



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.126

(08/95)

TERMINALS FOR TELEMATIC SERVICES

**MULTIPOINT STILL IMAGE AND
ANNOTATION PROTOCOL**

ITU-T Recommendation T.126

(Previously "CCITT Recommendation")

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation T.126 was prepared by ITU-T Study Group 8 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 11th of August 1995.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

© ITU 1996

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	<i>Page</i>
1 Scope.....	1
2 Normative references	1
3 Definitions.....	3
4 Abbreviations	4
5 Overview	4
5.1 SI application enrollment.....	4
5.2 Capabilities and profiles	5
5.3 Workspaces.....	5
5.4 Hard-copy devices	6
5.5 Bitmaps	6
5.6 Annotations.....	7
5.6.1 Drawing and erasing basic shapes	7
5.6.2 Drawing and erasing custom shapes	7
5.6.3 Text	7
5.7 Pointers	7
5.8 Remote events.....	8
5.9 Archives.....	8
5.10 Conducted mode behaviour	8
6 Use of MCS.....	8
6.1 MCS token and channel usage.....	8
6.2 Use of MCS data services.....	8
7 Use of GCC.....	13
7.1 GCC unique handles	13
8 Protocol specification.....	13
8.1 SI protocol parameters.....	13
8.2 SI application enrollment and initialization.....	14
8.2.1 Static mode.....	15
8.2.2 Multicast mode.....	17
8.2.3 Private mode	18
8.2.4 Forming registry keys	19
8.2.5 Establishing dynamic tokens.....	21
8.2.6 SI capabilities.....	21
8.3 Workspaces.....	32
8.3.1 Workspace structure.....	32
8.3.1.1 Plane stacking	32
8.3.1.2 Plane coordinate system.....	33
8.3.1.3 Workspace views	33
8.3.1.4 Drawing color palette.....	33
8.3.1.5 Workspace attributes.....	39
8.3.1.5.1 Workspace synchronization.....	39
8.3.1.5.2 Acceptance of keyboard and pointing device events.....	40
8.3.1.5.3 Preference to preserve the workspace.....	40
8.3.1.5.4 Background color	40
8.3.1.6 Workspace plane parameters	40
8.3.1.6.1 Plane editability	41
8.3.1.6.2 Plane usage	41
8.3.1.6.3 Plane protection	42

	<i>Page</i>
8.3.1.7	Workspace view parameters 42
8.3.1.7.1	View region 42
8.3.1.7.2	View state 43
8.3.1.7.3	Update enable flag 43
8.3.1.7.4	Source display indicator 43
8.3.2	Creating a workspace 44
8.3.3	Deleting a workspace 47
8.3.4	Editing workspace, plane, and view attributes 47
8.3.5	Copying workspace contents 49
8.3.6	Workspace refreshing for late arrivers 49
8.3.7	The effect of changes to the application roster 51
8.3.8	Workspace caching 52
8.4	Bitmaps 52
8.4.1	Creating bitmaps 53
8.4.2	Deleting Bitmaps 60
8.4.3	Editing bitmaps 61
8.4.4	Bitmap color definition 61
8.4.5	Bitmap color component sampling ratios 62
8.4.6	Bitmap formats 62
8.4.6.1	Uncompressed 62
8.4.6.2	Recommendation T.4 (G3) 65
8.4.6.3	Recommendation T.6 (G4) 65
8.4.6.4	Recommendation T.81 (JPEG) 66
8.4.6.5	Recommendation T.82 (JBIG) 68
8.4.6.6	Non-standard bitmap format 70
8.5	Pointers 70
8.6	Drawing 70
8.6.1	Drawing shapes 72
8.6.1.1	Point 72
8.6.1.2	Open polyline 73
8.6.1.3	Closed polyline 73
8.6.1.4	Rectangle 73
8.6.1.5	Ellipse 74
8.6.1.6	Non-standard 74
8.6.2	Drawing element attributes and parameters 75
8.6.2.1	Sample rate 75
8.6.2.2	Rotation 75
8.6.2.3	Bounding rectangle 75
8.6.2.4	Pen color 75
8.6.2.5	Pen thickness 75
8.6.2.6	Pen nib 76
8.6.2.7	Line style 76
8.6.2.8	Fill color 76
8.6.2.9	Highlighting 77
8.6.2.10	View state 77
8.6.2.11	Depth order 77
8.6.3	Creating drawing elements 77
8.6.4	Deleting drawing elements 80
8.6.5	Editing Drawing Elements 80
8.7	Remote events 81
8.7.1	Remote keyboard events 83
8.7.2	Remote pointing device events 84
8.7.3	Remote printing events 85

	<i>Page</i>
8.8 Archives.....	86
8.9 Conducted mode operation.....	88
9 SIPDU definitions.....	94
Annex A – SI profiles.....	127
Annex B – Static channel and token ID assignments.....	128
B.1 Static channel ID assignments.....	128
B.2 Static token ID assignments.....	128
Annex C – Object identifier assignments.....	128
Appendix I – Deriving intermediate palettes for bitplane progressive transmission of palettized images.....	129

SUMMARY

This Recommendation defines a protocol supporting the multipoint exchange of still images, annotations, pointers and remote events. In the case of still images, both hard – and soft – copy exchanges are supported. This protocol conforms to the conference conductorship model defined in Recommendation T.124. Protocol elements for creating and referencing archived images with associated annotations are also specified. This Recommendation uses services provided by Recommendations T.122 (MCS) and T.124 (GCC).

MULTIPOINT STILL IMAGE AND ANNOTATION PROTOCOL

(Geneva, 1995)

1 Scope

This Recommendation defines an application protocol that supports shared whiteboarding and both soft and hard copy still image conferencing with associated annotations. It uses services provided by Recommendations T.122 (MCS) and T.124 (GCC). Basic remote pointing and keyboard event exchanges have also been included such that terminals can implement basic computer application sharing.

The details of communication with the input and output devices and the user interfaces on the host terminal are considered out of the scope of this Recommendation and are left to the discretion of implementor. Therefore this Recommendation makes no assumption that these I/O devices are of any specific architecture.

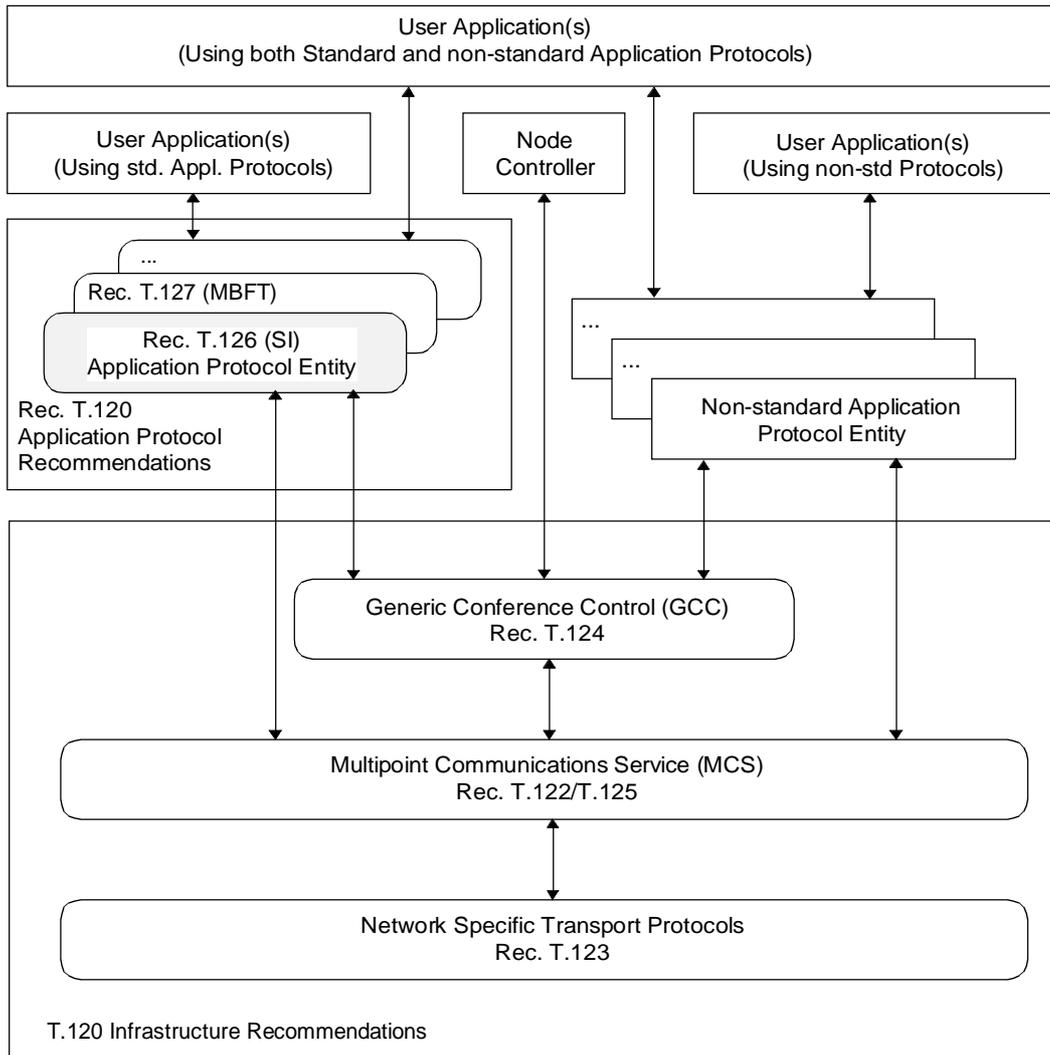
Figure 1-1 presents an overview of the scope of this Recommendation and its relationship to the other elements of the T.120 framework within a single node.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. A list of the currently valid ITU-T Recommendations is regularly published.

- CCITT Recommendation F.710 (1991), *General principles for audiographic conference service.*
- ITU-T Recommendation H.221 (1993), *Frame structure for a 64 to 1920 Kbits/s channel in audiovisual teleservices.*
- ITU-T Recommendation T.4 (1993), *Standardization of Group 3 facsimile apparatus for document transmission.*
- CCITT Recommendation T.6 (1988), *Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus.*
- CCITT Recommendation T.35 (1991), *Procedure for the allocation of CCITT defined codes for non-standard facilities.*
- ITU-T Recommendation T.42 (1994), *Continuous-tone color representation method for facsimile.*
- CCITT Recommendation T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange.*
- CCITT Recommendation T.81 (1992), *Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines.*
- ITU-T Recommendation T.82 (1993), *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression.*
- ITU-T Recommendation T.120¹⁾, *Data protocols for multimedia conferencing.*

¹⁾ Presently at the stage of draft.



T0819330-94/d01

FIGURE 1-1/T.126
Scope of T.126

- ITU-T Recommendation T.122 (1993), *Multipoint communication service for audiographics and audiovisual conferencing service definition.*
- ITU-T Recommendation T.123 (1994), *Protocol stacks for audiographic and audiovisual teleconference applications.*
- ITU-T Recommendation T.124 (1995), *Generic conference control.*
- ITU-T Recommendation T.125 (1994), *Multipoint communication service protocol specification.*
- ITU-T Recommendation X.680 (1994), *Information technology – Abstract Syntax Notation One (ASN.1)– Specification of basic notation.*
- ITU-T Recommendation X.690 (1994), *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER).*

- ITU-T Recommendation X.691 (1995), *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.
- ITU-R BT 601-1:1992, *Encoding Parameters of Digital Television for Studios*, Volume RBT.
- ITU-R BT 709:1990, *Basic Parameter Values for the Studio and for International Programme Exchange*, Volume XI-1.
- ISO/IEC 10646-1: 1993, *Information technology – Universal Multiple-Octet Coded Character Set (USC) – Architecture and basic multilingual plane*.
- CIE 1976 (L*a*b*) space, *CIE publication No. 15.2, Colorimetry, 2nd Ed. (1986)*.

3 Definitions

For the purposes of this Recommendation, the following definitions apply.

- 3.1 4:4:4:** A notation that defines the relative horizontal resolution of a three-color component raster to be equal.
- 3.2 4:2:2:** A notation that defines the relative horizontal resolution of a three-color component raster to have twice the horizontal resolution on the first channel than the other two.
- 3.3 4:2:0:** A three-color component raster having twice the horizontal resolution as well as twice the vertical resolution on the first channel.
- 3.4 annotation:** Real time drawings (freehand drawing, lines, rectangles, ellipses, etc.) and bitmaps (used for text and unsupported graphical elements, for example) shared between conferenced sites.
- 3.5 application conferencing:** The use of SIPDUs to accomplish simple application conferencing among conferenced SICEs. One SICE shall run an application that is controllable by any SICE in the conference.
- 3.6 bitmap:** A rectangular area described by a two dimensional array of pixels. These pixels can be coded using a variety of encoding methods.
- 3.7 control Points:** A set of points defined in terms of the workspace coordinate system that define a drawing shape parametrically.
- 3.8 drawing:** A type of annotation consisting of instructions for creating points, polylines, rectangles, ellipses, or non-standard drawing elements.
- 3.9 handle:** A conference-wide unique number used to identify an addressable item.
- 3.10 image:** Photographic or document oriented information which is transmitted in the form of an image bitmap.
- 3.11 non-standard capability:** The capability is outside the scope of this Recommendation but it has been determined through negotiation that it is recognized among all conference participants.
- 3.12 palette:** A finite set of colors defined by at least three linearly independent color primaries.
- 3.13 palettized:** A term used to describe visible objects (annotation bitmaps, drawing elements) comprised of palettized pixels. The color of a palettized pixel is specified by the color value at the location in a color look-up table referenced by the pixel value.
- 3.14 plane:** A virtual area defined to have the same pixel dimensions as the workspace with which it is associated. A plane provides a canvas for the use of annotation tools such as drawing, erasing, and text, as well as for bitmaps.

- 3.15 pointer plane:** A virtual area defined to have the same dimensions as the workspace with which it is associated. This virtual area in front of all other planes houses all the pointers referencing a given workspace.
- 3.16 pointer:** A bitmap that is moveable over the workspace that is used by its creator as an indicator of position.
- 3.17 standard capability:** The capability is defined within the scope of this Recommendation but is not required for all SICE implementation. Note that all standard capabilities must be negotiated before use.
- 3.18 still image conference entity:** An Application Protocol Entity that interacts with a user application above and with the local MCS (Multipoint Communication Service) and local GCC provider below. Data are exchanged between peer SICEs using SIPDUs (Still Image Protocol Data Unit).
- 3.19 unicode:** Multilingual text string format as defined in ISO/IEC 10646-1.
- 3.20 workspace:** A workspace is an area comprising N independent but coincident planes of the same pixel dimensions. The N planes whose assembly forms the complete display. At a given workspace coordinate, data in any plane hides data present in underlying planes in the stack unless the pixel value at that plane is transparent. If there are no data at a specific pixel location in the middle or front planes, the location is said to be transparent and does not hide the data from underlying planes.

