



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T H.248 Sub-series Implementors Guide

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(30 May 2003)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS
Infrastructure of audiovisual services – Communication
procedures

Implementors Guide for the H.248 Sub-series of Recommendations (“Media Gateway Control Protocol”)

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.

Summary

This document is a compilation of reported defects identified in the ITU-T H.248 sub-series of Recommendations currently in force. It must be read in conjunction with the Recommendations to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.248 sub-series Recommendations.

This revision contains all updates submitted up to and including those at Study Group 16 meeting in May 2003.

This document was approved by ITU-T Study Group 16 on 30 May 2003 and obsoletes the earlier version of this Implementors Guide approved on 25 October 2002.

Change Log

(All changes that were included in H.248.1 v1 (03/02) are omitted here.)

V10 (Bruges, June 2002)

Changed references to H.248 Amendment 1 to H.248.1 and added new sections for changes common to H.248.1 v1 and v2 and sections exclusively for H.248.1v2, renumbering the existing sections and IG item numbers.

New:

- 6.1 Specify types for rtp/jit and rtp/delay in Annex E.12.4
- 6.2 Define the '#' symbol in INEQUAL in text encoding
- 6.3 Empty Descriptor Syntax
- 6.4 Define the symbol for NULL Context in text encoding
- 6.5 Corrections to Appendix A example statistics
- 6.6 Corrections to Package Guidelines for Statistics in 12.1.5
- 6.7 Specification of the meaning of automatic in E.13 tdm package

V11 (Geneva, October 2002)

Modification:

- 6.7 Added additional changes to gain

New:

- 6.8 Protocol Version Negotiation
- 6.9 Statistics and Move
- 6.10 Incorrect reference
- 6.11 Additional Codepoint for Annex C
- 6.12 Wildcarding Principles
- 11.1 Additional error codes (this is a new section for H.248.8 and renumbers all higher IG sections)

V12 (San Jose, February 2003)

Modification:

- 11.1 Package-defined Error Codes (Section renamed – no other changes)

New:

- 6.13 Wildcarding in the Topology Descriptor
- 6.14 Binary Value for Packetization Time (Annex C)
- 10.3 Announcement Package Editorial Error
- 11.2 Additional Error Code

V13 (Geneva, May 2003)

New:

- 6.15 Modification of Terminations by MGCs
- 6.16 Optional Command in an Action
- 6.17 Ordering of Transactions
- 6.18 Replies to Actions with no Commands

Contact Information

ITU-T Study Group 16 /
Question 3 Rapporteur

Christian Groves
Australia

Tel: +61 3 9301 6116
Fax: none
E-mail: Christian.Groves@ericsson.com.au

Implementors' Guide ITU-T
Recommendation H.248
Editor

Kevin Boyle II
USA

Tel: +1 919 991 2690
Fax: none
E-mail: kboyle@nortelnetworks.com

Table of Contents

1	SCOPE.....	1
2	INTRODUCTION	1
3	DEFECT RESOLUTION PROCEDURE	2
4	REFERENCES	2
5	NOMENCLATURE	2
6	TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.1 VERSION 1 (03/2002).....	3
6.1	SPECIFY TYPES FOR RTP/JIT AND RTP/DELAY IN ANNEX E.12.4.....	3
6.2	DEFINE THE '#' SYMBOL IN INEQUAL IN TEXT ENCODING	3
6.3	EMPTY DESCRIPTOR SYNTAX	4
6.4	DEFINE THE SYMBOL FOR NULL CONTEXT IN TEXT ENCODING	5
6.5	CORRECTIONS TO APPENDIX A EXAMPLE STATISTICS.....	5
6.6	CORRECTIONS TO PACKAGE GUIDELINES FOR STATISTICS IN 12.1.5	6
6.7	SPECIFICATION OF THE MEANING OF AUTOMATIC IN E.13 TDM PACKAGE	7
6.8	PROTOCOL VERSION NEGOTIATION.....	8
6.9	STATISTICS AND MOVE.....	8
6.10	INCORRECT REFERENCE IN v2	10
6.11	ADDITIONAL CODEPOINT FOR ANNEX C.....	10
6.12	WILD CARDING PRINCIPLES	11
6.13	WILDCARDING IN THE TOPOLOGY DESCRIPTOR	13
6.14	BINARY VALUE FOR PACKETIZATION TIME (ANNEX C)	14
6.15	MODIFICATION OF TERMINATIONS BY MGCs	14
6.16	OPTIONAL COMMANDS IN AN ACTION.....	15
6.17	ORDERING OF TRANSACTIONS	15
6.18	REPLIES TO ACTIONS WITH NO COMMANDS	15
7	TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.2 (2000).....	16
7.1	PACKAGE ID OF TEXT TELEPHONE PACKAGE IN H.248.2 SHALL BE 0x0010	16
7.2	VALUE OF NAK.....	16
7.3	CORRECTION IN PARAMETER VALUES IN CALL TYPE DISCRIMINATION PACKAGE IN H.248.2.....	17
7.4	CORRECTION IN PARAMETER VALUES IN CALL TYPE DISCRIMINATION PACKAGE IN H.248.2.....	17
7.5	MISSING KEYWORDS IN H.248.2 CLAUSE 8.1.2 (EX-F.8.1.2).....	19
7.6	DUPLICATED PROPERTYID IN H.248.2 CLAUSE 8.1 (EX-F.8.1)	19
7.7	INCONSISTENCIES IN FAX TRANSPORT PROPERTY IN H.248.2 CLAUSE 9.1 (EX- F.9.1).....	20
7.8	DUPLICATED PROPERTYID IN H.248.2 CLAUSE 10.1 (EX-F.10.1).....	20
8	TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.3 (2000).....	20
8.1	CORRECT BINARY PROPERTYIDS IN SOFT KEY PACKAGE.....	20
8.2	CORRECT BINARY DIGITMAP COMPLETION EVENTID IN KEYPAD PACKAGE.....	21
9	TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.4 (2000).....	22
9.1	SCTP STREAMS.....	22
10	TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.7 (2000).....	23
10.1	SUPERFLUOUS INFORMATION	23
10.2	ANNOUNCEMENT PLAYING CLARIFICATION	23
10.3	ANNOUNCEMENT PACKAGE EDITORIAL ERROR	25
11	TECHNICAL AND EDITORIAL CORRECTIONS TO H.248.8 (2002).....	26
11.1	PACKAGE-DEFINED ERROR CODES	26
11.2	ADDITIONAL ERROR CODE	26

12	TECHNICAL AND EDITORIAL CORRECTIONS TO RFC-3015	27
12.1	TYPOGRAPHICAL ERRORS IN THE ASN.1 IN RFC3015	27
13	IMPLEMENTATION CLARIFICATIONS FOR H.248.1 VERSION 1 (03/2002)	27
14	IMPLEMENTATION CLARIFICATIONS FOR H.248.4 (2000)	28
14.1	MTP3 INTERWORKING	28
ANNEX: H.248 RECOMMENDATION SUB-SERIES DEFECT REPORT FORM		29

Revised Implementers Guide for the H.248 Sub-series of Recommendations

1 Scope

This guide resolves defects in the following categories:

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

In addition, the Implementers' 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 through contributions to the ITU-T.

2 Introduction

In order to give a clearer understanding of the text components and versions of Recommendation H.248, the Recommendation, including its annexes, has been renumbered into a sub-series according to the table below.

