, Events and Statistics defined in a package. It must also support all Signal and Event parameters. The MG may support a subset of the values listed in a package for a particular Property or Parameter.

...

---

*[End Correction]*

### 6.33 RequestID in AuditCapReply

<b>Description:</b>	Section 7.2.6 says "... The EventsDescriptor returns the list of possible events on the Termination together with the list of all possible values for the EventsDescriptor Parameters. ..."  What is the value of requestId sent in the events Descriptor from MG to MGC for a AuditCap reply ? ALL should be returned in this case.
---------------------	--

---

*[Begin Correction]*

### Section A.2 ASN.1 Syntax Specification

```
...  
-- For an AuditCapReply with all events, the RequestID SHALL be ALL.  
-- ALL is represented by 0xffffffff.  
RequestID ::= INTEGER(0..4294967295) -- 32 bit unsigned integer  
...
```

---

*[End Correction]*

---

*[Begin Correction]*

### Section B.2 ABNF Syntax Specification

```
...  
-- For an AuditCapReply with all events, the RequestID should be ALL.  
RequestID = ( UINT32 / "*" )  
...
```

---

*[End Correction]*

### 6.34 Context ID Audit

<b>Description:</b>	H.248 allows you the audit the Context ID of where a termination currently belongs. This is not represented in the table in section 7.2.5 Audit Value. It should be added.
---------------------	--

---

*[Begin Correction]*

### Section 7.2.5 Audit Value

---

All	wildcard	Audit of all matching Terminations and the Context to
-----	----------	---



		which they are associated
All	Root	List of all ContextIds
<u>All</u>	<u>Specific</u>	<u>(Non-null) Context Id in which the Termination currently exists</u>

---

*[End Correction]*

### 6.35 Context Priorities

<b>Description:</b>	It is unclear in the recommendation on what the values of the priorities represent.
---------------------	---

---

*[Begin Clarification]*

#### 6.1.1 Context Attributes and Descriptors

...

- The priority is used for a context in order to provide the MG with information about a certain precedence handling for a context. The MGC can also use the priority to control autonomously the traffic precedence in the MG in a smooth way in certain situations (e.g. restart), when a lot of contexts must be handled simultaneously. Priority 0 is the lowest priority and a priority of 15 is the highest priority.

...

---

*[End Clarification]*

### 6.36 Case sensitivity of ABNF and text encoding

<b>Description:</b>	It is unclear whether or the ABNF and its text encoding is case sensitive. The ABNF is not case sensitive.
---------------------	--

---

*[Begin Clarification]*

## Section B.2 ABNF specification

The protocol syntax is presented in ABNF according to RFC2234.

The ABNF and text encoding is case insensitive. The SDP is case sensitive as per RFC2327.

...

---

*[End Clarification]*

### 6.37 Error and topology descriptors

<b>Description:</b>	Section 7.3/H.248 correctly states that when a MG reports an error to a MGC, it does so in an error descriptor. However, the preceding sections listing (supposedly) all descriptors and detailing their contents do not contain any reference to error descriptors. The command descriptions in Section 7.2 also seems to disallow error descriptors to be returned by MGs in reponse to requests from MGCs. We propose to add clarifying text as shown below.  The topology descriptor was also accidentally omitted from the table in Section 6.2.4; its use is, however, explained in Section 7.1.18
---------------------	--

---

*[Begin Correction]*

### 6.2.4 Termination Properties and Descriptors

...

Statistics	In Subtract and Audit, Report of Statistics kept on a Termination
<u>Topology</u>	<u>Specifies flow directions between Terminations in a Context</u>
<u>Error</u>	<u>Contains an error code and optionally error text; it may occur in command replies and in Notify requests</u>

...

---

*[End Correction]*

---

*[Begin Correction]*

#### 7.1.19 Error descriptor

If a command responder encounters an error when processing a transaction request, it must include an error descriptor in its response. A Notify request may contain an error descriptor as well.