4 Abbreviations

For the purposes of this Recommendation, the following abbreviations are used:

GCC	Generic Conference Control
GCCSAP	Generic Conference Control Service Access Point
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MCS	Multipoint Communication Service
MCSAP	MCS Service Access Point
MCU	Multipoint Control Unit
PDU	Protocol Data Unit
SICE	Still Image Conferencing Entity
SIPDU	Still Image Protocol Data Unit

5 Overview

5.1 SI application enrollment

An SI application enrolls via the application enrollment mechanism specified in Recommendation T.124. An SICE receives an indication from GCC when it may begin the enrollment process. There are two scenarios where this happens. The first is the normal conference start-up process, where the local node controller joins a conference. After it locally establishes itself within the conference, a GCC-Application-Permission-To-Enroll indication is issued to all local applications. The second occurs during an established conference when one node decides to invoke a session of a specific application at multiple nodes.

5.2 Capabilities and profiles

The transactions defined by the SI protocol and the ranges on many of the associated parameters are governed by the capabilities set that is in effect at the time of the exchange. Capability profiles exist for terminals wishing to operate as a whiteboard only, operate as a soft copy image exchange terminal, operate as a soft copy image exchange terminal that can annotate and whiteboard, or operate as a hard copy image exchange device. For forward compatibility reasons, these profiles are simply lists of capabilities that must be advertised, therefore it is possible for a terminal to be capable of one or more of the above functions.

See Annex A for a detailed description of the standard SI profiles. The capabilities exchange mechanism is the method by which additional capabilities that are not assumed by the specific application class are negotiated. The vehicle for this negotiation is the GCC application enrollment facility which has a well-defined protocol for this purpose.

5.3 Workspaces

The workspace data structure and associated operators provides a self-contained, platform-independent method for describing, manipulating, and maintaining related annotation, pointer, and bitmap data. A workspace is composed of N depth ordered planes. Higher numbered planes (and all their associated contents) are assumed to be in front of their lower numbered counterparts within the same workspace. Each plane's contents may include images and/or annotations depending on how the plane is tagged at creation time. Depending on negotiated capabilities, a workspace may also include a virtual pointer plane which resides above all other planes. See Figure 5-1.

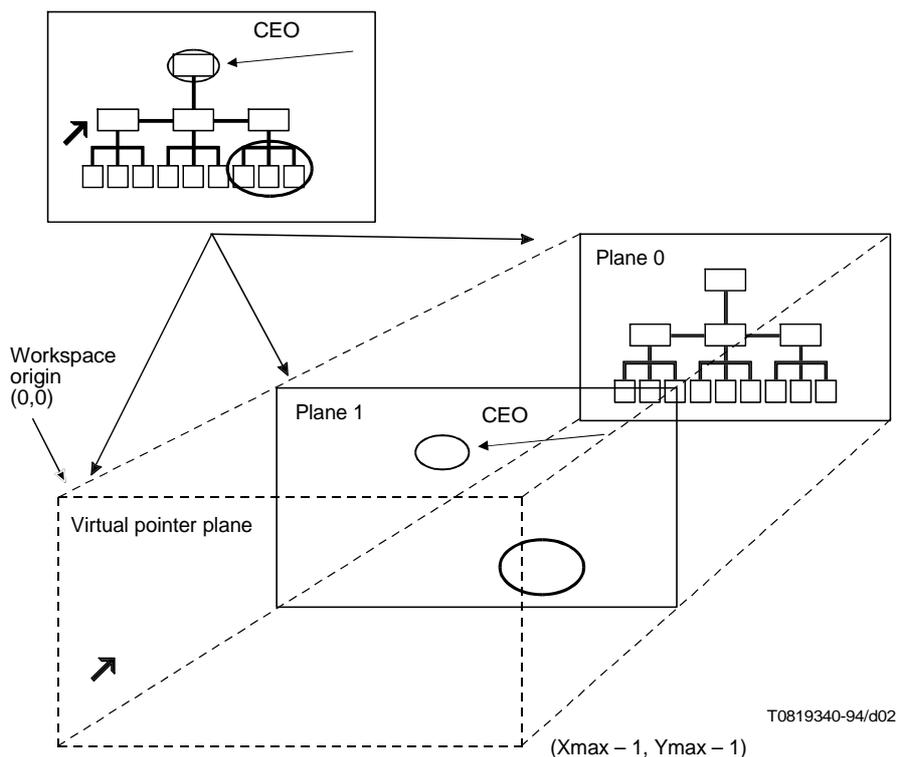


FIGURE 5-1/T.126

Workspace

A created workspace in a conference has the same pixel dimensions at all sites within the conference. Placement of annotations, and images each in their respective planes is indicated in workspace coordinates. All workspace coordinates are specified in units of whole pixels. The pixel aspect ratio is 1:1 but there is no absolute size associated with a pixel. Coordinates referenced in this Recommendation are shown as (X,Y) where the workspace origin (0,0) is defined to be at the upper left hand corner of the workspace. The lower right hand corner of the workspace has the coordinate value (Workspace horizontal dimension – 1, Workspace vertical dimension – 1). Even though workspace pixels are defined to have a square aspect ratio, bitmaps may be exchanged whose native resolutions are non-square. The allowable aspect ratios are dictated by the enforcing profile and capability set.

To make a workspace visible, one or more workspace views may be defined for a workspace. Each view corresponds to a rectangular region within the workspace to be viewed (which, in the simplest case, is the entire workspace). Within an SI session, only one view among all workspaces is designated as the focus view. This view is required to be made visible at all nodes.

A SICE implementation may not have a display subsystem which can accommodate the full dimensions of the negotiated maximum size workspace. In this case, the SICE may use any suitable method such as scroll bars or scaling to accommodate a workspace view of a size larger than the local display resolution. Similarly, a workspace view smaller than the local display resolution may, if desired, be scaled appropriately to fill the display. However, regardless of how an SICE chooses to display a workspace view, it must locate its annotations, pointers, and bitmaps based on the coordinate system defined for the workspace.

A conference can support multiple workspaces given sufficient storage in each SICE. Upon indication by GCC of new arrivals in the conference, all workspaces are deleted upon the creation of the first new one. All SICES shall create a new workspace whenever they detect that there is a transition from one logical boundary to another. The logical boundary condition is locally determined. An example of one might be the progress from one slide or view graph to the next in a presentation application using this Recommendation. Because, for capability set reasons, new conference participants cannot necessarily observe the conference information until the first workspace creation after they have joined, it is imperative that applications create workspaces as often as possible.

5.4 Hard-copy devices

An SICE can advertise an optional capability that expresses the ability to receive bitmap exchanges bound for hard copy devices directly. If at least two nodes in the conference have this capability, they can exchange these types of bitmaps, which are restricted by a separate capability set from those of the soft copy functions that accommodates to facsimile-like devices.

5.5 Bitmaps

The SI protocol supports the exchange of bitmaps. The supported standard formats include:

- Uncompressed;
- Recommendation T.4 (G3);
- Recommendation T.6 (G4);
- Recommendation T.81 (JPEG);
- Recommendation T.82 (JBIG).

Other formats can be negotiated. The governing profile may only allow a subset of the above list in a conference. The SI protocol supports bitmap scaling and cropping on a capability negotiated basis. These functions are useful, for example, when attempting to transmit a precompressed image where decompression and recompression would have to be performed if only a subregion of the bitmap is of interest or the bitmap is at an inappropriate resolution.

All submitted bitmaps can be optionally edited and deleted using SI defined exchanges. These operations are only allowed to workspace planes that will support them and may be disallowed by capabilities negotiation.

5.6 Annotations

5.6.1 Drawing and erasing basic shapes

The SI protocol defines exchanges for basic drawing shapes. These include:

- open and closed polylines (freehand drawing);
- points;
- rectangles;
- ellipses.

Attributes of the SI basic drawing shapes include:

- line color;
- fill color;
- line thickness;
- pen nib shape;
- line style.

Erasing is supported in two ways depending on whether the target workspace plane is of the permanent or editable type. If permanent, erasing is accomplished by drawing over the desired areas with the outline and fill colors set to transparent. Otherwise, erasing is supported by deleting the desired object using the SI drawing delete exchange.

All submitted drawings can be optionally edited and erased using SI defined exchanges. These operations are only allowable to workspace planes that will support them and may be disallowed from the entire conference by capabilities negotiation.

5.6.2 Drawing and erasing custom shapes

Custom shapes that are not supported by the SI protocol or that are disallowed by the specific profile being used or capabilities set negotiated can be supported within the conference by rendering them locally into a bitmap which is then submitted to the conference using the SI bitmap exchange mechanism. Note that bitmaps can be filled with a transparent color in all pixel locations that are not affected by the custom shape being rendered.

Custom shapes (and their associated custom attributes) can be supported within a conference using non-standard shape and attribute fields within the SI drawing PDUs if they are successfully negotiated.

5.6.3 Text

Direct support of text primitives is left for future study. Text can now be exchanged using the SI protocol by rendering the text locally into a bitmap and then transmitting that bitmap to the conference. Alternatively, non-standard text primitives can be negotiated and used within the conference.

5.7 Pointers

Pointers are supported using the bitmap exchange functions. Bitmaps specified as pointers do not belong to any specific workspace plane but instead are managed with respect to the workspace in a virtual plane on top of all other data bearing planes. Pointers are privately held by their creator and are removed if the creator leaves the conference in any fashion. Only the creator of a pointer may move it or delete it.

5.8 Remote events

The SI protocol supports the exchange of keyboard and pointing device events as well as requests for remote workspace printing. These facilities can be optionally supported by terminals. They are not negotiated and should be ignored by any terminal that does not support them. The intended application for remote keyboard and pointing device events is basic application sharing where an applications's display is transmitted to the conference (and updated when changed). Remote sites in the conference could then contribute keystrokes or pointing device events to remotely pilot the application.

5.9 Archives

This Recommendation supports an archiving function that allows the remote retrieval of information from pre-distributed databases. It also specifies a protocol that can be optionally used to create those databases and add to them remotely.

5.10 Conducted mode behaviour

When a conference is in conducted mode, the SICE at the conductor node may grant a set of privileges to perform various actions to one or more nodes in the conference. Without such privileges or global conducted-mode permission from the conducting node, an SICE is restricted from performing these actions.

6 Use of MCS

An SICE uses the following MCS service primitives to attach and detach from a domain, join and leave the SI channel, send and receive SIPDUs, and manage token operations. Table 6-1 describes each of the primitives used by an SICE.

MCS request and response primitives are directed from the SICE to the MCS provider, while indication and confirm primitives are directed from the MCS provider towards the SICE. Additional detail on the MCS primitives described above can be found in Recommendation T.122, Multipoint Communication Service for audiographic and audiovisual conferencing – Service definition.

6.1 MCS token and channel usage

Table 6-2 describes MCS channel and token usage for default (static), multicast, or private SICE sessions. In the case of the default SICE session, static tokens and channels (except for the MCS-USER-ID channels) shall be used as defined in Table 6-2 (symbolic IDs shown). For multicast or private SICE sessions, the Resource IDs shown in the table shall be used for allocating dynamic tokens and channels. The given Resource IDs shall be encoded as two octet T.50 text strings using the characters shown in quotes in Table 6-2.

6.2 Use of MCS data services

Table 6-3 lists the use of the MCS data services MCS-SEND-DATA and MCS-UNIFORM-SEND-DATA for each SIPDU. This table includes the channel over which the data is sent, which of the two MCS primitives shall be used for the case of synchronized and unsynchronized workspaces, and the data priority at which the data is sent. If more than one channel priority is mandated, the SIPDU must be sent on all.

All PDUs specified in this Recommendation are placed in the Data parameter of the MCS-SEND-DATA and MCS-UNIFORM-SEND-DATA primitives. The ASN.1 encoded PDUs are packed into the sequence of octets that form the Data parameter such that the leading bit is placed in the most significant bit of each octet, and filled toward the least significant bit of the octet.

TABLE 6-1/T.126

MCS primitives needed by an SICE

MCS Primitive	Description
MCS-ATTACH-USER	Creates an MCS attachment through an MCS SAP to a domain hosted by the MCS provider. A result is confirmed to the requester. If the request is accepted, a user ID is assigned.
MCS-DETACH-USER	Deletes an MCS attachment that was created previously by invocation of MCS-Attach-User. This primitive may be requested by a user or initiated by a provider. It delivers an indication at every other MCS attachment to the same domain. If provider initiated, an indication is also delivered at the deleted attachment.
MCS-CHANNEL-JOIN	Used by an application client to join an appropriate channel whose use is defined by the application. This is a prerequisite for receiving data sent to the channel.
MCS-CHANNEL-LEAVE	Used by an application client to leave a previously joined channel and thus stop receiving data sent to that channel. The primitive may be user initiated (request only) or provider initiated (indication to affected user only).
MCS-CHANNEL-CONVENE	Used to allocate a new private channel with the requesting user as manager.
MCS-CHANNEL-ADMIT	This primitive enlarges the authorized user group of a private channel at the request of its manager. An indication is delivered to the MCS user added. That user may thereafter send data on the channel or join it as a receiver.
MCS-SEND-DATA	Used to transmit data to other members of a domain. If the sender is a member of the destination channel, it will not receive its own data indications. However, it will receive data indications from other sources addressed to that channel.
MCS-UNIFORM-SEND-DATA	Used to transmit data to other members of a domain in a uniformly sequenced manner, i.e. the data will be received in the same sequence by all members of the destination channel. The different data units from the domain clients will be forwarded to the Top MCS provider which will send them back to all clients in the same sequence. Uniform Sequencing of data is guaranteed only for data of the same priority on the same channel.
MCS-TOKEN-GRAB	Used to take exclusive control of a specific token.
MCS-TOKEN-INHIBIT	Used to take non-exclusive control of a specific token.
MCS-TOKEN-RELEASE	Used to free up a previously grabbed or inhibited token.
MCS-TOKEN-TEST	Used to check if a token is available.
MCS-TOKEN-PLEASE	Used to request a token held by another node.
MCS-TOKEN-GIVE	Used to pass a token directly from one node to another.

TABLE 6-2/T.126

Description of SI tokens and channels

Mnemonic	Mnemonic for static channel and token IDs	Resource IDs for multicast or private sessions	Description
SI-{MCS-USER-ID}-CHANNEL	–	–	Certain SIPDUs are sent directly to individual SICEs. To do this, the individual MCS-USER-ID channels of the peer SICEs in the MCS domain are used.
SI-CHANNEL	SI-CHANNEL-0	“C0”	This channel bears all SIPDUs to be broadcast to all peer SICEs in a domain.
SI-BITMAP-CREATE-TOKEN	SI-TOKEN-0	“T0”	This token is used to restrict bitmap creation to allow only a single bitmap creation at a time. This token shall always be used for bitmaps destined to hard-copy workspaces. For soft-copy workspaces, this token shall only be used if the Soft-Copy-Bitmap-No-Token-Protection capability is not present in the negotiated capability set.
SI-WORKSPACE-REFRESH-TOKEN	SI-TOKEN-1	“T1”	This token is used to allow a specific SICE to become the designated workspace refresher. The SICE that holds this token is responsible for refreshing workspaces when a new peer SICE enrolls into the conference.