Renumbering table for Recommendation H.248

Previous numbering	New numbering	Title
H.248 (Main body and Annexes A to E)	H.248.1	Gateway control protocol Version 1 <i>Note: This version 1 of H.248.1 is considered to be that of H.248 (Main body and Annexes A to E) of 06/2000, updated with changes, clarifications and corrections (but no new functionality) approved 03/2002. Available for sale but to be superseded by H.248.1 Version 2.</i>
H.248, Annex F	H.248.2	Facsimile, text conversation and call discrimination packages
H.248, Annex G	H.248.3	User interface elements and action packages
H.248, Annex H	H.248.4	Transport over SCTP
H.248, Annex I	H.248.5	Transport over ATM
H.248, Annex J	H.248.6	Dynamic tone definition package
H.248, Annex K	H.248.7	Generic announcement package
H.248, Annex L	H.248.8	Error codes and service change reason description
H.248, Annex M.1	H.248.9	Advanced media server packages
H.248, Annex M.2	H.248.10	Media gateway resource congestion handling package
H.248, Annex M.3	H.248.11	Media gateway overload control package
H.248, Annex M.4	H.248.12	H.248 packages for H.323 and H.324 interworking
H.248, Annex M.5	H.248.13	Quality alert ceasing package
H.248, Annex M.6	H.248.14	Inactivity timer package
H.248, Annex N	H.248.15	SDP H.248 package

The H.248 Implementers' Guide is a compilation of reported defects for all versions of the H.248.x sub-series of Recommendations. In this edition of the Guide, reported defects identified as of 05/2003 are given for:

- H.248.1 version 1 (06/2000 plus corrections and editorial modifications of 03/2002)
- H.248.2 (11/2000)
- H.248.3 (11/2000)
- H.248.4 (11/2000)
- H.248.7 (11/2000)
- RFC3015

The Guide must be read in conjunction with the H.248.x sub-series of Recommendations to serve as an additional source of information for implementors. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected H.248.x Recommendations.

3 Defect Resolution Procedure

Upon discovering technical defects with any components of the H.248.x sub-series Recommendation, please provide a written description directly to the editors of the affected Recommendations with a copy to the Q.3/16 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 any interested party. Formal membership in the ITU is not required to participate in this process.

4 References

This document refers to the following H.248.x sub-series Recommendations:

- ITU-T Recommendation H.248.1 Version 1 (03/2002), *Media Gateway Control Protocol*
- ITU-T Recommendation H.248.1 Version 2 (05/2002), *Media Gateway Control Protocol*
- ITU-T Recommendation H.248.2 (2000), *Facsimile, Text Conversation and Call Discrimination packages*
- ITU-T Recommendation H.248.3 (2000), *User interface elements and action packages*
- ITU-T Recommendation H.248.4 (2000), *Transport over Stream Control Transmission Protocol (SCTP)*
- ITU-T Recommendation H.248.7 (2000), *Generic Announcement Package*
- ITU-T Recommendation H.248.8 (2002), *Error Code and ServiceChange Reason Descriptions*

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:

Symbol	Description
<u>[Begin Correction]</u>	Identifies the start of revision marked text based on extractions from the published Recommendations affected by the correction being described.
<u>[End Correction]</u>	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.
--- SPECIAL INSTRUCTIONS --- {instructions}	Indicates a set of special editing instructions to be followed.

6 Technical and Editorial Corrections to H.248.1 Version 1 (03/2002)

6.1 Specify types for rtp/jit and rtp/delay in Annex E.12.4

Description:	In Regard to rtp/jit and rtp/delay: These package elements do not have types (Integer, Double, etc.) in the 6/00 doc, the IG, or the corrigendum.
Reference:	From: Troy Cauble <troy@bell-labs.com> Date: 6/18/2002 7:22 PM Subject: [Megaco] rtp/jit, rtp/delay

E.12.4 Statistics

[Begin Correction]

Jitter

StatisticID: jit (0x0007)

Requests the current value of the interarrival jitter on an RTP stream as defined in IETF RFC 1889. Jitter measures the variation in interarrival time for RTP data packets.

Type: double

Possible Values: any 64 bit integer

Delay

StatisticID:delay (0x0008)

Requests the current value of packet propagation delay expressed in timestamp units. Same as average latency.

Type: double

Possible Values: any 64 bit integer

[End Correction]

6.2 Define the '#' symbol in INEQUAL in text encoding

Description:	INEQUAL = LWSP (">" / "<" / "#") LWSP The symbol "#" is not explained. From the ASN.1, it appears that it means "not equal". I found a comment in the archives describing it as "quantity of" (??) Which is correct?
Reference:	From: Raphael Tryster Raphael@tdsoft.com Date: 6/2/2002 4:21 AM Subject: [Megaco] INEQUAL, parmValue

B.2 ABNF Specification

[Begin Correction]

INEQUAL = LWSP (">" / "<" / "#") LWSP; "#" means 'not equal'

[End Correction]

6.3 Empty Descriptor Syntax

Description:	<p>BTW I'm not sure if "Events could be an empty eventsDescriptor OR an auditItem" problem was ever followed through. If not, than it might be a good time to do it as was suggested before by Troy Cauble to make it right.</p> <p>Troy wrote:</p> <p>This reminds me...</p> <p>Last June I pointed out that the change to which you are referring (IG 6.79) broke the ABNF in a fairly serious way.</p> <p>Here's an example. The EventsToken below could be one of two things.</p> <pre>MEGACO/1 [135.114.123.123]:1234 Reply = 5000 { Context = 1 { Add = t1 { Events ; this could be an empty eventsDescriptor ; OR an auditItem } } }</pre> <p>Because auditItem</p>
Reference:	<p>From: Aleksandr Ryabin <kengr@winphoria.com> Date: 5/30/2002 1:08 PM Subject: RE: [Megaco] Descriptor grammar issue</p>

B.2 ABNF Specification

[Begin Correction]

```
signalsDescriptor = SignalsToken [ LBRKT † signalParm
                                *(COMMA signalParm) † RBRKT ]
```

[End Correction]

[Begin Correction]

```
auditReturnParameter = (mediaDescriptor / modemDescriptor /
                        muxDescriptor / eventsDescriptor /
                        signalsDescriptor / digitMapDescriptor /
                        observedEventsDescriptor /
                        eventBufferDescriptor /
                        statisticsDescriptor / packagesDescriptor /
                        errorDescriptor / auditReturnItem)
```

```
auditReturnItem = (MuxToken / ModemToken / MediaToken /
                  DigitMapToken / StatsToken /
                  ObservedEventsToken / PackagesToken )
```

[End Correction]

[Begin Correction]

```
;at-most-once, and DigitMapToken and PackagesToken are not allowed
;in AuditCapabilities command
```

```
auditItem = ( auditReturnItem / SignalsToken /  
EventBufferToken / EventsToken )
```

```
auditItem = ( MuxToken / ModemToken / MediaToken /  
SignalsToken / EventBufferToken /  
DigitMapToken / StatsToken / EventsToken /  
ObservedEventsToken / PackagesToken )
```

[End Correction]

6.4 Define the symbol for NULL context in text encoding

Description:	In answering a recent question on the list, I was surprised to not be able to find where we define NULL as being encoded as '-' in text encoding. We define "*" and "\$" in B.1. "ROOT" appears in the syntax for TerminationID. Similarly '-' appears in the syntax for ContextID but nothing states that it stands for NULL. It seems obvious "ROOT" stands for ROOT but less so for '-'. Am I missing something or should be add something to B.2 to state this?
---------------------	---

B.2 ABNF Specification

[Begin Correction]

```
;The values 0x0, 0xFFFFFFFFE and 0xFFFFFFFFF are reserved,  
;"-" is used for NULL context.  
ContextID = (UINT32 / "*" / "-" / "$")
```

[End Correction]

6.5 Corrections to Appendix A example statistics

Description:	The statistics returned in step 22 of the example in Appendix A.1.1 omits some of the statistics in the packages implemented on the terminations and indicates the wrong units for nt/dur. While this example is not normative, errors are confusing to readers.
Reference:	Private discussion during Jun 2002 meeting in Bruges.