An error descriptor consists of an error code, optionally accompanied by an error text. Section 7.3 contains a list of valid error codes.

---

*[End Correction]*

---

*[Begin Correction]*

### 7.2 Command Application Programming Interface

Following is an Application Programming Interface (API) describing the Commands of the protocol. This API is shown to illustrate the Commands and their parameters and is not intended to specify implementation (e.g. via use of blocking function calls). It describes the

input parameters in parentheses after the command name and the return values in front of the Command. This is only for descriptive purposes; the actual Command syntax and encoding are specified in later subsections. The order of parameters to commands is not fixed. Descriptors may appear as parameters to commands in any order. The descriptors SHALL be processed in the order in which they appear.

An error descriptor is a possible reply to any command, the API does not specifically show this.

All parameters enclosed by square brackets ([ . . . ]) are considered optional.

---

*[End Correction]*

### 6.38 Error in digit map activation (section 7.1.14.6).

<b>Description:</b>	<p>The current text in section 7.1.14.6 specifies that a digitmap is activated by means of an (possibly embedded) events descriptor that includes a digit map completion event, which itself contains a digit map parameter.</p> <p>A digit map completion event, however, cannot contain a digit map parameter. Section E.6.2 also specifies that a digit map parameter has to be present.</p> <p>It is more accurate to say that the events descriptor that requests detection of the digitmap completion event must contain an eventDM parameter.</p>
---------------------	--

---

*[Begin Correction]*

#### 7.1.14.6 DigitMap Activation

A digit map is activated whenever a new event descriptor is applied to the termination or embedded event descriptor is activated, ~~and that event descriptor contains a digit map completion event and the digit map completion event which itself contains an eventDM field in the requested actions field. digit map parameter.~~ Each new activation of a digit map begins at step 1 of the above procedure, with a clear current dial string. Any previous contents of the current dial string from an earlier activation are lost.

A digit map completion event that does not contain an eventDM field in its requested actions field, is considered an error. Upon receipt of such an event in an EventsDescriptor, a MG shall respond with an error reponse, including error 457 – Missing parameter in signal or event.

---

*[End Correction]*

---

*[Begin Correction]*

### 7.3 Command Error Codes

...

456 – Parameter or Property appears twice in this descriptor  
457 – Missing parameter in signal or event  
471 – Implied Add for Multiplex failure

...

---

*[End Correction]*

---

*[Begin Correction]*

---

### E.6.2 Events

...

~~EventsDescriptor parameters: None EventsDescriptor parameters: digit map processing is activated only if a digit map parameter is present, specifying a digit map by name or by value. Other parameters such as a KeepActive flag or embedded Events or Signals Descriptors may be present.~~

### E.6.5 Procedures

...

~~None~~ Digit map processing is activated only if an events descriptor is activated that contains a digit map completion event as defined in Section E.6.2 and that digit map completion event contains an eventDM field in the requested actionsparameter as defined in Section 7.1.9. Other parameters such as KeepActive or embedded events of signals descriptors may also be present in the events descriptor and do not affect the activation of digit map processing.

A digit map completion event that does not contain an eventDM field in its requested actions field, is considered an error. Upon receipt of such an event in an EventsDescriptor, a MG shall respond with an error reponse, including error 457 – Missing parameter in signal or event.

---

*[End Correction]*

---

### 6.39 Use of wildcarded TerminationIDs in Add command

<b>Description:</b>	The text in Section 7.2.1/H.248 implies that a CHOOSE wildcard is used only in Add commands that create ephemeral terminations, and cannot be used to allow a MG to choose a particular physical Termination. Moreover, the text implies CHOOSE must be used to create ephemeral Terminations. Neither restriction is valid.
---------------------	--

*[Begin Correction]*

---

### Section 7.2.1 Add

...