TABLE 6-3/T.126

Use of MCS Data Primitives for SIPDUs

SIPDU	Channel	MDC data primitive		Priority
		Synchronized workspaces	Unsynchronized workspaces (or hard-copy exchanges for valid SIPDUs) or exchanges involving only creator-protected workspace planes (Note 3)	
ArchiveAcknowledgePDU	User ID Channel of source of ArchiveOpenPDU	MCS-Send-Data		Low
ArchiveClosePDU	SI-CHANNEL	MCS-Uniform-Send-Data		High, medium, low
ArchiveErrorPDU	User ID Channel of source of Archive command	MCS-Send-Data		Low

TABLE 6-3/T.126 (continued)

Use of MCS Data Primitives for SIPDUs

SIPDU	Channel	MDC data primitive		Priority
		Synchronized workspaces	Unsynchronized workspaces (or hard-copy exchanges for valid SIPDUs) or exchanges involving only creator-protected workspace planes (Note 3)	
ArchiveOpenPDU	SI-CHANNEL	MCS-Send-Data		Low
BitmapAbortPDU	User ID Channel of source of BitmapCreatePDU for request or SI-CHANNEL for announcement	MCS-Send-Data for request or MCS-Uniform-Send-Data for announcement	MCS-Send-Data or MCS-Uniform-Send-Data	(Note 1)
BitmapCheckpointPDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	High
BitmapCreatePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	(Note 1)
BitmapCreateContinuePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	(Note 1)
BitmapDeletePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	(Note 1)
BitmapEditPDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	(Note 1)
ConductorPrivilegeGrantPDU	SI-CHANNEL	MCS-Uniform-Send-Data		High
ConductorPrivilegeRequestPDU	User ID Channel of the SICE at the conducting node	MCS-Send-Data		High
DrawingCreatePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	Medium
DrawingDeletePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	Medium

TABLE 6-3/T.126 (continued)

Use of MCS Data Primitives for SIPDUs

SIPDU	Channel	MDC data primitive		Priority
		Synchronized workspaces	Unsynchronized workspaces (or hard-copy exchanges for valid SIPDUs) or exchanges involving only creator-protected workspace planes (Note 3)	
DrawingEditPDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	Medium
RemoteEventPermissionGrantPDU	SI-CHANNEL	MCS-Uniform-Send-Data		High
RemoteEventPermissionRequestPDU	User ID Channel of source of WorkspaceCreatePDU	MCS-Send-Data		High
RemoteKeyboardEventPDU	User ID Channel of source of WorkspaceCreatePDU	MCS-Send-Data		High
RemotePointingDeviceEventPDU	User ID Channel of source of WorkspaceCreatePDU	MCS-Send-Data		High
RemotePrintPDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	High (unsync'ed) High, medium, low (sync'ed)
SINonStandardPDU	(Note 2)	(Note 2)	(Note 2)	(Note 2)
WorkspaceCreatePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	High (unsync'ed) High, medium, low (sync'ed)
WorkspaceCreateAcknowledgePDU	User ID Channel of source of WorkspaceCreatePDU	–	MCS-Send-Data	High
WorkspaceDeletePDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	High (unsync'ed) High, medium, low (sync'ed)

TABLE 6-3/T.126 (concluded)

Use of MCS Data Primitives for SIPDUs

SIPDU	Channel	MDC data primitive		Priority
		Synchronized workspaces	Unsynchronized workspaces (or hard-copy exchanges for valid SIPDUs) or exchanges involving only creator-protected workspace planes (Note 3)	
WorkspaceEditPDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	High (sync'ed) High, Medium, Low (sync'ed)
WorkspacePlaneCopyPDU	SI-CHANNEL	MCS-Uniform-Send-Data	MCS-Send-Data or MCS-Uniform-Send-Data	(Note 1)
WorkspaceReadyPDU	SI-CHANNEL	–	MCS-Send-Data	High
WorkspaceRefreshStatusPDU	SI-CHANNEL	MCS-Uniform-Send-Data		High
<p>NOTES</p> <p>1 This indicates that the priority shall be Medium if the destination plane has the annotation usage designator set or in the case of bitmap operations, if the bitmap destination is the pointer plane, and Low otherwise.</p> <p>2 The use of the SINonStandardPDU is beyond the scope of this Recommendation.</p> <p>3 The SICE that created the workspace shall not use MCS-Send-Data unless it guarantees to never change the plane protection of a plane to unprotected.</p>				

7 Use of GCC

An SICE uses the following GCC service primitives to enroll and exchange capabilities, to allocate unique handles, to determine conductorship mode, and to allocate and register dynamic channels and tokens. Table 7-1 describes each of the primitives used by an SICE.

7.1 GCC unique handles

All handles used in the SI protocol are acquired from GCC using the GCC-Registry-Allocate-Handle primitive. Handles can be allocated at any time not just immediately prior to their use. It is suggested that applications allocate blocks of handles to minimize network traffic and perform this operation when idle to avoid associated latencies during periods of protocol activity.

8 Protocol specification

8.1 SI protocol parameters

The SI protocol has three parameters as shown in Table 8-1.

TABLE 7-1/T.126

GCC primitives needed by an SICE

GCC primitive	Default session	Multicast or private sessions
GCC-Application-Permission-To-Enroll	Indication	Indication
GCC-Application-Enroll	Request Confirm	Request Confirm
GCC-Application-Roster-Report	Indication	Indication
GCC-Registry-Retrieve-Entry	–	Request Confirm
GCC-Registry-Register-Channel	–	Request Confirm
GCC-Registry-Assign-Token	–	Request Confirm
GCC-Registry-Allocate-Handle	Request Confirm	Request Confirm
GCC-Conductor-Assign	Indication	Indication
GCC-Conductor-Release	Indication	Indication
GCC-Conductor-Permission-Grant	Indication	Indication

8.2 SI application enrollment and initialization

An SI session may be initiated locally by a user application, or remotely through use of the GCC-Application-Invoke mechanism. In both cases the parameters shown in Table 8-1 are passed to the SICE. The action to be taken by the SICE to identify the initial set of resources to be used for the session is determined by the session mode.

An SICE created in any mode must first establish a GCCSAP to allow it to communicate with the GCC provider at that node. When the node joins a conference, the local GCC provider will issue a GCC-Application-Permission-to-Enroll indication with the Grant/Revoke flag set to Grant. The SICE shall then issue a GCC-Application-Enroll request, regardless of whether the user application wishes to enroll at that time. If the user application does not wish to enroll, the SICE shall specify the conference ID and set the Enroll/Un-enroll flag in the GCC-Application-Enroll request to Un-enroll. No other parameters are required. The user application may instruct its SICE to enroll at any time subsequently, unless permission is revoked by receipt of a GCC-Application-Permission-to-Enroll indication with the Grant/Revoke flag set to Revoke.

If the user application wishes to receive information on all SI sessions in progress before deciding which session to participate in, the SICE may enroll Inactive, specifying the Session Key without a Session ID.

When the application decides to become active, the SICE shall issue an MCS-Attach-User request to the MCS provider, using the Conference ID contained in the GCC-Application-Permission-to-Enroll indication as the Domain Selector. On receipt of a successful MCS-Attach-User confirm in response, the SICE shall join the User ID channel indicated by issuing an MCS-Channel-Join request.

Subsequent operations are dependent on the session mode.

TABLE 8-1/T.126

SI Application protocol parameters

Parameter	Description
sessionMode	<p>This parameter can have one of the following three values:</p> <p><i>static</i>: This value indicates that the SICE should enroll using an Session Key composed of the SI object identifier and the sessionID parameter. It shall use the static predefined SI-CHANNEL.</p> <p><i>dynamic multicast</i>: This value indicates that the SICE should enroll using an application key composed of the SI object identifier and the sessionID parameter. All channel and token resources are dynamic and are allocated by the creator of the multicast session using the MCS-CHANNEL-JOIN mechanism and GCC Registry mechanism respectively. Members of a multicast session determine token IDs via the GCC Registry.</p> <p><i>dynamic private</i>: This value indicates that the SICE should enroll using a Session Key composed of the SI object identifier and the sessionID parameter.</p> <p>All token and channel resources are dynamic and are allocated by the Private Convenor SICE using the GCC Registry mechanism and MCS-CHANNEL-CONVENE mechanism respectively. The SICE creating the session may then admit to the SI-CHANNEL all peer SICES whose GCC User Ids are present in the admitList protocol parameter.</p> <p>A Private Member SICE must wait to be admitted by the convenor of the private SI-CHANNEL before attempting to join it.</p>
sessionID	<p>This parameter is used to differentiate the resources used by multiple sessions of the protocol that may be in existence simultaneously within the same MCS domain. The MCS Channel ID assigned to the SI-CHANNEL is used as the Session ID, since this is guaranteed to be unique within the conference domain.</p> <p>The SessionID must be specified if the application wishes to participate in a static session or in an existing multicast or private session. It is omitted if the application wishes to create a new multicast or private session.</p>
admitList	<p>SessionMode = private and SessionID omitted</p> <p>List of GCC User IDs corresponding to the SICES to be admitted to the privately convened channel.</p> <p>else</p> <p>Omitted</p>

8.2.1 Static mode

After obtaining an MCS User ID, the SICE shall join the static SI-CHANNEL by issuing an MCS-Channel-Join request specifying the SI-CHANNEL as the channel to join. See Figure 8-1. No dynamic tokens need to be assigned at this time. The SICE shall then enroll by issuing a GCC-Application-Enroll request to the GCC provider, with the parameters specified in Table 8-2. The Active/Inactive flag shall be set to Active, the Session ID shall be specified as part of the Session Key, the Start-Up Channel shall be specified as Static and the full Application Capabilities List shall be provided. The Conducting Operation flag is set only if this SICE is capable of responding to ConductorPrivilegeRequestPDUs.

TABLE 8-2/T.126

Parameters for GCC-Application-Enroll request

Parameter	Contents
Conference ID	Provided by GCC-Application-Permission-To-Enroll indication.
Session Key	{itu recommendation t si} and sessionID if this protocol parameter was provided to this session of the protocol.
Active/Inactive	ACTIVE when indicating that the SICE has joined the SI-CHANNEL and determined SI token IDs. INACTIVE in the case of dynamic channels and tokens when enrolling prior to joining the SI-CHANNEL or determining the SI token IDs.
Application User ID	Provided by MCS-Attach-User confirm.
Conducting Operation Capable	Indicates the ability to respond to the ConductorPrivilegeRequestPDU with ConductorPrivilegeGrantPDUs.
Non-Collapsing Capabilities List	There are no Non-Collapsing Capabilities specified by this Recommendation. A particular SICE implementation may include non-standard Non-Collapsing Capabilities in this field.
Application Capability List	See Tables 8-5 and 8-6; Omitted if Active/Inactive flag is set to INACTIVE.
Enroll/Un-enroll	ENROLL

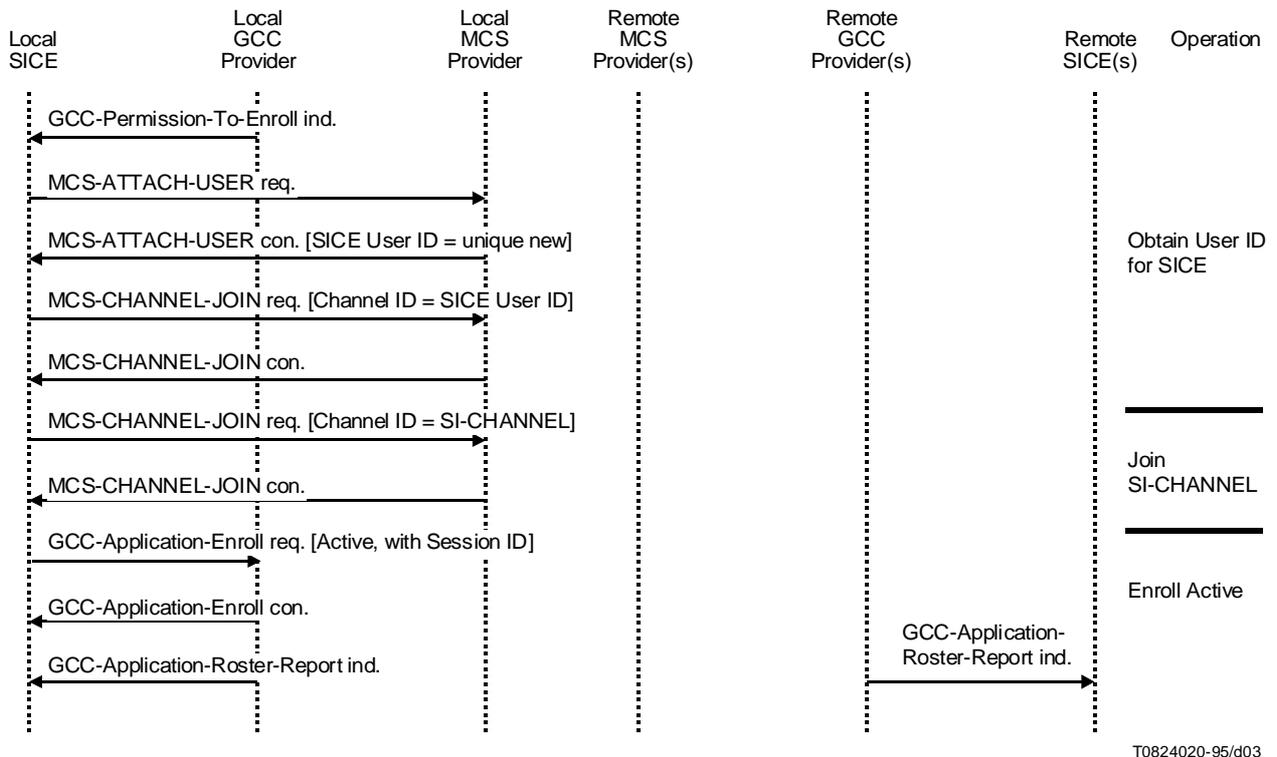


FIGURE 8-1/T.126

Static session protocol initiation sequence

8.2.2 Multicast mode

After obtaining an MCS User ID, the SICE issues a GCC-Application-Enroll request with the Active/Inactive flag set to Inactive and specifying the Session Key without a Session ID. The SICE shall examine the sessionID parameter to determine whether it is to participate in an existing multicast session (as a multicast member) or to create a new one (as a multicast creator). See Figures 8-2 and 8-3.

If the sessionID parameter is omitted, the SICE shall attempt to create a new multicast session by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Inactive and specifying the Session Key without a Session ID. The SICE shall then attempt to allocate a multicast channel for the session by issuing an MCS-Channel-Join request with Channel ID = 0. The returned MCS-Channel-Join confirm, if successful, contains the assigned Channel ID which shall be used as the SI-CHANNEL.

The SICE shall enroll actively by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Active, specifying the Channel ID of the SI-CHANNEL as Session ID and providing the full Application Capabilities List. The Start-Up Channel shall be specified as Dynamic Multicast.