Appendix A.1.1 step 22

[Begin Correction]

From MG2 to MGC:

```
MEGACO/1 [125.125.125.111]:55555  
Reply = 50009 {  
  Context = 5000 {  
    Subtract = A5555 {  
      Statistics {  
        nt/os=45123, ; Octets Sent  
        nt/or=45123, ; Octets Sent  
        nt/dur=40000 ; in milliseconds  
      }  
    },  
    Subtract = A5556 {
```

```

Statistics {
  rtp/ps=1245, ; packets sent
  nt/os=62345, ; octets sent
  rtp/pr=780, ; packets received
  nt/or=45123, ; octets received
  rtp/pl=10, ; % packets lost
  rtp/jit=27,
  rtp/delay=48 ; average latency
  nt/dur=38000 ; in millisec
}
}
}

```

[End Correction]

6.6 Corrections to Package Guidelines for Statistics in 12.1.5

Description:	The guidelines for defining statistics for packages only suggests indicating the units of the statistic but not its type or range. The packages in Annex E that define statistics uses sections similar to those for package parameters which include type and possible values. It would seem preferable to change the guidelines to recommend this format.
Reference:	Private discussion during Jun 2002 meeting in Bruges.

12.1.5 Statistics

[Begin Correction]

Statistics defined by the package, specifying:

Statistic name: only descriptive.

StatisticID: Is an identifier

StatisticID is used in a StatisticsDescriptor

Description

~~Units: unit of measure, e.g. milliseconds, packets~~

Type: One of:

Boolean

String: UTF-8 string

Octet String: A number of octets. See Annex A and Annex B.3 for encoding

Integer: 4 byte signed integer

Double: 8 byte signed integer

Character: Unicode UTF-8 encoding of a single letter.

Could be more than one octet.

Enumeration: One of a list of possible unique values (See 12.3)

Sub-list: A list of several values from a list. The type of sub-list SHALL also be specified. The type shall be chosen from the types specified in this section (with the exception of sub-list). For example,

Type: sub-list of enumeration. The encoding of sub-lists is specified in Annexes A and B.3.

Boolean

Possible Values:

A package must indicate the unit of measure, e.g. milliseconds, packets, either here or along with the type above, as well as indicating any restriction on the range.

[End Correction]

6.7 Specification of the meaning of automatic in E.13 tdm package

Description:	<p>The meaning of “automatic” in the gain parameter of E.13 tdm package is not well defined. and</p> <p>There have been several issues raised over the last month or so about gain in the TDM package (Annex E.13) as well as in Annex C. I will summarize the issues and propose changes to address them.</p> <p>1. tdm/gain is defined as integer which in 12.1.2 is clearly signed but the description and choice of value for "automatic" seem to have assumed that the value was unsigned. [consensus was that gain should be signed, negative values have useful meaning and that the reserved value for automatic should be changed. Note this last issue is definitely not backward compatible but no objections were raised.]</p> <p>2. tdm/gain does not specify if it applies to outbound signal level, inbound or both. [consensus was that it should be for outbound signal level.]</p> <p>3. Nigel Williams suggested that the value in text encoding be restricted to decimal (non-hex) representation for easier parsing. [there was no discussion on this, but since the comment the in B.2 specifies that either decimal and hexadecimal can be used for positive values of any integer property, I think we should NOT make this a special case. Note that the specification does require decimal for negative values.]</p> <p>4. There is also a Gain in C.1 (100C) for binary encoding. It is not clear what it applies to and it is defined as unsigned integer and 0.65535 (evidently 2 bytes). [Consensus was that this should be deprecated similar to what we did for echo control.]</p>
Reference:	<p>AVD-2191 a liaison from SG15 received at the Jun 2002, Bruges meeting and futher changes from:</p> <p>Subject: [Megaco] Gain in TDM package and in Annex C From: Terry L Anderson tla@lucent.com Date: 8/1/2002 12:24 PM To: Megaco Mailing List <megaco@ietf.org></p>

C.1 General media attributes

[Begin Correction]

Gain	100C	Unsigned integer	Gain in dB: 0..65535 Not Used. See H.248.1 Annex E.13 for an available gain property.
------	------	-----------------------------	--

[End Correction]

E.13.1 Properties

[Begin Correction]

Gain Control

PropertyID: gain (0x000a)

Gain control, or usage of ~~of~~ signal level adaptation and noise level reduction is used to adapt the level of the outbound signal. However, it is necessary, for example for modem calls, to turn off this function. When the value is set to “automatic”, the termination serves as an automatic level control (ALC) with a target level provisioned on the MG and the direction being outward.

Type: integer
 Possible values:

The gain control specifies the gain in decibels (positive or negative), with the maximum positive integer, 214748646 (0x7fffffff), reserved to represent "automatic". parameter may either be specified as "automatic" (0xffffffff), or as an explicit number of decibels of gain (any other integer value). The default value is provisioned in the MG.

Defined in: LocalControlDescriptor

Characteristics: read/write

[End Correction]

6.8 Protocol Version Negotiation

Description:	<p>Section 11.3 on protocol negotiation, describes the behavior of MGC specifying explicitly that it return version in the response for two cases:</p> <ol style="list-style-type: none">1) MGC supports only a lower version than that proposed by MG2) MGC supports a higher version but can support the version proposed by MG. It does not explicitly state behavior for the case: <p>3) MGC supports ONLY the same version proposed by MG but this is not actually stated. MG of course could not know the difference between #2 and #3. I suppose that an MG receiving a response with no version should assume that the version proposed is being accepted but of course this would have worked for #2 as well. The question is, is omission equivalent to returning the equal value.</p> <p>Description should be changed to cover this case and to indicate that return of version is optional if MG's choice is used.</p>
Reference:	<p>Subject: [Megaco] Protocol version negotiation From: Terry L Anderson tla@lucent.com Date: 7/23/2002 10:27 PM To: Megaco Mailing List <megaco@ietf.org></p>

11.3 Negotiation of protocol version

[Begin Correction]

A ServiceChange command from a MG that registers with an MGC shall contain the version number of the protocol supported by the MG in the ServiceChangeVersion parameter. Regardless of the version placed in the ServiceChangeVersion parameter the message containing the command shall be encoded as a version 1 message. Upon receiving such a message, if the MGC supports only a lower version, then the MGC shall send a ServiceChangeReply with the lower version and thereafter all the messages between MG and MGC shall conform to the lower version of the protocol. If the MG is unable to comply and it has established a transport connection to the MGC, it should close that connection. In any event, it should reject all subsequent requests from the MGC with Error 406 – Version Not Supported.

If the MGC only supports higher version(s) than the MG, it shall reject the association with Error 406 Version Not Supported.

If the MGC supports the version indicated by the MG, it shall conform to that version in all subsequent messages. In this case it is optional for the MGC to return a version in the ServiceChangeReply.

If the MGC supports a higher version than the MG but is able to support the lower version proposed by the MG, it shall send a ServiceChangeReply with the lower version and thereafter all the messages between MG and MGC shall conform to the lower version of the protocol. If the MGC is unable to comply, it shall reject the association, with Error 406 – Version Not Supported.