The TerminationID specifies the termination to be added to the Context. The Termination is either created, or taken from the null Context. ~~For an existing Termination, the TerminationID would be specific. For a Termination that does not yet exist, the TerminationID is specified as CHOOSE in the command. If a CHOOSE wildcard is used in the TerminationID, the selected~~ The new TerminationID will be returned. Wildcards may be used in an Add, but such usage would be unusual. If the wildcard matches more than one TerminationID, all possible matches are attempted, with results reported for each one. The order of attempts when multiple TerminationIDs match is not specified.

...

*[End Correction]*

---

### 6.40 Meaning of Transaction replies (Section 8/H.248)

<b>Description:</b>	<p>It is unclear when Transaction replies are sent, in particular in the presence of commands that activate signals. Is the reply sent when</p> <ul style="list-style-type: none"><li>• the signals have finished,</li><li>• the signals have been activated, or</li><li>• when the signals have been queued for activation?</li></ul> <p>The intention is that the reply is sent when the signals have been queued for activation, implying that the signals descriptor was syntactically correct and only supported signals were requested.</p>
---------------------	---

*[Begin Correction]*

---

### Section 8.2.2 TransactionReply

The TransactionReply is invoked by the receiver. There is one reply invocation per transaction. A reply contains one or more Actions, each of which must specify its target Context and one or more Responses per Context. The TransactionReply is invoked by the command responder when it has processed the TransactionRequest.

A TransactionRequest has been processed

- when all commands in that TransactionRequest have been processed, or
- when an error is encountered in processing a non-optional command in that TransactionRequest.

A command has been processed when all descriptors in that command have been processed.

A SignalsDescriptor is considered to have been processed when it has been established that the descriptor is syntactically valid, the requested signals are supported and they have been queued to be played out.

An EventsDescriptor or EventBufferDescriptor is considered to have been processed when it has been established that the descriptor is syntactically valid, the requested events can be observed, any embedded signals can be generated, any embedded events can be detected, and the MG has been brought in a state in which the events will be detected.-

...

---

*[End Correction]*

## 6.41 Empty Action requests

<b>Description:</b>	The syntax specification in Annex B/H.248 forbids actions that are completely empty. In particular, an Action has to contain at least a command, a context modification request or a context audit request. The text in Section 8 does not reflect this.
---------------------	--

---

*[Begin Correction]*

## Section 8. Transactions

Commands between the Media Gateway Controller and the Media Gateway are grouped into Transactions, each of which is identified by a TransactionID. Transactions consist of one or more Actions. An Action consists of a non-empty series of Commands, Context property modifications, or Context property audits that are limited to operating within a single Context.

...

---

*[End Correction]*

## 6.42 Auditing list of TerminationIDs

<b>Description:</b>	The protocol contains syntax to allow a MGC to audit which Terminations are in a Context. The relevant clauses in the binary and text encodings are contextAuditResult and contextTerminationAudit.  The intention that the binary and text versions have the same functionality is not met in this case.
---------------------	---

---

*[Begin Correction]*

## Section A.2 ASN.1 Syntax Specification

AuditReply ::= ~~SEQUENCECHOICE~~ ...  
{  
    ~~contextAuditResult~~ TerminationIDList,  
    ~~error~~ ErrorDescriptor,  
    ~~terminationID~~ TerminationID,  
    auditResult AuditResult,

```
    ...  
}  
AuditResult ::= SEQUENCECHOICE  
{  
    contextAuditResult TerminationIDList,  
    terminationID TerminationID,  
    terminationAuditResult TerminationAudit  
}  
TerminationAudit ::= SEQUENCE OF AuditReturnParameter  
...  

```

---

*[End Correction]*

### 6.43 Handoff in case of MGC failure

<b>Description:</b>	Section 11.5/H.248 contains procedures to be followed by MGs in case of MGC failure. The scenario addressed in the second paragraph of this section states that a MG that does not receive an Audit request after having established a control relationship with a backup MGC, acts as if this backup MGC failed. This imposes restrictions on MGC behavior that are unnecessary. For instance, the backup MGC could be a hot standby that does not need to audit the MG when it takes over control. Therefore we propose striking the clause stating this.  Furthermore, the text states that the MG <i>should</i> follow its controlling MGC's Handoff request.
---------------------	---

---

*[Begin Correction]*

### Section 11.5 Failure of an MG

...