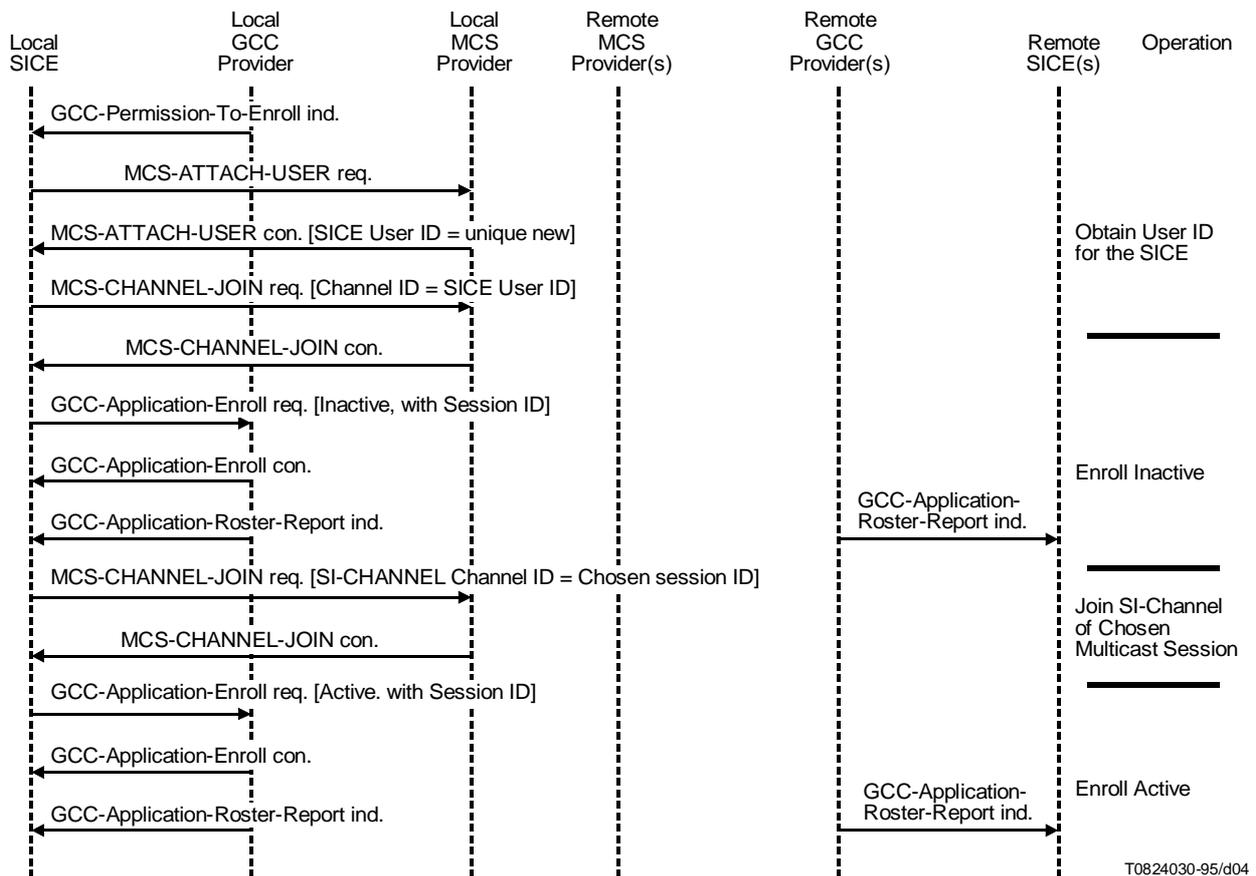


FIGURE 8-2/T.126

Multicast session protocol initiation sequence (member)

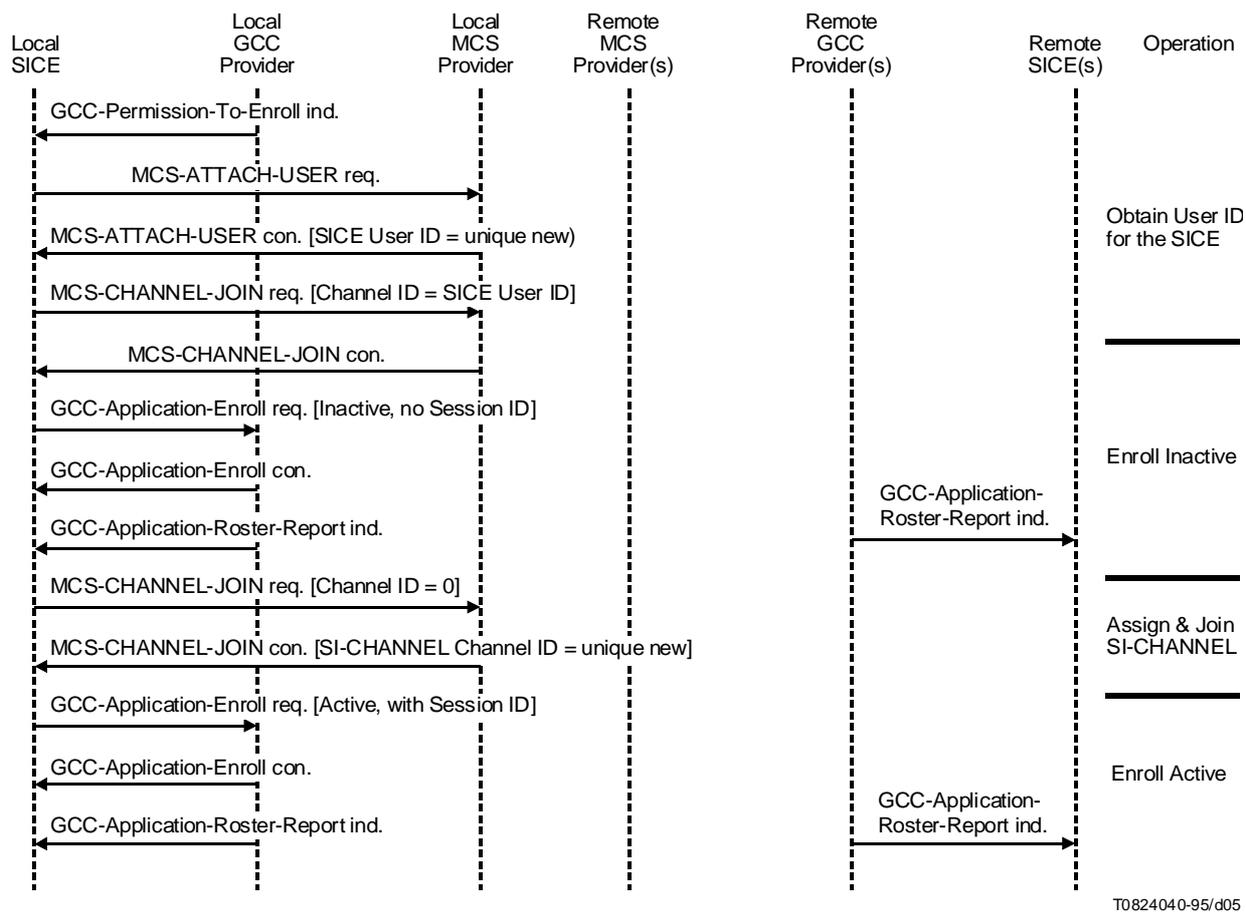


FIGURE 8-3/T.126
Multicast session protocol initiation sequence (creator)

If the sessionID parameter is present, the SICE shall attempt to join the session indicated by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Inactive and specifying the Session Key with the requisite Session ID. It shall then issue an MCC-Channel-Join request, specifying the Session ID of the chosen session as the Channel-ID parameter. This Channel ID shall be used as the SI-CHANNEL.

The SICE shall then enroll actively by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Active, specifying the chosen SessionID as part of the Session Key and providing the full Application Capabilities List.

Both the convening and member SICEs must wait until they have received a GCC-Application-Roster-Report indication which contains an entry corresponding to themselves as a result of enrolling with the Active/Inactive flag set to Active. They are then able to determine the dynamic tokens to be used for the SI session via the registry. This requires a registry key to be formed. The process for forming this key is described in 8.2.4. The process for determination of dynamic Token IDs is described in 8.2.5.

8.2.3 Private mode

After obtaining an MCS User ID, the SICE shall examine the sessionID parameter to determine whether it is to participate in an existing private session (as a private member) or to create a new session (as a private convenor).

If the sessionID parameter is omitted, the SICE shall attempt to create a new session by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Inactive and specifying the Session Key. The Session ID and Application Capabilities List are omitted. The private convenor SICE shall then attempt to obtain a private channel by issuing an MCS-Channel-Convene request. If successful, the returned MCS-Channel-Convene confirm contains the ID of the channel allocated. The channel shall be used as the SI-CHANNEL. The SICE shall then join this channel by issuing an MCS-Channel-Join request, specifying the Channel ID returned in the MCS-Channel-Convene confirm.

The SICE shall then enroll actively by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Active, specifying the Channel ID of the SI-CHANNEL as Session ID, specifying the Start-Up Channel as Dynamic Private and providing the full Application Capabilities List.

The SICE shall then attempt to invite other nodes to participate in the private session by issuing a GCC-Application-Invoke request, specifying a list of GCC User IDs to be invited or NULL (indicating that all nodes in the conference are to be invited) as the destination Nodes parameter. It shall also specify Dynamic Private as the Start-Up Channel in the Application Protocol entry.

The convening SICE must then wait until it has received a GCC-Application-Roster-Report indication containing the MCS User IDs of the SICES to be invited to the private session (a timeout is recommended in case there is no peer SICE at one or more of the invited nodes). The SICE shall then issue an MCS-Channel-Admit request for the private channel, specifying the MCS -User IDs of the SICES to be invited as the list of MCS User IDs. The list of invitees is provided by the user application.

If the sessionID parameter is present, the SICE shall attempt to join the session indicated by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Inactive and specifying the Session Key with the requisite Session ID. The Application Capabilities List is omitted. The SICE shall then wait until it has received an MCS-Channel-Admit indication from the peer SICE at the node convening the private session. It then attempts to join the channel indicated in this primitive by issuing an MCS-Channel-Join request. This channel shall be used as the SI-CHANNEL.

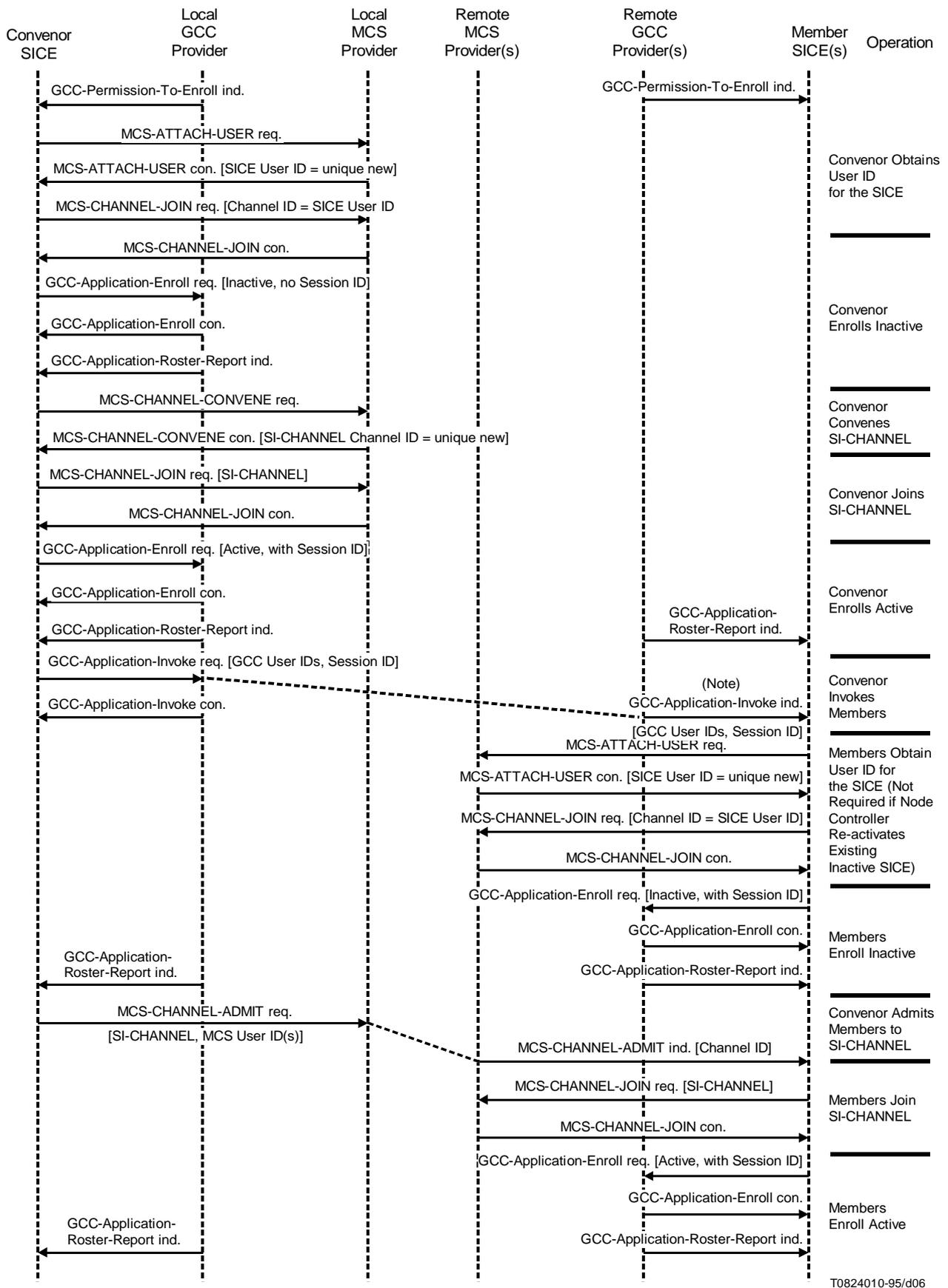
The SICE shall then enroll actively by issuing a GCC-Application-Enroll request with the Active/Inactive flag set to Active, specifying the chosen Session ID and providing the full Application Capabilities List.

Following receipt of a GCC-Application-Roster-Report indication containing its MCS User ID, the private member SICE shall re-enroll with the Active/Inactive flag set to Active, continuing to include the Session ID as part of the Session Key and providing the full Application Capabilities List.

Both the convening and member SICES must wait until they have received a GCC-Application-Roster-Report indication which contains an entry corresponding to themselves as a result of enrolling with the Active/Inactive flag set to Active. They are then able to determine the dynamic tokens to be used for the SI session via the registry. This requires a registry key to be formed. The process for forming this key is described in 8.2.4. The process for determination of dynamic Token IDs is described in 8.2.5. See Figure 8-4.

8.2.4 Forming registry keys

For each SI dynamic resource (channel or token), the SICE shall form a key to be used in attempting to determine the ID of this resource. Registry keys are composed of an abstract type consisting of the Session Key for this session of the application (the same key used in the application enrollment) with a Resource ID. The value of the Resource ID for each channel or token is defined in clause 6.



NOTE – GCC-Application-Invoke ind. is sent to the Node Controller.

FIGURE 8-4/T.126

Private session protocol initiation sequence

8.2.5 Establishing dynamic tokens

The SICE shall also attempt to establish the token ID for all dynamic tokens that it might need. For each token, this is done by first issuing a GCC-Registry-Assign-Token request to the GCC provider. The parameters used in this request are as shown in Table 8-4. If the Result parameter returned in the GCC-Registry-Assign-Token confirm is “successful,” the Token ID contained in the returned confirm primitive is used as the Token ID for the token corresponding to the Resource ID used in the Registry Key. If the Result parameter returned in the GCC-Registry-Assign-Token confirm is “index owned by another agent,” then the SICE shall issue a GCC-Registry-Retrieve-Entry request to the GCC provider. The parameters used in this request are as shown in Table 8-3. The Token ID contained in the resulting GCC-Registry-Retrieve-Entry confirm is used as the Token ID for the token corresponding to the Resource ID in the Registry Key.