[End Correction]

6.9 Statistics and Move

Description:	<p>Statistics on a termination are defined in two ways:</p> <ol style="list-style-type: none"> 6.2 says: "Statistics are reported...when the termination is taken out of the call it is in." 7.1.15 says: "The Statistics parameter provides information describing the status and usage of a Termination during its existence with a specific Context." <p>I don't think a "call" is a well defined concept in H.248, but perhaps these two statements are consistent if we associate a "call" with a context. The issue is, if a termination is Moved to a different termination, is it in the same call (one party is probably the same but the other may not be)? Or, more specifically are its statistics zero'ed.</p> <p>Subtract clearly zeros them since ephemerals disappear and physicals get default values when returned to NULL context, and we state this as the last sentence in 7.1.15. The issue is for a Move. We never clearly state this for Move, but from the current language defining what statistics mean, I assume that they ARE zero'ed. It would help if that last sentence in 7.1.15 included Move (or if they are NOT cleared, we state that).</p> <p>We report statistics, by default, on a Subtract, but NOT on a Move. So if statistics are zero'ed by the move, then one must remember to add a Statistics Descriptor to Move to get them where this is not needed for a Subtract. It would seem to be consistent we would have reported statistics by default on all commands that zero them rather than only on Subtract.</p>
Reference:	<p>Subject: [Megaco] Statistics and Move From: Terry L Anderson tla@lucent.com Date: 3/7/2002 4:23 PM To: Megaco Mailing List <megaco@ietf.org></p>

6.2 Terminations [H.248.1v1 (03/02)]

[Begin Correction]

Terminations may have signals applied to them (see 7.1.11). Terminations may be programmed to detect Events, the occurrence of which can trigger notification messages to the MGC, or action by the MG. Statistics may be accumulated on a Termination. Statistics are reported to the MGC upon request (by means of the AuditValue command, see 7.2.5) and when the Termination ceases to exist or is returned to the null context due to a Subtract command, ~~is subtracted from a context.~~

[End Correction]

6.2 Terminations [H.248.1v2 (05/02)]

[Begin Correction]

Terminations may have signals applied to them (see 7.1.11). Terminations may be programmed to detect Events, the occurrence of which can trigger notification messages to the MGC, or action by the MG. Statistics may be accumulated on a Termination. Statistics are reported to the MGC upon request (by means of the AuditValue command, see 7.2.5) and when the Termination ceases to exist or is returned to the null context due to a Subtract command, ~~is taken out of the call it is in.~~

[End Correction]

7.1.15 Statistics descriptor

[Begin Correction]

The Statistics Descriptor provides information describing the status and usage of a Termination during its existence ~~(ephemeral) or while it is outside the null context (physical) within a specific Context.~~ There is a set of standard statistics kept for each Termination where appropriate (number of octets sent and received for example). The particular statistical properties that are reported for a given Termination are determined by the Packages realized by the

Termination. By default, statistics are reported when the Termination ceases to exist or is returned to the null context due to a Subtract command~~is Subtracted from the Context~~. This behaviour can be overridden by including an empty AuditDescriptor in the Subtract command. Statistics may also be returned from the AuditValue command, or any Add/Move/Modify command using the Audit descriptor.

Statistics are cumulative; reporting Statistics does not reset them. Statistics are reset when a ceases to exist or is returned to the null context due to a Subtract command~~Termination is Subtracted from a Context~~.

[End Correction]

E.11.4 Statistics

[Begin Correction]

Duration

StatisticsID: dur (0x0001)

Description: provides duration of time the termination has existed or been out of the null context~~been in the Context~~.

Type: double, in milliseconds

[End Correction]

6.10 Incorrect Reference in v2

Description:	Section 12.5 says see clause 13 for IANA considerations. In v2 the IANA considerations are in clause 14.
---------------------	--

12.5 Package Registration [H.248.1v2 (05/02)]

[Begin Correction]

A package can be registered with IANA for interoperability reasons. See clause ~~13~~14 for IANA considerations.

[End Correction]

6.11 Additional Codepoint for Annex C

Description:	H.248.20 needs an additional code point in Annex C and should be available in both v1 and v2 of H.248.1
Reference:	D-280 from Geneva 10/2002

C.12 H.245

[Begin Correction]

PropertyID	Property tag	Type	Value
------------	--------------	------	-------

PropertyID	Property tag	Type	Value
OLC	C001	Octet string	The value of H.245 OpenLogicalChannel structure. Ref.: ITU-T H.245
OLCack	C002	Octet string	The value of H.245 OpenLogicalChannelAck structure. Ref.: ITU-T H.245
OLCcnf	C003	Octet string	The value of H.245 OpenLogicalChannelConfirm structure. Ref.: ITU-T H.245
OLCrej	C004	Octet string	The value of H.245 OpenLogicalChannelReject structure. Ref.: ITU-T H.245
CLC	C005	Octet string	The value of H.245 CloseLogicalChannel structure. Ref.: ITU-T H.245
CLCack	C006	Octet string	The value of H.245 CloseLogicalChannelAck structure. Ref.: ITU-T H.245
<u>LCN</u>	<u>C007</u>	<u>Integer</u>	<u>The value of H.245 Local Channel Number</u> <u>0 - 65535.</u> <u>Ref.: ITU-T H.245</u>

[End Correction]

6.12 Wild Carding Principles

Description:	<p>Over the last several months on the IETF Megaco list the issue of wild carded context command handling has been raised several times. There has been a significant amount of discussion surrounding this topic and it is apparent that the H.248.1 recommendation is unclear with regards to the handling of wildcarded commands.</p> <p>From the discussions on the list the main issues that need to be clarified are:</p> <ul style="list-style-type: none"> • All commands should have the same behavior with respect to wildcarding • Context=ALL does NOT cover the NULL context. • The reply to commands using a wildcarded Context=ALL and partial Termination ID should not contain an error if at least one of the Full TerminationIDs referenced by the partial TerminationID appear in a context. <p>The intent is to clarify the use of wildcard not change the specification, but since the current text has many ambiguities the clarification will not match all current implementation.</p>
Reference:	D-283 to the Geneva 10/2002 meeting and discussions at the meeting.

[Begin Correction]

6.2.2 TerminationIDs

Terminations are referenced by a TerminationID, which is an arbitrary schema chosen by the MG.

TerminationIDs of physical Terminations are provisioned in the Media Gateway. The TerminationIDs may be chosen to have structure. For instance, a TerminationID may consist of trunk group and a trunk within the group.

A wildcarding mechanism using two types of wildcards can be used with TerminationIDs. The two wildcards are ALL and CHOOSE. The former is used to address multiple Terminations at once, while the latter is used to indicate to a

media gateway that it must select a Termination satisfying the partially specified TerminationID. This allows, for instance, that a MGC instructs a MG to choose a circuit within a trunk group.

~~When ALL is used in the TerminationID of a command, the effect is identical to repeating the command with each of the matching TerminationIDs. The use of ALL does not address the ROOT termination. Since each of these commands may generate a response, the size of the entire response may be large. If individual responses are not required, a wildcard response may be requested. In such a case, a single response is generated, which contains the UNION of all of the individual responses which otherwise would have been generated, with duplicate values suppressed. For instance, given a Termination Ta with properties p1=a, p2=b and Termination Tb with properties p2=c, p3=d, a UNION response would consist of a wildcarded TerminationID and the sequence of properties p1=a, p2=b,c and p3=d. Wildcard response may be particularly useful in the Audit commands.~~

~~The encoding of the wildcarding mechanism is detailed in Annexes A and B.~~

[End Correction]

[Begin Correction]

6.6 Wildcarding Principles

This clause specifies the behaviour for wildcarding Context and Termination Identities that shall be applied to all commands. In processing these commands two forms of wildcarding must be considered:

1. Context Wildcarding
2. Termination Wildcarding

When executing a transaction that contains wildcarded contexts and optionally terminations, all commands in the transaction are executed in order for a particular instance of ContextID before moving to a subsequent ContextID instance. In the case that there are multiple commands in a transaction, only when the TerminationID (wildcarded or specific) specified in the first command matches a specific instance of a ContextID are subsequent commands in the transaction executed. If a TerminationID (wildcarded or specific) of the subsequent command/s in that transaction does not match the specific ContextID instance then an error code 431 is returned and processing of subsequent instances of the wildcard ContextID are stopped unless the command that generated the error is marked optional.

The execution of particular wildcard combinations is discussed below.

6.6.1 ContextID specific with TerminationID wildcarded

In the case where the ContextID is specific, w~~When ALL is used in the TerminationID of a command, the effect is identical to repeating the command with each of the matching TerminationIDs. The use of ALL does not address the ROOT termination. Since each of these commands may generate a response, the size of the entire response may be large. Thus if the wildcard matches more than one TerminationID in the context, all possible matches are attempted, with results reported for each one. If none of the Terminations referenced by the wildcarded TerminationID are in the specific context then error code 431 is returned. No errors are returned for individual terminations specified by the wildcarded TerminationID that are not in the specified context.~~

For example: Assume that a gateway has 4 terminations: t1/1, t1/2, t2/1 and t2/2. Assume that Context 1 has t1/1 and t2/1 in it and that Context 2 has t1/2 and t2/2 in it.