In partial failure, or manual maintenance reasons, an MGC may wish to direct its controlled MGs to use a different MGC. To do so, it sends a ServiceChange method to the MG with a "HandOff" method, and its designated replacement in ServiceChangeMgcId. If "Handoff" is supported ~~the MG should shall~~ send a ServiceChange message with a "Handoff" method and a "MGC directed change" reason to the designated MGC. If it fails to get a reply from the designated MGC, ~~or fails to see an Audit command subsequently~~, it the MG shall ~~should~~ behave as if its MGC failed, and start contacting secondary MGCs as specified in the previous paragraph. If the MG is unable to establish a control relationship with any MGC, it shall wait a random amount of time as described in section 9.2 and then start contacting its primary, and if necessary, its secondary MGCs again.

---

*[End Correction]*

## 6.44 Syntax for signal and event parameters in Annex A.2

<b>Description:</b>	Section A.2 contains a clause defining EventParameters as a SEQUENCE consisting of an eventParameterName followed by a value. The type of the value is defined as OCTET STRING. In order to support lists of values mentioned in Section 12.2/H.248, the type of the value field has to be changed to SEQUENCE OF OCTET STRING.
---------------------	---

*[Begin Correction]*

---

### Section A.2 ASN.1 Syntax Specification

```
TimeNotation ::= SEQUENCE
{
    date      IA5String(SIZE(8)), -- yyyyymmdd format
    time      IA5String(SIZE(8))  -- hhmmssss format
}

Value ::= SEQUENCE OF OCTET STRING

END
```

*[End Correction]*

---

## 6.45 Definition of PathName in Annex A.3

<b>Description:</b>	Annex A states that it reproduces the definition of PathName of Annex B. However, the definition presented there is not the same as that provided in Annex B. The appropriate text from Annex B should be copied to Annex A, replacing the current definition given in A.3.
---------------------	---

*[Begin Correction]*

---

### Section A.3 Digit maps and path names

...

A path name is also a string with syntactic restrictions imposed upon it. The ABNF production defining it is copied from Annex B.

```
PathName = NAME *([ "/" ] [ "*" ] [ "@" ] (ALPHA / DIGIT)) [ "*" ]
; Total length of pathNAME must not exceed 64 chars.
pathNAME = [ "*" ] NAME *("/" / "*" / ALPHA / DIGIT / " " / "$" )
           [ "@" pathDomainName ]

; ABNF allows two or more consecutive "." although it is meaningless
; in a path domain name.
pathDomainName = (ALPHA / DIGIT / "*" )
                *63 (ALPHA / DIGIT / "-" / "*" / ".")
```



NAME = ALPHA \*63 (ALPHA / DIGIT / "\_" )

---

*[End Correction]*

#### 6.46 Packaged Name in Modem Descriptor in ABNF

<b>Description:</b>	The ASN.1 Modem Descriptor contains a sequence of Modem parameters of format Packaged Name. The ABNF only contains NAME which does not allow for package definition. Package definition should be allowed.
---------------------	--

---

*[Begin Correction]*

#### Section B.2 ABNF Syntax Specification

...

```
modemDescriptor = ModemToken (( EQUAL modemType) /  
    (LSBRKT modemType *(COMMA modemType) RSBKRT))  
    [ LBRKT propertyParmNAME parmValue  
    *(COMMA propertyParmNAME parmValue) RBRKT ]
```

---

*[End Correction]*

#### 6.47 Error descriptor in Notify request

<b>Description:</b>	Recommendation H.248 allows Notify requests to contain error descriptors. The recommendation does not specify under which circumstances error descriptors are to be included in Notify requests. One case where this is useful is in a Notify request that contains the generic error event defined in Annex E.1.2. This is used in the case when a general error event is triggered.
---------------------	---

---

*[Begin Correction]*

#### Section 7.2.7 Notify

...