TABLE 8-3/T.126

Parameters for GCC-Registry-Retrieve-Entry request

Parameter	Contents
Conference ID	Provided by GCC-Application-Permission-To-Enroll indication
Registry Key	Registry Key formed as described in 8.2.4

TABLE 8-4/T.126

Parameters for GCC-Registry-Assign-Token request

Parameter	Contents
Conference ID	Provided by GCC-Application-Permission-To-Enroll indication
Registry Key	Registry Key formed as described in 8.2.4

8.2.6 SI capabilities

SICES use the application enrollment mechanism for capabilities negotiation. The Application Capabilities List parameter of the GCC-Application-Enroll request PDU is used to specify the list of capabilities supported by the local SICE.

The Application Capabilities List included in the GCC-Application-Enroll request is composed of list capabilities to be advertised as supported by this SICE. Note that at least one of the following capabilities shall be included in this list: Hard-Copy-Workspace or Soft-Copy-Workspace. If Soft-Copy-Workspace is included, then at least one of the following shall also be included: Soft-Copy-Image or Soft-Copy-Annotation.

In Table 8-5 certain capabilities are indicated as being dependent on other capabilities. This implies that the capability shall not be included in the Application Capabilities List unless the capability on which it depends is also included.

The result of the capabilities negotiation procedure is made apparent to the SICE by the receipt of a GCC-Application-Roster-Report indication from the GCC provider. The application roster report includes the Application Roster for peer SICEs within the indicated conference – that is, SICEs which have designated the same Session Key. The Application Roster includes a list of nodes for which a peer SICE has enrolled. For each node, the list contains the GCC User ID of that node, and the Application User ID of the peer SICE at that node. The Application Roster also includes an instance number, a flag indicating whether new nodes have been added since the last instance, a flag indicating whether nodes have been removed since the last instance, a flag indicating whether the Application Capabilities List has been updated since the last instance, and if so, the new Application Capabilities List. In the case of a newly enrolling SICE, the Application Capabilities List is always updated since previous instances of the list are not available to this SICE.

When first enrolling, an SICE shall ignore received GCC-Application-Roster-Report indications in which the SICE is not included (if there is no entry in the application roster which has the GCC User ID of the local node and the Application Protocol Entity ID of the local SICE as indicated in the GCC-Application-Enroll confirm). Once a roster is received which does include the local SICE, that SICE is now considered part of the conference and may proceed examining the roster to determine how to proceed.

The Application Capabilities List received as part of the GCC-Application-Roster-Report indication corresponds to the collapsed Application Capabilities Lists of all enrolled peer SICEs. That is, the list includes an entry for each capability which has been issued by any peer SICE. For each entry, it includes the Capability ID, the number of peer SICEs (including the local one) which had advertised this capability as part of their enroll procedure, and, in the case of capabilities of the MIN or MAX class, the minimum or maximum value of the parameter among all peer SICEs which advertised this capability. For each capability, the rules used to determine the result of the capability exchange is indicated in the Node Count Rule column of Table 8-5. The notation in this column explaining how these rules are to be applied is described in Table 8-6.

At any time while an SICE is enrolled in a conference, the SICE may receive additional GCC-Application-Roster-Report indications from the GCC provider indicating that the contents of the roster have changed. This may be due to new peer SICEs enrolling in the conference, peer SICEs leaving the conference, or peer SICEs having modified their enrollment information.

If, at any time, the local SICE desires to indicate a change in its Application Capabilities List, it may re-enroll. This is done by re-issuing a GCC-Application-Enroll request to the GCC provider with the Enroll/Un-enroll flag set to Enroll and the updated Application Capabilities List, as well as the other parameters normally included in the enroll request. The result of this is potentially a change to the Application Roster, resulting in receipt of a GCC-Application-Roster-Report indication to the local SICE as well as all other peer SICEs in the conference.

If, at any time, the local SICE desires to terminate its participation in the conference, it may un-enroll by issuing a GCC-Application-Enroll request to the GCC provider with the Enroll/Un-enroll flag set to Un-enroll. In this case, no other parameters need be included.

If at any time the SICE receives a GCC-Application-Roster-Report indication in which it is no longer included (if there is no entry in the application roster which has the GCC User ID of the local node and the Application Protocol Entity ID of the local SICE as indicated in the GCC-Application-Enroll confirm), the SICE shall immediately issue an MCS-Detach-User request to the MCS provider to detach the connection to the specified conference. At this point, the SICE is no longer considered enrolled to the conference. The SICE may re-issue a GCC-Application-Enroll to attempt to become re-enrolled in the conference.

If at any time the SICE receives a GCC-Application-Permission-To-Enroll indication with the Grant/Revoke flag set to Revoke, if the SICE has already attached to MCS, it shall immediately issue an MCS-Detach-User request to the MCS provider to detach the connection to the specified conference. At this point, the SICE is no longer considered enrolled to the conference and shall not attempt to re-enroll.

TABLE 8-5/T.126

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
<p>Hard-Copy-Image:</p> <p>This capability is used to negotiate the use of hard-copy image exchanges. This capability implies a maximum image size of 1728 horizontal by 2300 vertical. It also implies the ability to support unscaled image bitmap creation using either Uncompressed or T.4 (G3) formats with a single bitplane and either fax1 or fax2 pixel aspect ratios.</p>	1	L	>1	–	–
<p>Hard-Copy-Image-Bitmap-Max-Width (1728):</p> <p>This capability is used to negotiate the maximum width of an image bitmap for hard-copy image exchanges. This dimension is relative to the pixel aspect ratio of the image.</p>	2	MIN	= D	(1729..21845)	Hard-Copy-Image
<p>Hard-Copy-Image-Bitmap-Max-Height (2300):</p> <p>This capability is used to negotiate the maximum height of an image bitmap for hard-copy image exchanges. This dimension is relative to the pixel aspect ratio of the image.</p>	3	MIN	= D	(2301..21845)	Hard-Copy-Image
<p>Hard-Copy-Image-Bitmap-Any-Aspect-Ratio:</p> <p>This capability is used to negotiate the ability to transmit image bitmaps to a hard-copy workspace with an arbitrary aspect ratio.</p>	4	L	= D	–	Hard-Copy-Image
<p>Hard-Copy-Image-Bitmap-Format-T.6:</p> <p>This capability is used to negotiate the ability to support bitmap creation using T.6 (G4) image compression format with a single bitplane and either fax1 or fax2 pixel aspect ratios.</p>	5	L	= D	–	Hard-Copy-Image
<p>Hard-Copy-Image-Bitmap-Format-T.82:</p> <p>This capability is used to negotiate the ability to support bitmap creation using T.82 (JBIG) image compression format. This capability implies the ability to handle 1 bit plane with 1:1 pixel aspect ratio and the ability to only handle bitmaps encoded without the use of JBIG resolution reduction.</p>	6	L	= D	–	Hard-Copy-Image

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
Soft-Copy-Workspace: This capability is used to negotiate the ability to support at least one workspace for soft-copy information. This capability implies a maximum workspace size of 384 horizontal by 288 vertical with workspace background Colors Black and White. Presence of this capability also implies that one of the capabilities Soft-Copy-Annotation or Soft-Copy-Image shall also be included in the Application Capabilities List.	7	L	>1	–	–
Soft-Copy-Workspace-Max-Width (384): This capability is used to negotiate the maximum workspace width. This dimension is relative to a 1:1 pixel aspect ratio (square pixels).	8	MIN	= D	(385..21845)	Soft-Copy-Workspace
Soft-Copy-Workspace-Max-Height (288): This capability is used to negotiate the maximum workspace height. This dimension is relative to a 1:1 pixel aspect ratio (square pixels).	9	MIN	= D	(289..21845)	Soft-Copy-Workspace
Soft-Copy-Workspace-Max-Planes (1): This capability is used to negotiate the maximum number of planes allowed in any workspace.	10	MIN	= D	(2..256)	Soft-Copy-Workspace
Soft-Copy-Color-16: This capability is used to negotiate the use of the 16-color palette for use in workspace backgrounds or, if the Soft-Copy-Annotation capability is negotiated, in drawing elements.	11	L	= D	–	Soft-Copy-Workspace
Soft-Copy-Color-202: This capability is used to negotiate the use of the 202 color palette for use in workspace backgrounds or, if the Soft-Copy-Annotation capability is negotiated, in drawing elements.	12	L	= D	–	Soft-Copy-Workspace
Soft-Copy-Color-True: This capability is used to negotiate the use of the true Color (24-bit RGB) as well as the use of the 202 Color palette for use in workspace backgrounds or, if the Soft-Copy-Annotation capability is negotiated, in drawing elements.	13	L	= D	–	Soft-Copy-Workspace
Soft-Copy-Plane-Editing: This capability is used to negotiate the ability to declare any workspace plane to be editable.	14	L	= D	–	Soft-Copy-Workspace

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
Soft-Copy-Scaling: This capability is used to negotiate the ability to declare a scaling rectangle during creation of soft-copy bitmaps. Without this capability, bitmaps are applied to the destination workspace without scaling (other than that required for non 1:1 pixel aspect ratios).	15	L	= D	–	Soft-Copy-Workspace
Soft-Copy-Bitmap-No-Token-Protection: This capability is used to negotiate the ability to transmit soft-copy bitmaps of any variety without the need to hold the SI-BITMAP-CREATE-TOKEN.	16	L	= D	–	Soft-Copy-Workspace
Soft-Copy-Pointing: This capability is used to negotiate the use of pointer bitmaps on soft-copy workspaces. Successful negotiation of this capability allows the following coding formats and associated parameter constraints for pointer bitmaps: – Uncompressed format of either 8-bit greyscale, RGB 4:4:4 or 1, 4, or 8 bit palettized with a 1:1 pixel aspect ratio. This capability implies the ability to handle pointer bitmaps up to a maximum size of 32 by 32 pixels.	17	L	>1	–	Soft-Copy-Workspace
Soft-Copy-Pointing-Bitmap-Max-Width (32): This capability is used to negotiate the maximum width of a pointer bitmap. This dimension is relative to a 1:1 pixel aspect ratio (square pixels).	18	MIN	= D	(33..21845)	Soft-Copy-Pointing
Soft-Copy-Pointing-Bitmap-Max-Height (32): This capability is used to negotiate the maximum height of a pointer bitmap. This dimension is relative to a 1:1 pixel aspect ratio (square pixels).	19	MIN	= D	(33..21845)	Soft-Copy-Pointing
Soft-Copy-Pointing-Bitmap-Format-T.82: This capability is used to negotiate the ability to use T.82 (JBIG) compression format for encoding pointer bitmaps. This capability implies the ability to handle either 8-bit greyscale, or up to 8 palettized bitplanes with a 1:1 pixel aspect ratio and the ability to only handle bitmaps encoded without the use of JBIG resolution reduction.	20	L	= D	–	Soft-Copy-Pointing

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
<p>Soft-Copy-Annotation:</p> <p>This capability is used to negotiate the use of annotation on soft-copy workspaces. The presence of this capability in the negotiated capability set implies the ability to create workspaces with annotation specified as the usage-designator of workspace planes. Successful negotiation of this capability also allows the following coding formats and associated parameter constraints for annotation bitmaps:</p> <ul style="list-style-type: none"> – Uncompressed bitmap format of either 8-bit greyscale, RGB 4:4:4 or 1, 4, or 8 bit palettized raster and Color formats with a 1:1 pixel aspect ratio. <p>This capability also implies the ability to support the creation of drawings using the DrawingCreatePDU with a pen thickness of 3 to 16 pixels, and a round pen nib.</p>	21	L	>1	–	Soft-Copy-Workspace
<p>Soft-Copy-Annotation-Bitmap-Max-Width (384):</p> <p>This capability is used to negotiate the maximum width of an annotation bitmap. This dimension is relative to a 1:1 pixel aspect ratio (square pixels).</p>	22	MIN	= D	(385..65536)	Soft-Copy-Annotation
<p>Soft-Copy-Annotation-Bitmap-Max-Height (288):</p> <p>This capability is used to negotiate the maximum height of an annotation bitmap. This dimension is relative to a 1:1 pixel aspect ratio (square pixels).</p>	23	MIN	= D	(289..65536)	Soft-Copy-Annotation
<p>Soft-Copy-Annotation-Drawing-Pen-Min-Thickness (3):</p> <p>This capability is used to negotiate the minimum thickness in pixels of lines drawn using the DrawingCreatePDU.</p>	24	MAX	= D	(1..2)	Soft-Copy-Annotation
<p>Soft-Copy-Annotation-Drawing-Pen-Max-Thickness (16):</p> <p>The capability is used to negotiate the maximum thickness in pixels of lines drawn using the DrawingCreatePDU.</p>	25	MIN	= D	(17..255)	Soft-Copy-Annotation
<p>Soft-Copy-Annotation-Drawing-Ellipse:</p> <p>This capability is used to negotiate the ability to use the ellipse drawing type when using the DrawingCreatePDU.</p>	26	L	= D	–	Soft-Copy-Annotation
<p>Soft-Copy-Annotation-Drawing-Pen-Square-Nib:</p> <p>This capability is used to negotiate the ability to use a square nib shape in creation of lines drawn using the DrawingCreatePDU.</p>	27	L	= D	–	Soft-Copy-Annotation

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
<p>Soft-Copy-Annotation-Drawing-Highlight:</p> <p>This capability is used to negotiate the ability to make use of the Highlight line style for drawing.</p>	28	L	= D	–	Soft-Copy-Annotation
<p>Soft-Copy-Annotation-Bitmap-Format-T.82:</p> <p>This capability is used to negotiate the ability to use T.82 (JBIG) compression format for encoding annotation bitmaps. This capability implies the ability to handle either 8-bit greyscale, or up to 8 palettized bitplanes with a 1:1 pixel aspect ratio and the ability to only handle bitmaps encoded without the use of JBIG resolution reduction.</p>	29	L	= D	–	Soft-Copy-Annotation
<p>Soft-Copy-Image:</p> <p>This capability is used to negotiate the use of image bitmaps on soft-copy workspaces. The presence of this capability in the negotiated capability set implies the ability to create workspaces with image specified as the usage-designator of workspace planes. Successful negotiation of this capability allows the following coding formats and associated parameter constraints for image bitmaps:</p> <ol style="list-style-type: none"> 1) JBIG: This capability implies the ability to handle either 8-bit greyscale, RGB 4:4:4, or up to 8 palettized bitplanes and the ability to only handle bitmaps encoded without the use of JBIG resolution reduction. Both 1:1 and CIF pixel aspect ratios shall be supported. 2) JPEG: This capability implies the ability to handle the Baseline DCT encoding mode, with baseline sequential transmission and 8-bit/sample data precision in component interleaved format only, using a Color space and Color resolution mode of YCbCr 4:2:2, or greyscale. Both 1:1 and CIF pixel aspect ratios shall be supported. 3) Uncompressed: This capability implies the ability to handle 8-bit greyscale, RGB 4:4:4, YCbCr 4:2:2, or palettized 1, 4, or 8 bits per pixel. Both 1:1 and CIF pixel aspect ratios shall be supported. 	30	L	>1	–	Soft-Copy-Workspace
<p>Soft-Copy-Image-Bitmap-Max-Width (384):</p> <p>This capability is used to negotiate the maximum workspace width for soft-copy image bitmap exchanges. This dimension is relative to the pixel aspect ratio of the image bitmap.</p>	31	MIN	= D	(385..65536)	Soft-Copy-Image
<p>Soft-Copy-Image-Bitmap-Max-Height (288):</p> <p>This capability is used to negotiate the maximum workspace height for soft-copy image bitmap exchanges. This dimension is relative to the pixel aspect ratio of the image bitmap.</p>	32	MIN	= D	(289..65536)	Soft-Copy-Image