The command:

Context=1{Command=t1/{Descriptor/s}}*

Returns:

Context=1{Command=t1/1{Descriptor/s}}

6.6.2 ContextID wildcarded (ALL) with TerminationID specific

In the case where the ContextID is wildcarded(i.e. ContextID = ALL) and the TerminationID is fully specified, the effect is identical to a command specifying the non-NULL context that contains the specified termination. Thus a search must be made to find the context and only one instance of the command is executed. No errors are reported for Contexts that do not contain the specified termination. If the termination is not contained in any (non-NULL) context

then error 431 is returned. Use of this form of action rather than one specifying the ContextID is discouraged but may be useful, for example in correcting conflicting state between MG and MGC.

For example: Taking the above gateway configuration. The command:

*Context=**{Command=t1/1{Descriptor/s}}

Returns:

Context=1{Command=t1/1{Descriptor/s}}

6.6.3 ContextID wildcarded (ALL) with TerminationID wildcarded

In the case where the ContextID is wildcarded (i.e. Context ID = ALL) and the TerminationID is wildcarded, the effect is identical to repeating the command with each of the TerminationIDs matching the wildcard for each non-NULL context that contains one or more of those matching TerminationIDs. Thus if the wildcard matches more than one TerminationID in the specific instance of the wildcarded ContextID, all possible matches are attempted, with results reported for each one. No errors are reported for Contexts that do not contain a termination matching the wildcarded TerminationID. No errors are returned for individual terminations specified in the wildcarded TerminationID that are not in a specific instance of the wildcarded ContextID. If there are no matches to the wildcarded ContextID and TerminationID then error 431 is returned.

For example: Taking the above gateway configuration.

The command:

*Context=**{Command=t1/*{Descriptor/s}}

Returns:

Context=1{Command=t1/1{Descriptor/s}}

Context=2 {Command=t1/2{Descriptor/s}}

In the case that that multiple commands are contained in a wildcarded TerminationID and/or wildcarded ContextID request then if the first command does not match the first ContextID and TerminationID instance then the subsequent command in the request will not be executed for that instance.

6.6.4 Wildcarded Responses

If individual responses are not required, a wildcard response may be requested. In such a case, a single response is generated, which contains the UNION of all of the individual responses which otherwise would have been generated, with duplicate values suppressed. For instance, given a Termination Ta with properties p1=a, p2=b and Termination Tb with properties p2=c, p3=d, a UNION response would consist of a wildcarded TerminationId and the sequence of properties p1=a, p2=b,c and p3=d. Wildcard response may be particularly useful in the Audit commands. If a wildcard UNION response is used in conjunction with a wildcarded Context then a single response is sent with the UNION of all the individual termination/s referenced by the TerminationID. The response would contain Context=all, a wildcarded TerminationId and the sequence of properties.

If an error occurs during the execution of a wildcarded request that specifies a wildcarded response special handling is required to provide useful information about the error(s) while still maintaining a modest sized response. When a wildcarded response is requested all instances (as specified above) of the command shall be executed even if one or more result in errors, but later commands in the transaction will not be executed (unless optional was specified). Multiple command responses shall be returned for the command that encountered the error. The first command response shall be the normal wildcard response containing the UNION of responses for those commands that succeeded. If none of them succeeded the UNION shall be empty. Additional command responses for each transactionID that failed shall be returned with the appropriate Error Descriptor.

For example

The command:

*Context=**{Command=t1/*{Descriptor/s}}

Response to an error:

*Context=**{Command=t1/*{Union response descriptors},

Command=t1/3{Error=errorcode}}

The encoding of the wildcarding mechanism is detailed in Annexes A and B.

6.13 Wildcarding in the Topology Descriptor

Description:	During discussions on the Megaco Mailing list an ambiguity was recognised in the text describing how wildcarding work in the case of a one way topology descriptor. It was thought that a "*" wildcard was not allowed to be used in the topology triple (T1,T2,oneway). This is incorrect as a "*" wildcard may be used in one termination ID but not both.
Reference:	Subject: RE: [Megaco] Question on Section 7.1.18 Megaco V2 Date: Tue, 14 Jan 2003 13:31:54 -0500 From: "Kevin Boyle" <kboyle@nortelnetworks.com>

[Begin Correction]

7.1.18 Topology Descriptor

- ...
- (T1, T2, oneway) means that the Terminations that match T2 receive media from the Terminations matching T1, but not vice versa. In this case use of the ALL wildcard such that there are Terminations that match either both T1 and T2 but not both is ~~not~~ allowed.
- ...

[End Correction]

6.14 Binary Value for Packetization Time (Annex C)

Description:	During discussion on the Megaco Mailing list it was highlighted that there was no equivalent binary codepoint for the SDP a=ptime. It was felt that a new Annex C property should be defined.
Reference:	Subject: [Megaco] Binary equivalent for SDP ptime attribute Date: Mon, 20 Jan 2003 15:16:25 -0600 From: John_Poplett@3com.com

[Begin Correction]

C.1 General media attribute

RTPpayload	100F	Integer	Payload type in RTP Profile for Audio and Video Conferences with Minimal Control Ref.: RFC 1890
<u>Ptime</u>	<u>1010</u>	<u>Integer</u>	<packet Packetization Time <u>This gives the length of time in milliseconds represented by the media in a packet.</u> Ref.: RFC 2327

[End Correction]

6.15 Modification of Terminations by MGCs

Description:	During discussion on the Megaco Mailing list, it was highlighted that the spec prohibited modification of terms in the NULL context. This clearly was not the intent of the passage, and many implementations allow this very function.
Reference:	Subject: [Megaco] Restriction on Modification Of Terminations Date: Tue, 15 Apr 2003 22:00:09 -0400 From: Christian.Groves@ericsson.com.au

[Begin Correction]

6.2.1 Termination dynamics

The protocol can be used to create new Terminations and to modify property values of existing Terminations. These modifications include the possibility of adding or removing events and/or signals. The Termination properties, and events and signals are described in the ensuing subclauses. An MGC can only release/modify Terminations and the resources that the Termination represents which are in the NULL context or which it has have been previously seized via, e.g. the Add command.

[End Correction]

6.16 Optional Commands in an Action

Description:	There is ambiguity in section 8 on whether or not commands are optional in transaction requests. A command is optional if a context property modification or audit is included.
Reference:	D-347 to the 05/03 Geneva meeting and discussions at the meeting

[Begin Correction]

8.2.1 TransactionRequest

The TransactionRequest is invoked by the sender. There is one Transaction per request invocation. A request contains one or more Actions, each of which specifies its target Context ~~and one or more Commands per Context.~~

...

[End Correction]

6.17 Ordering of Transactions

Description:	Transaction processing order when multiple transactions are contained in a message is not guaranteed. This behavior can generate errors in scenarios where there was successful command execution on the MG. For example: the MGC may receive a message with a transaction reply containing the results of a command ADD(ContextID=?, TerminationID=?) and a transaction request containing a notification from the contextID and TerminationID in the transaction reply. If the MGC chooses to execute the second transaction request first it will generate an error 410 "Incorrect identifier" although the transaction reply contained the ContextID and Termination ID. This error can be avoided if transaction replies are processed before transaction requests.
Reference:	D-347 to the 05/03 Geneva meeting and discussions at the meeting

[Begin Correction]

8 Transactions

...

Transactions guarantee ordered Command processing. That is, Commands within a Transaction are executed sequentially. Ordering of Transactions is NOT guaranteed – transactions may be executed in any order, or simultaneously however transaction replies should be executed before transaction requests when both are contained in a message.

...