The RequestID returns the RequestID parameter of the EventsDescriptor that triggered the Notify Command. It is used to correlate the notification with the request that triggered it. The events in the list must have been requested via the triggering EventsDescriptor or embedded events descriptor unless the RequestID is 0 (which is for further study).

The ErrorDescriptor may be sent in the Notify as a result of Error 518 - Event buffer full.

---

*[End Correction]*

## 6.48 Octet strings in text encoding

<b>Description:</b>	<p>Sometimes it is desirable to transfer octet strings between MG and MGC. The definition of octet string in Annex B is not general enough because it is essentially a text string. Not all characters are allowed in text strings. The null character (0x00) is an example of a character that is not allowed in a text string.</p> <p>A solution to this problem is to use a standard way of encoding the octet string into a text string. Prescribing one way to be used in all cases facilitates uniform encoding and decoding routines.</p> <p>Another problem with the current definition of octetString in Annex B/H.248 is the fact that opening and closing braces must be escaped (i.e. preceded by a backslash). This contradicts the provision in section 7.1.8 that SDP session descriptions conformant to RFC 2327 must be accepted.</p>
---------------------	--

*[Begin Correction]*

---

### Section Annex B Text encoding of the protocol (Normative)

...

#### **B.3 Hexadecimal octet coding:**

Hexadecimal octet coding is a means for representing a string of octets as a string of hexadecimal digits, with two digits representing each octet. This octet encoding should be used when encoding octet strings in the text version of the protocol.

For each octet, the 8-bit sequence is encoded as two hexadecimal digits. Bit 0 is the first transmitted; bit 7 is the last.

Bits 7-4 are encoded as the first hexadecimal digit, with Bit 7 as MSB and Bit 4 as LSB. Bits 3-0 are encoded as the second hexadecimal digit, with Bit 3 as MSB and Bit 0 as LSB.

Examples:

<u>Octet bit pattern</u>	<u>Hexadecimal coding</u>
<u>00011011</u>	<u>D8</u>
<u>11100100</u>	<u>27</u>
<u>10000011 10100010 11001000 00001001</u>	<u>C1451390</u>

#### **B.4 Hexadecimal octet sequence:**

A hexadecimal octet sequence is an even number of hexadecimal digits, terminated by a <CR> character.

*[End Correction]*

---

## 6.49 Annex C USI Correction

<b>Description:</b>	H.248 Annex C lists tag 9008 as being for the USI. However the values only represent the User Information Layer 1. The whole USI value should be supported and the layer 1 protocol.
---------------------	--

*[Begin Correction]*

### Section Annex C.9 Bearer Capabilities

...

<u>layer1prot</u> <u>USI</u>	9008	5 BITS	User Information Layer 1 Protocol Reference: ITU Recommendation Q.931 Bits 5 4 3 2 1 00001 – CCITT standardized rate adaption V.110 and X.30. 00010 - Recommendation G.711 u-law 00011 - Recommendation G.711 A-law 00100 – Recommendation G.721 32 kbit/s ADPCM and Recommendation I.460. 00101 - Recommendations H.221 and H.242 00110 – Recommendations H.223 and H.245 00111 – Non-ITU-T standardized rate adaption. 01000 – ITU-T standardized rate adaption V.120. 01001 – CCITT standardized rate adaption X.31 HDLC flag stuffing. All other values are reserved.
------------------------------	------	--------	---

...