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
Soft-Copy-Image-Bitmap-Any-Aspect-Ratio: This capability is used to negotiate the ability to transmit image bitmaps to a soft-copy workspace with an arbitrary aspect ratio.	33	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.82- Differential: This capability is used to negotiate the ability to use resolution reduction (differential layers) when encoding a JBIG format image bitmap.	34	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.82-Differential-Deterministic-Prediction: This capability is used to negotiate the ability to use deterministic prediction when encoding a JBIG format image bitmap using resolution reduction (differential layers).	35	L	= D	–	Soft-Copy-Image- Bitmap- Format- T.82- Differential
Soft-Copy-Image-Bitmap-Format-T.82-12-Bit-Grey-Scale: This capability is used to negotiate the ability to use 12-bit planes when encoding a JBIG format image bitmap.	36	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Extended-Sequential-DCT: This capability is used to negotiate the ability to use Extended Sequential DCT mode when encoding a JPEG format image bitmap.	37	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Progressive-DCT: This capability is used to negotiate the ability to use Progressive DCT mode when encoding a JPEG format image bitmap.	38	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Spatial-DPCM: This capability is used to negotiate the ability to use Spatial DPCM mode when encoding a JPEG format image bitmap.	39	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Differential-Sequential-DCT: This capability is used to negotiate the ability to use Differential Sequential DCT mode when encoding a JPEG format image bitmap.	40	L	= D	–	Soft-Copy-Image

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
Soft-Copy-Image-Bitmap-Format-T.81-Differential-Progressive-DCT: This capability is used to negotiate the ability to use Differential Progressive DCT mode when encoding a JPEG format image bitmap.	41	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Differential-Spatial-DPCM: This capability is used to negotiate the ability to use Differential Spatial DPCM mode when encoding a JPEG format image bitmap.	42	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Extended-Sequential-DCT-Arithmetic: This capability is used to negotiate the ability to use Extended Sequential DCT mode using Arithmetic encoding when encoding a JPEG format image bitmap.	43	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Progressive-DCT-Arithmetic: This capability is used to negotiate the ability to use Progressive DCT mode using Arithmetic encoding when encoding a JPEG format image bitmap.	44	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Spatial-DPCM-Arithmetic: This capability is used to negotiate the ability to use Spatial DPCM mode using Arithmetic encoding when encoding a JPEG format image bitmap.	45	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Differential-Sequential-DCT-Arithmetic: This capability is used to negotiate the ability to use Differential Sequential DCT mode using Arithmetic encoding when encoding a JPEG format image bitmap.	46	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81- Differential-Progressive-DCT-Arithmetic: This capability is used to negotiate the ability to use Differential Progressive DCT mode using Arithmetic encoding when encoding a JPEG format image bitmap.	47	L	= D	–	Soft-Copy-Image

TABLE 8-5/T.126 (continued)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
Soft-Copy-Image-Bitmap-Format-T.81-Differential-Spatial-DPCM-Arithmetic: This capability is used to negotiate the ability to use Differential Spatial DPCM mode using Arithmetic encoding when encoding a JPEG format image bitmap.	48	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-YCbCr-4:2:0: This capability is used to negotiate the ability to use a chroma format of YCbCr 4:2:0 when encoding a JPEG format image bitmap.	49	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-YCbCr-4:4:4: This capability is used to negotiate the ability to use a chroma format of YCbCr 4:4:4 when encoding a JPEG format image bitmap.	50	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-RGB-4:4:4: This capability is used to negotiate the ability to use a chroma format of RGB 4:4:4 when encoding a JPEG format image bitmap.	51	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-CIELab-4:2:0: This capability is used to negotiate the ability to use a chroma format of CIELab 4:2:0 when encoding a JPEG format image bitmap.	52	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-CIELab-4:2:2: This capability is used to negotiate the ability to use a chroma format of CIELab 4:2:2 when encoding a JPEG format image bitmap.	53	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-CIELab-4:4:4: This capability is used to negotiate the ability to use a chroma format of CIELab 4:4:4 when encoding a JPEG format image bitmap.	54	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-T.81-Non-Interleaved: This capability is used to negotiate the ability to use non-interleaved ordering of Color components.	55	L	= D	–	Soft-Copy-Image

TABLE 8-5/T.126 (concluded)

Application capabilities list elements

Capability name (default value if numeric class): Description	ID	Class	SICE count rule	Parameter	Dependency
Soft-Copy-Image-Bitmap-Format-Uncompressed-YCbCr-4:2:0: This capability is used to negotiate the ability to use a chroma format of YCbCr 4:2:0 when encoding an Uncompressed format image bitmap.	56	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-Uncompressed-YCbCr-4:4:4: This capability is used to negotiate the ability to use a chroma format of YCbCr 4:4:4 when encoding an Uncompressed format image bitmap.	57	L	= D		Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-Uncompressed-CIELab-4:2:0: This capability is used to negotiate the ability to use a chroma format of CIELab 4:2:0 when encoding an Uncompressed format image bitmap.	58	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-Uncompressed-CIELab-4:2:2: This capability is used to negotiate the ability to use a chroma format of CIELab 4:2:2 when encoding an Uncompressed format image bitmap.	59	L	= D	–	Soft-Copy-Image
Soft-Copy-Image-Bitmap-Format-Uncompressed-CIELab-4:4:4: This capability is used to negotiate the ability to use a chroma format of CIELab 4:4:4 when encoding an Uncompressed format image bitmap.	60	L	= D	–	Soft-Copy-Image
Archive-Support: This capability is used to negotiate the support of archives.	61	L	–	–	–
Non-Standard-Capability: This capability is used to negotiate non-standard functions. Any number of these may appear in the Application Capabilities List as long as they each have a unique Non-Standard-Identifier. The interpretation of these capabilities is not covered by this Recommendation.	Non-Standard Identifier	–	–	–	–

TABLE 8-6/T.126

Capability list notation

Class	L	Logical
	MIN	Unsigned Minimum
	MAX	Unsigned Maximum
SICE Count Rule	>1	The Number-of-Entities parameter returned by the GCC-Application-Roster-Report Indication for this capability must be greater than 1 for it to be considered established for the conference.
	= D	In the case of Logical capabilities, the Number-of-Nodes parameter returned by the GCC-Application-Roster-Report Indication for this capability must be equal to that of the dependent capability for it to be considered established. In the case of numerical capabilities (MIN or MAX), if the Number-of-Nodes parameter is equal to that of the dependent capability, the result of the MIN or MAX operation is established as the conference-wide value, otherwise the default value of the capability is established.

8.3 Workspaces

A workspace may be created if the Soft-Copy-Workspace capability is present in the negotiated capability set. Multiple workspaces may coexist if sufficient storage exists within all peer SICEs. Multiple workspaces can be useful for multi-document applications as well as serving as a method to cache workspaces. In the latter case, a series of workspaces can be created and pre-loaded in advance of their use to help avoid the delays that would otherwise be incurred if the information had to be sent at the time it was needed in the conference.

Workspaces may be created and deleted at any time with different attributes during a conference. In fact, it is important that applications create them as often as possible, else new arrivers to the conference will not be able to enter. Workspace attributes within a multi-workspace SI conference can be different from one workspace to another. No attributes shall exceed any limits negotiated during application enrollment.

8.3.1 Workspace structure**8.3.1.1 Plane stacking**

A workspace consists of a number of planes which determine how information on each plane occludes that of other planes. The number of planes in a workspace is defined at the time of creation. The number of planes may range from 1 to the maximum value negotiated for the conference using the Soft-Copy-Workspace-Max-Planes capability. An additional virtual plane used exclusively for pointers is present if the Soft-Copy-Pointing capability has been negotiated for the conference.

For a workspace with N planes, the planes are ordered from 0, being the farthest back, to N – 1, being the farthest forward. If pointing capability has been negotiated, the pointer plane forms a virtual plane N in front of all other planes.

When rendering an image to be displayed, the color value of each pixel (prior to any format conversion necessary for display) is determined by the following set of rules:

- If plane X contains a solid colored drawing element or pointer, or a bitmap image pixel, and all planes farther forward (from X + 1 through N), if any, contain transparent pixel values (no bitmap, solid colored drawing element, or solid colored pointer), then the resulting pixel shall be set to the color of this pixel on plane X.

- If the pixel value for all planes 0 through N are set to transparent (no bitmap, solid colored drawing element, or solid colored pointer), then the resulting pixel shall be set to the background color specified when the workspace was created. If the background color is specified as transparent, the color White shall be used instead.
- If plane X contains a drawing element specified with a line style of Highlight, and all planes farther forward (from X + 1 through N), if any, contain transparent pixel values (no bitmap, solid colored drawing element, or solid colored pointer), then a locally defined rule is applied to modify the pixel value that would result if these rules were applied only to planes 0 through X – 1 (X is equal to 0, the modification rule is applied to the background color). The locally defined modification rule should have the effect of modifying the image formed by rendering the layers below X in such a way that it appears that the resulting pixel value has been highlighted with a semi-transparent colored value of the color specified at this pixel in plane X. The specific algorithm for this rule is outside of the scope of this Recommendation.

8.3.1.2 Plane coordinate system

All planes on a workspace are of the same size with the origin of each co-located. Within each plane, pixels are indexed from the origin (0,0) which is defined to be the upper left corner of the workspace, to the size of the workspace (X – 1, Y – 1) which is defined to be the lower right corner of the workspace, where X and Y are the number of pixels in the workspace, in the horizontal and vertical dimensions, respectively, as specified when the workspace was created. The sizes X and Y shall be greater than or equal to one and less than or equal to the negotiated maximum values of Soft-Copy-Workspace-Max-Width and Soft-Copy-Workspace-Max-Height, respectively.

Positional references to a workspace plane are designated by specifying a Point. A Point is an ordered pair of Workspace Coordinates specifying the horizontal and vertical position in the workspace, respectively. The value of a Workspace Coordinate is defined to be within the range –21845, 43690. The use of negative values allows the ability to reference certain objects (e.g. the origin of a bitmap, or a control point in a drawing element) relative to a point above or to the left of the origin of the workspace plane.

All references to workspace size and plane coordinates assumes 1:1 pixel aspect ratio (square pixels) regardless of the aspect ratio of bitmaps that may be transmitted to the workspace.

8.3.1.3 Workspace views

For some or all of a workspace to be viewed by participants in a conference, at least one workspace view must be defined for the workspace. A workspace view is a region of the workspace to be viewed and associated characteristics to describe how that region should be viewed. A single workspace may be defined to have up to 256 views. Each view may cover distinct regions of the workspace, or some views may show common portions of the workspace.

Only active workspaces may be defined to have views – archived workspaces cannot. If views are defined for an archived workspace they shall be ignored.

Views do not exist separately from the workspace to which they are associated. If a workspace is deleted, all of its associated views are automatically deleted as well.

Figure 8-5 shows the correspondence between a set of workspace views and the workspace to which they are associated.

8.3.1.4 Drawing color palette

Colors used within a workspace for drawing or for defining the background color of the workspace are specified as belonging to the color space negotiated in the capability set reference by the Application Roster Instance for the workspace. If the Soft-Copy-Annotation capability is present in the negotiated capability set, but none of the Soft-Copy-Color capabilities, the color space is a palette of two colors. In addition to these two colors, drawing elements may be defined to be Transparent. The workspace background color shall not use Transparent as a valid color.

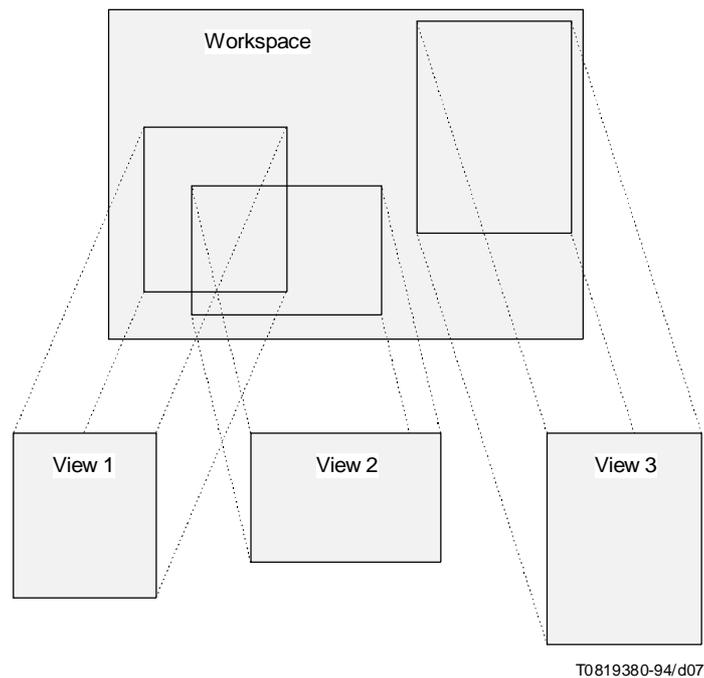


FIGURE 8-5/T.126
Workspace views

If a broader color space was negotiated for use, the corresponding color spaces may be used. Specifically, if the Soft-Copy-Color-16, Soft-Copy-Color-202, or Soft-Copy-Color-True capabilities are present in the negotiated capability set, then the 16-color palette, the 202 color palette, or a true-color specifier may be used, respectively. The three palettes and true color are defined as follows:

- *2 color*: This palette is mandatory given support for Soft-Copy-Annotation. It has two entries, Black and White. The entries in this palette are listed as the first two entries in Table 8-7.
- *16 color*: This palette is a super-set of the first. It may be used only if the Soft-Copy-Color-16 capability is present in the negotiated capability set. The entries in this palette are listed as the first 16 entries in Table 8-7.
- *202 color*: This palette is a super-set of the first two palettes. This palette may be used only if the Soft-Copy-Color-202 capability is present in the negotiated capability set. The entries in this palette are all of the entries listed in Table 8-7.
- *True color*: If the Soft-Copy-Color-True capability is present in the negotiated capability set, the color of drawing elements may optionally be specified as RGB color components, each with 8-bit precision.

A gamma of 1.8 is assumed for the workspace palette. Given that the color accuracy of the data types rendered to this palette is not critical, the color temperature and RGB primary values are left unspecified.