[End Correction]

6.18 Replies to Actions with no Commands

Description:	The ABNF commandReply construct requires replies to actions without commands to include the contextProperties specified in the action. This is different from the ASN.1, which does not require this and different from commands in that commandReplies are not required to return fully-specified descriptors set by the command. This is because the
---------------------	--

	ABNF puts the contextProperties construct in the commandReply construct, rather in the actionReply construct. When H.248.1 V3 is produced, then consideration should be given to moving the contextProperties construct up to the actionReply level.
Reference:	Discussions at the 05/03 meeting

[Begin Correction]

7.2.10 Generic Command Syntax

...
A complete ABNF of the text encoding of the protocol per IETF RFC 2234 is given in Annex B. SDP is used as the encoding of the Local and Remote descriptors for use with the text encoding as modified in 7.1.8.

Note: The ABNF is different from the ASN.1 in the case of a reply to an action with no commands. The ABNF requires the contextProperties to be included in the reply even when fully specified in the request, while the ASN.1 does not.

[End Correction]

7 Technical and Editorial Corrections to H.248.2 (2000)1

7.1 Package ID of Text Telephone Package in H.248.2 shall be 0x0010

Description:	The numeric ID of the Text Telephone package in Section 7 of H.248.2 shall be changed to 0x0010 to match the IANA registration.
---------------------	---

[Begin Correction]

~~F.7~~ Text Telephone package

PackageID: txp (0x0010~~6~~)

...

[End Correction]

7.2 Value of NAK

Description:	The numeric value of NAK shall be 0x000D, in the V8bistype parameter of the dtone event in the Call Type Discrimination package.
---------------------	--

[Begin Correction]

~~F.8.2.1~~ Discriminating tone detected

EventID: dtone (0x0001)

...

ObservedEventDescriptor parameters:

....
DiscriminatingToneValue
ParameterID: dtvalue (0x0002)

....

V8bistype
ParameterID: v8bist (0x0004)

Type: enumeration

Possible values:

ESi	(0x0001)	V.8bis signal ESi
ESr	(0x0002)	V.8bis signal ESr

¹ Formerly known as H.248 Annex F.

MRe	(0x0003)	V.8bis signal MRe
MRdi	(0x0004)	V.8bis signal MRd from initiator
MRdr	(0x0005)	V.8bis signal MRd from responder
CRe	(0x0006)	V.8bis signal CRe
CRdi	(0x0007)	V.8bis signal CRd from initiator
CRdr	(0x0008)	V.8bis signal CRd from responder
MS	(0x0009)	V.8bis message MS with contents in "dtvalue"
CL	(0x000A)	V.8bis message CL with contents in "dtvalue"
CLR	(0x000B)	V.8bis message CLR with contents in "dtvalue"
ACK	(0x000C)	V.8bis message ACK with contents in "dtvalue"
NAK	(0x000DE)	V.8bis message NAK with contents in "dtvalue"

...

[End Correction]

7.3 Correction in parameter values in Call Type Discrimination package in H.248.2

Description:	Correction of conflicting parameter values for MRdrh, MRdrl and CReh in the V8bsn parameter of the V8bisSignal signal in the Call Type Discrimination package.
---------------------	--

[Begin Correction]

F-8.3.4 V8bisSignal

SignalID: v8bs (0x0004)

Signaltype: BR

Parameters:

V8bisSigname

ParameterID: V8bsn (0x0001)

Type: Enumeration

Possible values:

ESi	(0x0001)	V.8bis signal ESi
ESr	(0x0002)	V.8bis signal ESr
MRe	(0x0003)	V.8bis signal MRe
MRdi	(0x0004)	V.8bis signal MRd from initiator
MRdrh	(0x0005)	V.8bis signal MRd from responder on high power
MRdrl	(0x0005)	V.8bis signal MRd from responder on low power
CReh	(0x0007)	V.8bis signal CRe on high power
CRel	(0x0006)	V.8bis signal CRe on low power
CRdi	(0x0007)	V.8bis signal CRd from initiator
CRdr	(0x0008)	V.8bis signal CRd from responder
MS	(0x0009)	V.8bis message MS with contents in signalvalue
CL	(0x000A)	V.8bis message CL with contents in signalvalue
CLR	(0x000B)	V.8bis message CLR with contents in signalvalue
ACK	(0x000C)	V.8bis message ACK with contents in signalvalue
NAK	(0x000D)	V.8bis message NAK with contents in signalvalue
MRdrh	(0x000E)	V.8bis signal MRd from responder on high power
CReh	(0x000F)	V.8bis signal CRe on high power

Default may be provisioned

...

[End Correction]

7.4 Correction in parameter values in Call Type Discrimination package in H.248.2

Description:	Correction of conflicting parameter values for dtt parameter in dtone event. in the Call Type Discrimination package.
---------------------	---

F-8.2.1 Discriminating tone detected

EventID: dtone (0x0001)

Description:

This event indicates that a signal valid for detection and discrimination of mode was detected. The signal name is given as a parameter. Further logic is needed in some cases to discriminate the call type from this information. The V.8bis related parameters are returned only when V.8bis is supported [5].

Note that some textphones operate with DTMF tones. This package decodes initial DTMF signals according to the specification for text telephones in V.18 [6]. DTMF detection may be indicated also from the "dd" package if that is active.

EventsDescriptor parameters:

none

ObservedEventDescriptor parameters:

DiscriminatingToneType

ParameterID: dtt (0x0001)

Type: Enumeration

Possible values:

For FAX

CNG	(0x0001)	a T.30 fax calling tone
V21flag	(0x0002)	V21 tone and flags for fax answering

For TEXT

XCI	(0x0003)	a V.18 XCI
V18txp1	(0x0004)	a V.18 txp signal in channel V.21(1)
V18txp2	(0x0005)	a V.18 txp signal in channel V.21(2)
BellHi	(0x0006)	a Bell 103 carrier on the high channel
BellLo	(0x0007)	a Bell 103 low channel
Baudot45	(0x0008)	a Baudot45 initial carrier and characters
Baudot50	(0x0009)	a Baudot50 initial carrier and characters
Edt	(0x000A)	an EDT initial tone and characters
DTMF	(0x000B)	DTMF signals

For DATA

Sig	(0x000CB)	Modulation signal from a mode only used for data, i.e. not V.21, V.23 nor Bell 103
-----	-----------	--

Common to TEXT and DATA:

CT	(0x000DC)	a V.25 calling tone
V21hi	(0x000ED)	a V.21 carrier on the higher frequency channel
V21lo	(0x000FE)	a V.21 carrier on the low frequency channel
V23hi	(0x00100F)	a V.23 high carrier
V23lo	(0x00110)	a V.23 low carrier
CI	(0x0012+)	a V.8 CI with contents in "dtvalue"

Common to FAX, TEXT and DATA:

ANS	(0x00132)	V.25 ANS, equivalent to T.30 CED from answering terminal
ANSbar	(0x00143)	V.25 ANS with phase reversals
ANSAM	(0x00154)	V.8 ANSam
ANSAMbar	(0x00165)	V.8 ANSam with phase reversals
CM	(0x00176)	V.8 CM with contents in "dtvalue"
CJ	(0x00187)	V.8 CJ
JM	(0x00198)	V.8 JM with contents in "dtvalue"

ENDOSIG	(0x001A19)	End of reported signal detected reported for continuous or repeated signals
V8BIS	(0x001B20)	V.8bis signal, with signal type in parameter V8bistype and value in "dtvalue"
		...

[End Correction]

7.5 Missing Keywords in H.248.2 Clause 8.1.2 (ex-F.8.1.2)

Description:	[H.248.2 F.] 8.1.2 neglects to specify "Defined in:" or "Characteristics:"
Reference:	Subject: Re: H.248 Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com, MEGACO list <megaco@fore.com>

[Begin Correction]

F.8.1.2 Text Call Types

V18 (0x0008)

Description:

This parameter indicates for what text telephone modes the termination is monitored, used in TEXT mode.

Defined in: Termination State

Characteristics: Read / Write

...