<u>USI</u>	<u>9023</u>	<u>OCTET STRING</u>	<u>User Service Information</u> <u>Reference ITU Recommendation Q.763</u> <u>Section 3.57</u>
------------	-------------	---------------------	---

*[End Correction]*

## 7 Implementation Clarifications

### 7.1 Returning a Context ID List

<b>Description:</b>	In section 7.2.5 AuditValue it has the following table: The following illustrates other information that can be obtained with the Audit Command: <b>ContextID    TerminationID    Information Obtained</b> :
---------------------	--

	All	wildcard	Audit of all matching Terminations and the Context to which they are associated.
	All	Root	List of all ContextIds

***[Begin Clarification]***

---

The list of Context Ids should be returned by using multiple Action Repls, each containing a context from the list.

***[End Clarification]***

---

## 7.2 Service Change Address and Ports

<b>Description:</b>	There is some confusion on when to use either ServiceChange Address or ServiceChange MGCID. The text below offers some advice on
---------------------	--

***[Begin Clarification]***

---

- 1) The use of ServiceChangeAddress is not encouraged
- 2) MGCs must be able to cope with ServiceChangeAddress with either a full address or just a port number in the case of TCP transport.

***[End Clarification]***

---

## 7.3 Audit Response with and without wildcard response (w-)

<b>Description:</b>	There is some confusion on what should be sent in the response to an Audit in the cases where you have wildcarded the context or termination and have specified wildcarded response.
---------------------	--

***[Begin Clarification]***

---

Assume the gateway has 4 terminations : t1/1, t1/2, t2/1 and t2/2

Assume terminations t1/\* has implemented packages aaa and bbb and terminations t2/\* has implemented packages ccc and ddd.

Assume context 1 has t1/1 and t2/1 in it, and context 2 has t1/2 and t2/2 in it.

The command:

```
Context=1{Audit=t1/1{Audit{Packages}}}
```

Returns:

```
Context=1{Audit=t1/1{Packages{aaa,bbb}}}}
```

The command:

Context=\*{Audit=t2/\*{Audit{Packages}}}

Returns:

Context=1{Audit=t2/1{Packages{ccc,ddd}}},

Context=2{Audit=t2/2{Packages{ccc,ddd}}}

The command:

Context=\*{W-Audit=t1/\*{Audit{Packages}}}

Returns:

Context=\*{W-Audit=t1/\*{Packages{aaa,bbb}}}

Wildcard response may also be used for other commands such as Subtract.

---

*[End Clarification]*

## 7.4 Package Extension and Referencing

<b>Description:</b>	The current text in H.248 Section 12.1 is ambiguous about the usage of package names when referencing properties, events, signals and statistics in a base package and an extended package.
---------------------	---

---

*[Begin Clarification]*

### Section 12.1

When packages are extended, the properties, events, signals and statistics defined in the base package can be referred to using either the extended package name. For example, if Package A defines event e1, and Package B extends Package A, then B/e1 is an event for a termination implementing Package B. By definition, the MG MUST also implement the base Package, but it optional to publish the base package as an allowed interface. If it does publish A, then A would be reported on the Package Descriptor in AuditValue as well as B, and event A/e1 would be available on a termination. If the MG does not publish A, then only B/e1 would be available. If published through AuditValue, A/e1 and B/e1 are the same event.

For the purpose of improved interoperability and backward compatibility, the an MG MAY publish all Packages supported by its Terminations, including base Packages from which extended Packages are derived. An exception to this is in cases where the base packages are expressly designed to be extended by others, not directly controlled, and may not have any function on their own or be nonsensical on their own, in which case the MG SHOULD NOT publish the base Packages.

---

*[End Clarification]*

## 8 H.248 Recommendation Series Defect Report Form

<b>DATE:</b>	
<b>CONTACT INFORMATION</b>  <b>NAME:</b> <b>COMPANY:</b> <b>ADDRESS:</b>  <b>TEL:</b> <b>FAX:</b> <b>EMAIL:</b>	
<b>AFFECTED RECOMMENDATIONS:</b>	
<b>DESCRIPTION OF PROBLEM:</b>	
<b>SUGGESTIONS FOR RESOLUTION:</b>	

NOTE - Attach additional pages if more space is required than is provided above.

---