TABLE 8-7/T.126

Drawing Color palette

Index	R	G	B
0	0	0	0
1	255	255	255
2	128	0	0
3	0	128	0
4	128	128	0
5	0	0	128
6	128	0	128
7	0	128	128
8	192	192	192
9	128	128	128
10	255	0	0
11	0	255	0
12	255	255	0
13	0	0	255
14	255	0	255
15	0	255	255
16	192	220	192
17	166	202	240
18	255	251	240
19	160	160	164
20	0	0	0
21	8	8	8
22	16	16	16
23	25	25	25
24	33	33	33
25	41	41	41
26	49	49	49
27	58	58	58
28	66	66	66
29	74	74	74
30	82	82	82
31	90	90	90
32	99	99	99
33	107	107	107
34	115	115	115
35	123	123	123
36	132	132	132
37	140	140	140
38	148	148	148
39	156	156	156
40	165	165	165
41	173	173	173
42	181	181	181
43	189	189	189
44	197	197	197
45	206	206	206
46	214	214	214

TABLE 8-7/T.126 (continued)

Drawing Color palette

Index	R	G	B
47	222	222	222
48	230	230	230
49	239	239	239
50	247	247	247
51	255	255	255
52	0	0	0
53	0	0	63
54	0	0	127
55	0	0	191
56	0	0	255
57	63	0	0
58	63	0	63
59	63	0	127
60	63	0	191
61	63	0	255
62	127	0	0
63	127	0	63
64	127	0	127
65	127	0	191
66	127	0	255
67	191	0	0
68	191	0	63
69	191	0	127
70	191	0	191
71	191	0	255
72	255	0	0
73	255	0	63
74	255	0	127
75	255	0	191
76	255	0	255
77	0	51	0
78	0	51	63
79	0	51	127
80	0	51	191
81	0	51	255
82	63	51	0
83	63	51	63
84	63	51	127
85	63	51	191
86	63	51	255
87	127	51	0
88	127	51	63
89	127	51	127
90	127	51	191
91	127	51	255
92	191	51	0
93	191	51	63
94	191	51	127
95	191	51	191

TABLE 8-7/T.126 (continued)

Drawing Color palette

Index	R	G	B
96	191	51	255
97	255	51	0
98	255	51	63
99	255	51	127
100	255	51	191
101	255	51	255
102	0	102	0
103	0	102	63
104	0	102	127
105	0	102	191
106	0	102	255
107	63	102	0
108	63	102	63
109	63	102	127
110	63	102	191
111	63	102	255
112	127	102	0
113	127	102	63
114	127	102	127
115	127	102	191
116	127	102	255
117	191	102	0
118	191	102	63
119	191	102	127
120	191	102	191
121	191	102	255
122	255	102	0
123	255	102	63
124	255	102	127
125	255	102	191
126	255	102	255
127	0	153	0
128	0	153	63
129	0	153	127
130	0	153	191
131	0	153	255
132	63	153	0
133	63	153	63
134	63	153	127
135	63	153	191
136	63	153	255
137	127	153	0
138	127	153	63
139	127	153	127
140	127	153	191
141	127	153	255
142	191	153	0
143	191	153	63
144	191	153	127

TABLE 8-7/T.126 (continued)

Drawing Color palette

Index	R	G	B
145	191	153	191
146	191	153	255
147	255	153	0
148	255	153	63
149	255	153	127
150	255	153	191
151	255	153	255
152	0	204	0
153	0	204	63
154	0	204	127
155	0	204	191
156	0	204	255
157	63	204	0
158	63	204	63
159	63	204	127
160	63	204	191
161	63	204	255
162	127	204	0
163	127	204	63
164	127	204	127
165	127	204	191
166	127	204	255
167	191	204	0
168	191	204	63
169	191	204	127
170	191	204	191
171	191	204	255
172	255	204	0
173	255	204	63
174	255	204	127
175	255	204	191
176	255	204	255
177	0	255	0
178	0	255	63
179	0	255	127
180	0	255	191
181	0	255	255
182	63	255	0
183	63	255	63
184	63	255	127
185	63	255	191
186	63	255	255
187	127	255	0
188	127	255	63
189	127	255	127
190	127	255	191
191	127	255	255
192	191	255	0

TABLE 8-7/T.126 (concluded)

Drawing Color palette

Index	R	G	B
193	191	255	63
194	191	255	127
195	191	255	191
196	191	255	255
197	255	255	0
198	255	255	63
199	255	255	127
200	255	255	191
201	255	255	255

8.3.1.5 Workspace attributes

When a workspace is created, it is defined to have a number of attributes described in Table 8-8. These can be later edited via the WorkspaceEditPDU.

TABLE 8-8/T.126

Workspace attributes

Attribute	Default value	Description
preserve	FALSE	This is an optional parameter which indicates the preference to preserve this workspace if possible after an automatic transition of one of its views from the Focus state to the Foreground state (due to a view from another workspace being put in the Focus state).
backgroundColor	White	This attribute specifies the background Color of the workspace.
nonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation.

8.3.1.5.1 Workspace synchronization

Any workspace may be specified as being synchronized or unsynchronized. A synchronized workspace is one that guarantees that the workspace at each node in the conference contains the same information (once a period of time has passed after transmitting information to that workspace). This is done by requiring synchronized transmission for all SIPDUs using the MCS-Uniform-Send-Data service. For an unsynchronized workspace no such guarantee is made, and it is possible for the workspaces at different nodes to contain different information due to timing conditions in the communications network. In the unsynchronized case, MCS-Send-Data is used for some SIPDUs rather than MCS-Uniform-Send-Data resulting in a reduction in network traffic.

For a synchronized workspace, any SICE transmitted over the SI-CHANNEL shall make use of MCS-Uniform-Send-Data rather than MCS-Send-Data.

For an unsynchronized workspace, certain SIPDUs transmitted over the SI-CHANNEL shall use MCS-Send-Data, while others shall use MCS-Uniform-Send-Data.

The use of MCS-Send-Data vs. MCS-Uniform-Send-Data is summarized in Table 6-3.

8.3.1.5.2 Acceptance of keyboard and pointing device events

When a workspace is created, it may be specified as having the ability to accept keyboard events, pointing device events, or both by setting the corresponding flags in the WorkspaceCreatePDU. If the flag is set indicating acceptance of keyboard events, it indicates that RemoteKeyboardEventPDUs may be directed at that workspace. If the flag is set indicating acceptance of pointing device events, it indicates that RemotePointingEventPDUs may be directed at that workspace.

8.3.1.5.3 Preference to preserve the workspace

When a workspace is created, a flag may be set to indicate that it is preferred that the workspace be preserved, if storage permits, when a view of the workspace is automatically changed from the Focus state to the Foreground state due to a view from another workspace being placed in the Focus state. In this case, the workspace is not placed in the previously viewed workspace queue, but is placed in the preserved workspace store.

The preserve flag may be modified by sending a WorkspaceEditPDU specifying a new value of the preserve flag.

8.3.1.5.4 Background color

When a workspace is created, a background color shall be specified. The background color is used to determine the resulting color for the entire workspace if all planes (including the virtual pointer plane) is set to a transparent value, or is a drawing element with line style specified as highlight. In the former case, the background color is used directly, while in the latter, the background color may be modified by the highlighting rule.

The color space over which the background color may be chosen is determined by the negotiated palette for the corresponding workspace type (see 8.3.1.4). The baseline palette is the two-color palette. If a broader color space was negotiated for use, the corresponding color spaces may be used to set the background color. Specifically, if the Soft-Copy-Color-16, Soft-Copy-Color-202, or Soft-Copy-Color-True capabilities are present in the negotiated capability set, then the 16-color palette, the 202 color palette, or a true-color specifier may be used, respectively.

For any of the palettes which may be used, Transparent shall not be specified as the background color. If received, the color White shall be used instead.

The background color may be modified by sending a WorkspaceEditPDU specifying a new value of the background color.

8.3.1.6 Workspace plane parameters

When a workspace is created, each plane is defined to have a number of characteristics: whether or not the plane is editable, the intended usage of the plane, and whether the plane is protected for exclusive use by the creator. The latter two of these characteristics are attributes of the workspace plane which may be modified after the workspace has been created. See Table 8-9.

TABLE 8-9/T.126

Plane attributes

Attribute	Default value	Description
protection	creatorProtected = FALSE	This flag indicates any access restrictions for a plane. The creatorProtected sub-parameter, if set to TRUE, restricts all access to the plane to be by the creator of the workspace only. A value of FALSE imposes no restrictions.
nonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation.

8.3.1.6.1 Plane editability

When a workspace is created, each plane is designated as either editable or permanent as indicated by the editability flag in the WorkspaceCreatePDU. A plane shall be designated as permanent unless the Soft-Copy-Plane-Editing capability is included in the negotiated Application Capabilities List, in which case it may be designated as either permanent or editable.

For a plane designated as permanent, the effect of bitmap creation or drawing creation shall be to modify the image as a single-plane raster image. That is, each pixel in the plane has one and only one value that determines its color (or transparency) and carries no other state information. Bitmaps and drawing elements on a permanent plane cannot be edited or deleted. They can be modified only by overwriting their pixel locations with other bitmaps or drawing elements. The following PDUs shall not be directed toward a plane that is designated as permanent:

- BitmapDeletePDU;
- BitmapEditPDU;
- DrawingDeletePDU;
- DrawingEditPDU.

If a SICE receives these PDUs directed toward a permanent plane, they shall be ignored.

For a plane designated as editable, each SICE shall maintain a database of state information for each bitmap and drawing element created on that plane. For each object, this shall include the information defining the object and its position relative to the origin of the plane (either the full bitmap image and its origin, or the drawing type and control points), and the position of the object in the object-list for that plane (the z-order). The object-list determines the order of the objects within the plane, from front to back, used to determine which objects occlude other objects.

8.3.1.6.2 Plane usage

When a workspace is created, the intended use of each plane in the workspace shall be designated using the usage designator in the WorkspaceCreatePDU. The usage designator is a sequence of flags. The first flag indicates whether the plane may be used for annotation. The second indicates whether the plane may be used for images. At least one of these flags shall be set for every plane.

The annotation flag shall not be set unless the Soft-Copy-Annotation capability has been negotiated for the conference, and the image flag shall not be set unless the Soft-Copy-Image capability has been negotiated for the conference.

For a plane designated as allowing annotation, the plane may be used for drawing elements or annotation bitmaps. Drawing elements are those created using the DrawingCreatePDU, while annotation bitmaps are those created using the BitmapCreatePDU with the destinationAddress parameter set to softCopyAnnotationPlane.

For a plane designated as allowing image usage, the plane may be used for image bitmaps. Image bitmaps are those created using BitmapCreatePDU with the destinationAddress parameter set to softCopyImagePlane.

The usage designator of each plane determines the data priority used to transmit any SIPDU directed toward that plane. If the annotation plane usage flag is set for a plane (independent of whether or not the image plane usage flag is set), Medium data priority shall be used to send PDUs directed toward that plane. If the image plane usage flag is set, but the annotation plane usage flag is not set, Low data priority shall be used instead. This rule applies to the following SIPDUs:

- BitmapAbortPDU;
- BitmapCreatePDU;
- BitmapCreateContinuePDU;
- BitmapDeletePDU;
- BitmapEditPDU;
- WorkspacePlaneCopyPDU.

8.3.1.6.3 Plane protection

Each plane in a workspace may be designated as either unprotected or creator-protected. This designation is specified when the workspace is created as part of the plane-attributes parameter in the WorkspaceCreatePDU. Its value may also be modified for an existing workspace by using the WorkspaceEditPDU. Only the creator of the workspace may edit this attribute using the WorkspaceEditPDU. If a WorkspaceEditPDU is received from any node other than the node which created the specified workspace which indicates a change in the value of this attribute, this change shall be ignored (other attribute changes shall still be processed as normal).

If a workspace plane is designated as unprotected, any SICE in a conference may direct information (bitmaps or drawing elements) toward the plane.

If a workspace plane is designated as creator-protected, only the SICE which issued the WorkspaceCreatePDU to create this workspace may direct information toward the plane. Moreover, the creating SICE can choose to use MCS-Send-Data instead of MCS-Uniform-Send-Data for transactions directed to a workspace plane when the destination workspace plane is synchronized, if the creating SICE never intends to unprotect the target plane.

8.3.1.7 Workspace view parameters

When a workspace is created, it may be created along with one or more workspace views. Each view has several editable attributes. These attributes are listed in Table 8-10.

TABLE 8-10/T.126

View attributes

Attribute	Default value	Description
viewRegion	fullWorkspace	This attribute defines the portion of the workspace to be associated with this view. This may be set to either fullWorkspace, which indicates that the view corresponds to the complete workspace, or partialWorkspace. In the latter case, a rectangular region is specified, in workspace coordinates, indicating the desired portion of the workspace to view. The viewRegion shall not extend beyond the boundary of the workspace in any dimension.
viewState	focus	This attribute indicates the initial state of the view, either hidden, background, foreground, focus, or nonStandardState.
updatesEnabled	TRUE	This attribute specifies whether or not it is recommended that updates to the workspace associated with this view are displayed, or if the content of the view is instead fixed until updates are enabled. This attribute is to be interpreted as a recommendation on how to display the view rather than a requirement. This attribute applies independently from independent SICEs such that all SICEs disabling updates must re-enable updates for updates to be re-enabled for this view.
sourceDisplayIndicator	–	This attribute specifies the size and position of the view relative to the display at the sourcing terminal. This is to be interpreted as a recommendation to the receiving SICE to display the view in a similar manner. However, there is no requirement that the receiving SICE actually display the view in the indicated manner.
nonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation.

8.3.1.7.1 View region

Each workspace view is defined to cover a rectangular region of the workspace to which it corresponds. The viewRegion parameter defines the size and location of this region. Its default value is to view the entire workspace. Alternatively, it may be set to view a particular subregion of the workspace. The viewRegion shall not extend beyond the boundary of the corresponding workspace in any dimension.

8.3.1.7.2 View state

When a workspace view is created, the state of the view may be specified as having one of the following values:

- Hidden – Indicates this view should not be shown to the user.
- Background – Indicates that the display of this view is optional.
- Foreground – Indicates that the display of this view is desirable.
- Focus – Indicates that the display of this view is mandatory.

The state of the view may be modified by sending a WorkspaceEditPDU specifying a new value of the viewState attribute for this view.

Only one view among all views on all workspaces may simultaneously be set to the Focus state. Setting a view to this state forces any other view currently in the Focus state into the Foreground state.

An SICE may choose to send PDUs transmitted to workspaces in whose views are in different states at different throughput rates. For example, a SICE may send PDUs to workspaces whose views are Background or Hidden at a lower rate than PDUs to workspaces with Foreground views or the workspace with a view in the Focus state. This would minimize interference on information bound for the more important workspaces from information being sent in the background. The mechanism for controlling the throughput of PDUs intended for different workspaces is a local matter not covered by this Recommendation.