[End Correction]

7.6 Duplicated propertyID in H.248.2 Clause 8.1 (ex-F.8.1)

Description:	[H.248.2 F.] 8.1.3 and [H.248.2 F.] 8.1.6 have the same PropertyID string (v8bsup). [CHG] Yes. The authors can specify an appropriate name.
Reference:	Subject: Re: H.248 Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com, MEGACO list <megaco@fore.com>

[Begin Correction]

F.8.1.6 PhasereversalDetect

PropertyID: ~~phrevdet~~v8bsup (0x0006)
Type: Boolean

...

[End Correction]

7.7 Inconsistencies in Fax Transport property in H.248.2 Clause 9.1 (ex- F.9.1)

Description:	[H.248.2 F.] 9.1.1 and [H.248.2 F.] 9.1.2 have the same PropertyID number (0x01). [H.248.2 F.] 9.1.2 updated. There is also a spurious dot in one of the value names.
Reference:	Subject: Re: H.248 Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com, MEGACO list <megaco@fore.com>

[Begin Correction]

F.9.1.2 Fax Transport

PropertyID: ftrpt (0x000~~4~~)
Type: Enumeration
Possible values:
T30 (0x0001) for T.30 PSTN sessions without ECM
T30ECM (0x0002) for T.30 PSTN sessions with ECM (non-V.34)
T:30V34 (0x0003) for T.30 PSTN sessions with V.34 (half-duplex)

[End Correction]

7.8 Duplicated PropertyID in H.248.2 Clause 10.1 (ex-F.10.1)

Description:	[H.248.2 F.] 10.1.1 and [H.248.2 F.] 10.1.2 have the same PropertyID number (0x01). [H.248.2 F.] 10.1.2 to be updated.
Reference:	Subject: Re: H.248 Annex F typos Date: Wed, 02 May 2001 16:06:27 +1000 From: Christian Groves <Christian.Groves@ericsson.com> To: Troy Cauble <troy@bell-labs.com> CC: gunnar.hellstrom@era.ericsson.se, gparsons@nortelnetworks.com, jraff@brooktrout.com, rspitzer@telogy.com, MEGACO list <megaco@fore.com>

[Begin Correction]

F.10.1.2 IPFaxTransport

PropertyID: ipftrpt (0x0007~~4~~)
Type: Enumeration
...

[End Correction]

8 Technical and Editorial Corrections to H.248.3 (2000)2

8.1 Correct Binary PropertyIDs in Soft Key Package

Description:	In the Soft Key Package (ks), H.248.3, the Property ID of Property - nskeys (Number of softkeys) is given as 1. This package extends the Label Key package (labelkey) whose Property – keylist (Key List) also has the Property ID 1. Is the clash of the property IDs an oversight?
Reference:	Subject: [Megaco] Regarding Soft Key Package

² Formerly known as H.248 Annex G.

	Date: Thu, 26 Jul 2001 16:49:50 -0700 From: "Anand, Rashim" <r_anand@trillium.com> To: "megaco@ietf.org" <megaco@ietf.org>
--	--

[Begin Correction]

Soft Key Package

PackageID: ks, 0x001a

Properties

Number of softkeys

PropertyID: nskeys (0x0002±)
PropertyType: Integer
Characteristics: read only
Defined in: TerminationState
Description: Maximum number of individual soft keys.

Display size

PropertyID: sz (0x0003±)
Type: Integer
Characteristics: read only
Defined in: TerminationState
Description: Maximum number of characters that can be displayed in each softkey.

Supported unicode code pages

PropertyID: cdpgs (0x0004±)
Description: a list of supported unicode pages
Type: list of enumerated type
Defined in: TerminationState.
Characteristics: read only

[End Correction]

8.2 Correct Binary DigitMap Completion EventID in Keypad Package

Description:	The Event ID ce (0x0001) of Keypad Package, which is derived from Key Package clashes with the Event ID kd (0x0001) of Key Package. What should be the Correct ID for the ce Event?
Reference:	From: Anand, Rashim [mailto:r_anand@trillium.com] Sent: August 30, 2001 18:41 To: 'megaco@ietf.org' Subject: [Megaco] Regarding H.248 Annex G

[Begin Correction]

Keypad Package

Events

DigitMap Completion Event

EventID: ce (0x0004~~3~~)

Generated when a digit map completes as described in Megaco/H.248 Protocol section 7.1.14. Form of this event is identical to its definition in DTMF Detection Package (dd), Megaco/H.248 Protocol section E.6.2.

[End Correction]

9 Technical and Editorial Corrections to H.248.4 (2000)3

9.1 SCTP Streams

Description:	In clause [H.248.4 H.] 8, Stream Independence within H.248.4, it reads: "SCTP can provide up to 65536 unidirectional streams ... " this is correct there can be 65536 unique stream numbers (0-65535). Though the number of streams is limited to what is specified in the INIT / INITACK. There, according to the SCTP RFC 2960 variables: Number of Outbound Streams Number of Inbound Streams are represented by a 16 bit variables where the value of 0 (zero streams) is not allowed. Hence the actual number of streams which may ever be requested and accepted is 0xFFFF (65535). Therefore, the 65536 value in chapter H.8 in H.248.4 should be 65535.
Reference:	Editor

[Begin Correction]

H.8 Stream Independence

SCTP can provide up to 6553~~6~~ unidirectional streams in each direction of an MGC-MG association. SCTP transmits messages and processes received messages in one stream independent to the order or status of messages in any other streams. H.248 may avoid head-of-line blocking by transmitting unrelated transactions on different streams. Reliability is still provided. Ordering of messages is available per-stream.

[End Correction]

³ Formerly known as H.248 Annex H.

If the parameter noc is 0, the signal is played specified by the signal duration only or when not included in the signal by the corresponding announcement default.

To provide additional information when an announcement is to be played, the MGC sends a play variable announcement signal to the MG. H.248 doesn't guarantee in sequence processing of transactions. To ensure sequential playing of an announcement, a transaction reply for a command that affects the announcement signal should be received by the Media Gateway Controller before it sends the additional variable announcement data. If the Media Gateway receives a signal with the keep active flag with additional variable announcement data for an already playing announcement it shall continue playing the announcement according to the additional data.

...

If the signal duration is 0, the signal is played specified by the noc parameter only or when not included in the signal by the announcements number of cycle default.

If the parameter noc is 0, the signal is played specified by the signal duration only or when not included in the signal by the corresponding announcement default.

Table 1 shows the possible combinations of Signal type, Duration, Number of Cycles and the resultant effect.

Table 1 / H.248.7 – Announcement Playing Result

<u>Signal Type</u>	<u>Signal Duration</u>	<u>Number of Cycles (iterations)</u>	<u>Result</u>	
Brief	Not included	Not included	<u>Plays message a number of times up to default duration or a default number of times which ever one is shorter. The message may be stopped part way through a signal.</u>	
		0	<u>Plays the message n times up to the default duration.</u>	
		1	<u>Plays message once or for the default duration which ever one is shorter. The message may be stopped part way through a signal.</u>	
		n times	<u>Plays message a number of times up to default duration or n number of times which ever one is shorter. The message may be stopped part way through a signal.</u>	
	0	Not Included	<u>Plays the message n times according to the default Number of Cycles.</u>	
		0	<u>Plays multiple iterations (endless play)</u>	
		n times	<u>Plays the message n times</u>	
	Duration > per-announcement Duration	Not included	Not included	<u>Plays message a number of times for the specified duration or a default number of times which ever one is shorter. The message may be stopped part way through a signal.</u>
			0	<u>Plays the message a number of times up to the specified duration. The message may be stopped part way through a signal.</u>
			1	<u>Plays message once</u>
			n times	<u>Plays the message n times up to the specified duration. The message may be stopped part way through a signal.</u>
	Duration <= per-announcement Duration	Not included, 0, 1 or n times	<u>Plays message for the specified duration, message stops before being fully played.</u>	
	Timeout	Not Included	Not included	<u>Plays message a number of times for the default duration or a default number of times which ever one is shorter.</u>
			0	<u>Plays the message n times up to the default duration.</u>
			1	<u>Plays message once up to the default duration</u>
n times			<u>Plays the message n times up to the default duration.</u>	
0		Not Included	<u>Plays the message n times according to the default Number of Cycles.</u>	
		0	<u>Plays multiple iterations (endless play)</u>	
		n times	<u>Plays the message n times</u>	
Duration > per-announcement Duration		Not included	<u>Plays message a number of times for the specified duration or a default number of times which ever one is shorter. The message may be stopped part way through a signal.</u>	

		0	Plays multiple iterations until the specified duration
		1	Plays message once, as 1 iteration is shorter than duration
		n times	Plays message a number of times for the specified duration or the specified number of times which ever one is shorter. The message may be stopped part way through a signal.
	Duration <= per-announcement Duration	Not included, 0, 1, n times	Plays for the specified duration, message stops before the message fully plays.
On/Off	Ignored	Not included or 0	Plays multiple iterations (endless play)
		once	Plays multiple iterations (endless play)
		n times	Plays multiple iterations (endless play)

Note: This table is only applicable to H.248.7. The behaviour of signal playing is modified by the use of the Number of Cycles parameter.

...

[End Correction]

10.3 Announcement Package Editorial Error

Description:	H.248.7 contains an editorial error whereby the descriptive text for number of cycles has been placed in the section of the package. This should be fixed.
Reference:	AVD-2292

[Begin Correction]

Editor's Note: When this is included in an updated version of H.248.7, the editor should reorder the parameters to be in numerical order.

4.3.2 Variable: Announcement play

...

Parameters:

Announcement name

ParameterID: an (0x0001)
Type: enumeration of announcements

~~Parameter Number of Cycles Values: any~~

~~0 is used to represent an announcement that continuously plays or loops (OO).~~

Default: provisioned per announcement in the MG

Announcement Variant

ParameterID: av (0x0003)
Type: string
Indicates a specific announcement variant for instance voice type or language.

Announcement Direction

ParameterID: di (0x0007)
Type: enum
Values: External ext (0x01),
Internal int (0x02),
Both both (0x03)

Default: ext

Number of cycles

ParameterID: noc (0x0002)
Type: integer
Values: any, 0 is used to represent an announcement that continuously plays or loops (OO).

Default: 1

Number

ParameterID: num (0x0004)

Type: integer
 Values: any

Specific parameters interpretation
 ParameterID: spi (0x0005)
 Type: enumeration
 Values: any

Specific parameters
 ParameterID: sp (0x0006)
 Type: string
 Values: any
 Default is provisioned for every announcement.

[End Correction]

11 Technical and Editorial Corrections to H.248.8 (2002)

11.1 Package-defined Error Codes

Description:	H.248.8 only contains the error codes for core H.248.1. A statement should be added making it clear that packages may add additional error codes and these will not be added to H.248.8. IANA should be consulted for the complete list of error codes. In addition, the Scope of H.248.8 contains an erroneous reference to “Clause L.5.2”.
Reference:	Discussions at Geneva 10/2002

1 Scope

This Recommendation defines the Error Codes that are used in the core H.248.1 protocol. It lists the Error Code number associated with the Error Code name. It then provides a definition of when the Error Code is to be used and if the Error Code is defined in a package. It also provides an indication of what text may be included in the Error Text descriptor to allow further interpretation of the Error Code. Clause 4.2 provides these details. The normal actions for failed transactions and commands apply as specified in H.248.1 Clause 8. Packages may define error codes not documented in this recommendation. The IANA registry should be consulted for the complete list of error codes.

This Recommendation also defines the Service Change Reasons that are used in the core H.248.1 protocol. It lists the Service Change Reason code associated with the Service Change Reason code name. It provides a definition of when the Service Change Reason is to be used. It also provides an indication of what text may be included in the Service Change extension to allow further interpretation of the Service Change Reason. Clause ~~L.5.4.2~~ provides these details.

11.2 Additional Error Code

Description:	H.248.8 contains many protocol related error codes however there is no error code for the occurrence where the MGC receives a notification for an event that it did not request. This should be seen as a unique error situation that should be reported. <i>Note 1: The proposed Error Code has not yet been registered with IANA. The number assigned will be updated upon IANA registration of the error code.</i>
Reference:	AVD-2292

[Begin Correction]

Editor's Note: The existing sections subsequent to 4.2.33 will be renumbered when H.248.8 is reissued with its corrections, in order to maintain numerical ordering in the Recommendation.

4.2.34 Error Code #: 458 Name: Unexpected Event / Request ID

Definition: The command was disregarded because a notification for an Event / Request ID that the MGC did not request was received.

Package: –

Reference: –

Error Text in the error Descriptor: The Request ID is included in the Error text in the error descriptor.

Comment: –

...

[End Correction]

12 Technical and Editorial Corrections to RFC-3015

This section contains technical and editorial correction to RFC-3015 only, that the faults described in this section do not affect the published ITU-T Recommendation H.248.1 (06/2000).

12.1 Typographical Errors in the ASN.1 in RFC3015

Description:	When producing RFC3015 from Recommendation H.248.1 (06/2000), two lines were omitted. It missed out the line defining IP4Address, which should be before IP6Address and there is a missing "... " at the end of the ServiceChangeParm definition.
Reference:	Subject: FW: Typos in RFC 3015 Date: Tue, 9 Jan 2001 14:27:55 -0500 From: "Rosen, Brian" <Brian.Rosen@marconi.com> To: "Tom Taylor (E-mail)" <taylor@nortelnetworks.com>, "Christian Groves (E-mail)" <Christian.Groves@ericsson.com> CC: "sob@harvard.edu" <sob@harvard.edu>

[Begin Correction]

A.2 ASN.1 Syntax Specification

...

IP4Address ::= SEQUENCE

```
{  
  address      OCTET STRING (SIZE(4)),  
  portNumber   INTEGER(0..65535) OPTIONAL  
}
```

IP6Address ::= SEQUENCE

```
{  
  address      OCTET STRING (SIZE(16)),
```

...

ServiceChangeParm ::= SEQUENCE

```
{  
  serviceChangeMethod  ServiceChangeMethod,  
  serviceChangeAddress  ServiceChangeAddress OPTIONAL,  
  serviceChangeVersion  INTEGER(0..99) OPTIONAL,  
  serviceChangeProfile  ServiceChangeProfile OPTIONAL,  
  serviceChangeReason   Value,  
  serviceChangeDelay    INTEGER(0..4294967295) OPTIONAL,  
                        -- 32 bit unsigned integer  
  serviceChangeMgcId    MId OPTIONAL,  
  timeStamp             TimeNotation OPTIONAL,  
  nonStandardData      NonStandardData OPTIONAL,  
  ...  
}
```

[End Correction]

13 Implementation Clarifications for H.248.1 Version 1 (03/2002)

None.

14 Implementation Clarifications for H.248.4 (2000)

14.1 MTP3 Interworking

Description:	When studying some network scenarios for a certain networks, there is a need to evolve the signalling transport from SS7 MTP3B in an ATM environment to the use of SCTP in IP environments. To provide this M3UA on top of SCTP can be used. It is also seen that M3UA supports flexible implementation scenarios. Therefore some addition indicating the use of M3UA on top of SCTP needs to specified in H.248.4.
Reference:	Subject: MTP 3 interworking Date: Thu, 3 May 2001 16:51:14 +0200 From: "Alf Heidermark (UAB)" <Alf.Heidermark@uab.ericsson.se> To: "Megaco (E-mail)" <megaco@fore.com>

[Begin Clarification]

To provide interworking between MTP3B and SCTP and to allow for flexible implementations of gateways and controllers in order to offer efficient use of SCTP associations the M3UA layer may be added on top of SCTP.

[End Clarification]

Annex: Defect Report Form for H.248 Sub-series of Recommendations

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

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