8.3.1.7.3 Update enable flag

This attribute specifies whether or not it is recommended that updates to the workspace associated with this view are displayed, or if the content of the view is instead fixed until updates are enabled. This attribute is to be interpreted as a recommendation on how to display the view rather than a requirement. If more than one SICE set this parameter to the Disabled state (FALSE), a SICE which makes use of this attribute shall keep track of all such requesting SICEs and should only re-enable updates to the view when all of these SICEs edited the attribute back to the TRUE state (at which time views are re-enabled). If a SICE leaves the conference after setting this attribute to FALSE all other SICEs should interpret this as an implied re-editing of this attribute to TRUE (from that SICE only). It is recommended that updates not be disabled for more than short periods of time (e.g. only during bursts of rapid activity on the workspace).

8.3.1.7.4 Source display indicator

The creator of a view may specify the size and position of the view relative to the display at the creator’s terminal. This is meant as a recommendation to receiving SICEs that this view be displayed in a similar manner. However, there is no requirement that the receiving terminal actually displays the view in this way. The sourceDisplayIndicator is composed of the set of parameters indicated in Table 8-11.

TABLE 8-11/T.126

Source display indicator

Parameter	Description
displayAspectRatio	The aspect ratio of the display device being indicated. This is defined to be the ratio of horizontal to vertical dimensions of the display area.
horizontalSizeRatio	This is the ratio of the horizontal size of the view to the horizontal display size.
horizontalPosition	This is the horizontal offset of the upper left hand corner of the view relative to the upper left hand corner of the display. This is given in normalized units relative to a horizontal display width of 1.0.
verticalPosition	This is the vertical offset of the upper left hand corner of the view relative to the upper left hand corner of the display. This is given in normalized units relative to a vertical display height of 1.0.

8.3.2 Creating a workspace

To create a workspace, an SICE shall send a WorkspaceCreatePDU to all SICES in the conference in the manner indicated in Table 6-3. The content of the WorkspaceCreatePDU is shown in Table 8-12. If the workspace is synchronized, it shall send this SIPDU three separate times on each of the three priority channels, High, Medium, and Low. This is done to avoid any possibility of data bound for this workspace on any priority channel being received prior to the WorkspaceCreatePDU itself. The first one received shall be acted on by the receiving SICE. All others shall be ignored. In the case of an unsynchronized workspace, only a single WorkspaceCreatePDU is issued as indicated in Table 6-3.

TABLE 8-12/T.126
WorkspaceCreatePDU

Parameter	Description
workspaceIdentifier	If this is an active (non-archived) workspace, this is a Handle allocated using the GCC-Registry-Allocate-Handle primitive. If this is an archived workspace, this consists of the Handle of the opened archive along with the entry name to be used to identify this workspace within the specified archive. Also included in this case is a modificationTime parameter indicating the current time. This shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set.
appRosterInstance	Application Roster Instance Number as returned in the most recently received GCC-Application-Roster-Report indication.
synchronized	Either Synchronized or Unsynchronized. See 8.3.1.5.1 for a description of this flag.
acceptKeyboardEvents	This flag expresses the ability of the workspace to accept keyboard events via the RemoteKeyboardEventPDU. See 8.3.1.5.2 for a description of this flag.
acceptPointingDeviceEvents	This flag expresses the ability of the workspace to accept pointing device events via the RemotePointingDeviceEventPDU. See 8.3.1.5.2 for a description of this flag.
workspaceSize	This parameter indicates the horizontal and vertical dimension of the workspace. This parameter is expressed in units of pixels. The horizontal component of this parameter may span the range (1..X) where X is the value of the Soft-Copy-Workspace-Max-Width parameter in the negotiated capability set. The vertical component of this parameter may span the range (1..Y) where Y is the value of the Soft-Copy-Workspace-Max-Height parameter in the negotiated capability set. The pixels used to define the workspace size are defined as having a 1:1 pixel aspect ratio (square pixels).
workspaceAttributes (Optional)	Workspace attributes are described in Table 8-8
planeParameters	Workspace plane parameters are described in Table 8-13
viewParameters (Optional)	Workspace view parameters are described in Table 8-14.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-13/T.126

Workspace plane parameters

Parameter	Description
editable	This flag indicates whether the plane shall be Permanent or Editable. This flag shall be set to Permanent unless the Soft-Copy-Workspace-Editing capability is present in the negotiated capability set.
usage	This flag indicates the intended usage for the plane: for either image bitmaps, annotation (drawing elements or annotation bitmaps), nonStandardPlaneUsage or a combination of these. This parameter may include annotation only if the Soft-Copy-Annotation capability is present in the negotiated capability set. This parameter may include image only if the Soft-Copy-Image capability is present in the negotiated capability set. This parameter may include a nonStandardPlaneUsage only if a corresponding non-standard capability is present in the negotiated capability set.
planeAttributes (Optional)	A set of attributes for a plane (see Table 8-9).

TABLE 8-14/T.126

Workspace view parameters

Parameter	Description
viewHandle	Unique Handle returned from GCC-Registry-Allocate-Handle exchange. This handle will be used to reference this view in all future SIPDUs.
viewAttributes (Optional)	A set of attributes for a view (see Table 8-10).

Any node that receives a WorkspaceCreatePDU on any of the three priority channels, it shall first examine the synchronized flag in the SIPDU to determine if the workspace is designated as synchronized or unsynchronized. If the workspace is unsynchronized, the SICE shall send a WorkspaceCreateAcknowledgePDU to the node that issued the WorkspaceCreatePDU. This is done as indicated in Table 6-3. The content of the WorkspaceCreateAcknowledgePDU is shown in Table 8-15. A SICE shall not issue any exchanges that require the use of a priority channel that a WorkspaceCreatePDU has not been received on in the case of synchronized workspaces. Only a single WorkspaceCreatePDU is issued in the case of unsynchronized workspaces.

Next, the SICE shall determine if it has received the Application Roster with the instance number associated with the Application Roster Instance parameter in the SIPDU. If it has not, it shall ignore the WorkspaceCreatePDU (and all WorkspaceCreatePDUs with the same Workspace Handle received later on other priority channels). If the designated Application Roster instance had been received (even if a later instance has subsequently been received), the WorkspaceCreatePDU is accepted and processed, creating the workspace with the characteristics designated in the parameters in the SIPDU.

As the SICE receives the remaining two WorkspaceCreatePDUs on each of the other two priority channels, it shall check that their Workspace Handle is the same as that of a WorkspaceCreatePDU previously received on another priority channel, and if so, it shall ignore it.

TABLE 8-15/T.126

WorkspaceCreateAcknowledgePDU

Parameter	Description
workspaceIdentifier	This specifies the workspace being acknowledged. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except that the modificationTime parameter shall not be included.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

If more than one workspace is being created at the same time by different SICEs, it is possible that the three WorkspaceCreatePDUs from the first SICE will be interleaved with the three from the second (i.e. the order that they are received on each of the three priority channels may be different). If both of the WorkspaceCreatePDUs specify views with Focus set as the viewState, since the order of receipt is indeterminate (normally the viewState of the view in the workspace first created would be changed to the Foreground state when the second workspace is created with a view in the Focus state), the value of the Workspace Handle is used to determine which view will remain in the Focus state, and which will be changed to the Foreground state. The view corresponding to the WorkspaceCreatePDU with the highest numbered Workspace Handle (interpreted as an unsigned integer) shall remain in the Focus state.

If the workspace is designated as unsynchronized, the originating SICE shall wait to receive WorkspaceCreateAcknowledgePDUs from each of the SICEs which are present and marked as Active in the instance of the Application Roster to which the workspace was referenced. If a new Application Roster is received in which some of the nodes in the original instance are no longer present, the SICE shall consider those nodes as having acknowledged and will no longer wait for a response from them. If the SICE receives an acknowledgment from any node which was not in the Application Roster instance to which the workspace was referenced, these acknowledgments shall be ignored. Once all acknowledgments have been received, the SICE creating the workspace shall issue a WorkspaceReadyPDU with the parameters set according to Table 6-3. This SICE can then perform all allowable exchanges to the new workspace. All other SICEs in the conference shall wait to receive the WorkspaceReadyPDU before performing any exchanges to the new workspace. However, information from other SICEs directed at the workspace can be received prior to reception of the WorkspaceReadyPDU and should be considered valid.

TABLE 8-16/T.126

WorkspaceReadyPDU

Parameter	Description
workspaceIdentifier	This specifies the workspace being identified as ready. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall not be included.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.3.3 Deleting a workspace

To delete a workspace, a SICE shall send a WorkspaceDeletePDU to all SICES in the conference in the manner indicated in Table 6-3. The content of the WorkspaceDeletePDU is shown in Table 8-17. For a synchronized workspace, it shall send this SIPDU three separate times on each of the three priority channels: High, Medium, and Low. This is done to allow a terminal which might wish to save the contents of a deleted workspace locally, to do so without ambiguity ensuring that all data bound for that workspace prior to the deletion has been consistently received at all nodes.

TABLE 8-17/T.126

WorkspaceDeletePDU

Parameter	Description
workspaceIdentifier	This specifies the workspace being deleted. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall correspond to the current time rather than the time of creation.
reason	Reason for workspace deletion. May be either userInitiated, insufficientStorage, or a nonStandardReason.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

Any node that receives a WorkspaceDeletePDU on any of the three priority channels shall first check the WorkspaceHandle to determine if the receiving SICE has a copy of this workspace. If so, it may either delete the workspace immediately, or, if it wishes to maintain a local copy of the deleted workspace in a form consistent with that which might be maintained on other workspaces, it may continue to apply information bound to this workspace from each of the priority channels up until the time that it receives the WorkspaceDeletePDU on that channel. Once it has received all three WorkspaceDeletePDUs, it has the workspace in its final form and may maintain it for local use. Note that this procedure only ensures consistency of the final form of the workspace if the workspace had been designated as synchronized.

8.3.4 Editing workspace, plane, and view attributes

At any time an SICE may modify the workspace, plane, and/or view attributes of a workspace by sending a WorkspaceEditPDU to all nodes in the conference in the manner indicated in Table 6-3. The content of the WorkspaceEditPDU is shown in Table 8-18. For a synchronized workspace, it shall send this SIPDU redundantly on all priority channels used for SIPDUs (High, Medium, and Low). In this case, on each priority channel, data received after reception of the WorkspaceEditPDU assumes the constraints of the new attribute set. Based upon the nature of the attribute changes, the application to the workspace of data received after the WorkspaceEditPDU on a given priority channel may need to be delayed until reception of the remaining copies of the WorkspaceEditPDU on all other priority channels.

TABLE 8-18/T.126

WorkspaceEditPDU

Parameter	Description
workspaceIdentifier	This specifies the workspace being edited. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall correspond to the current time rather than the time of creation.
attributeEdits (Optional)	Sequence of workspace attributes to be modified. Workspace attributes are described in Table 8-8.
planeEdits (Optional)	Sequence of planes for which plane attributes are to be modified. For each plane, the plane edit parameters are described in Table 8-19.
viewEdits (Optional)	List of views to create, edit, or delete. For each view, the view edit parameters are described in Table 8-19.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-19/T.126

Workspace plane edits

Parameter	Description
plane	This parameter is the number of the plane whose attributes are to be edited.
planeAttributes	Set of plane attributes to be modified. Plane attribute parameters are described in Table 8-9.

TABLE 8-20/T.126

Workspace view edits

Parameter	Description
viewHandle	Unique Handle indicating the view to be modified.
action	This parameter indicates the action to be taken on the view. Possible actions are createNewView, editView, deleteView, and nonStandardAction. If createNewView is indicated, a new view shall be created and associated with the workspace. An optional list of view attributes may be included to describe the characteristics of the new view. If editView is indicated, the characteristics of an existing view shall be modified. A list of view attributes indicates which attributes are to be changed. If deleteView is indicated, the indicated view shall be deleted. There are no additional parameters in this case. The nonStandardAction is only allowed if a corresponding non-standard capability has been negotiated. View attribute parameters are described in Table 8-10.

On receipt of a WorkspaceEditPDU, an SICE shall examine the Workspace Handle and determine whether it has a copy of this workspace. If so, the SICE shall apply the new workspace, plane, and view attributes indicated in the received WorkspaceEditPDU to the local copy of the workspace.

If the plane protection attribute is among those in the list of attributes to be modified, the receiving SICE shall check that the User ID of the source of the WorkspaceEditPDU is that same as that User ID of the SICE which created the workspace. If so, this attribute shall be modified as indicated in the PDU. If not, this attribute shall not be modified.

8.3.5 Copying workspace contents

If an SICE wishes to copy a rectangular region of one plane of a workspace to another rectangular region of the same plane, a different plane in the same workspace, or a different workspace, it may do so by sending a WorkspacePlaneCopyPDU to all peer SICES in the manner indicated in Table 6-3. The content of the WorkspacePlaneCopyPDU is shown in Table 8-21.

The WorkspacePlaneCopyPDU is allowed from all SICES if the specified destination plane is designated unprotected. If the designated plane is creator-protected, the WorkspacePlaneCopyPDU shall only be transmitted by the creator of the workspace. If an SICE receives a WorkspacePlaneCopyPDU with a destination plane specified which is protected from an SICE which is not allowed to modify this plane, the SICE shall ignore the received SIPDU.

The WorkspacePlaneCopyPDU is allowed only between planes which have identical usage designators and identical editability flags (either both editable or both permanent). If an SICE receives a WorkspacePlaneCopyPDU specifying a source and destination plane which do not meet these qualifications, the SICE shall ignore the received SIPDU.

8.3.6 Workspace refreshing for late arrivers

SICES may choose to implement facilities to retransmit active (non archived) workspaces which have been deleted in response to the reception of a GCC-Application-Roster-Report indication from the GCC provider indicating that one or more new SICES have joined the conference. The retransmitted data must conform to the constraints imposed by the new capabilities list contained within the GCC-Application-Roster-Report indication. To guarantee that only one SICE in a conference performs this function (should multiple SICES be capable of it), an SICE must attempt to establish itself as the refresher by first grabbing the SI-WORKSPACE-REFRESH-TOKEN. Upon successful acquisition, the SICE shall then broadcast a WorkspaceRefreshStatusPDU (see Table 8-22) with the parameters set to indicate that this SICE is the designated refresher. This indicates to other SICES that the refresh token has been grabbed.

If the refresher chooses to remove itself from this role, it shall first release the SI-WORKSPACE-REFRESH-TOKEN and then broadcast a WorkspaceRefreshStatusPDU (see Table 8-22) with the parameters set to indicate that this SICE is no longer the designated refresher. When other SICES which are capable of becoming the refresher receive this PDU, it is an indication to them that they may attempt to grab the SI-WORKSPACE-REFRESH-TOKEN to take over the role of refresher.

If the designated refresher has unexpectedly left the conference, other SICES will receive a GCC-Application-Roster-Report indication in which the designated refresher is no longer included. If this should occur, other SICES may consider this an indication that the SI-WORKSPACE-REFRESH-TOKEN is no longer grabbed and may then attempt to grab it themselves.

On receipt of a GCC-Application-Roster-Report indication which indicates that a new peer SICE has joined the conference, the refresher shall rebroadcast the WorkspaceRefreshStatusPDU (see Table 8-22) with the parameters set to indicate that this SICE is the designated refresher.

TABLE 8-21

WorkspacePlaneCopyPDU

Parameter	Description
sourceWorkspaceIdentifier	This specifies the source workspace from which a portion of a plane is to be copied. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall not be included.
sourcePlane	This parameter indicates the plane from which the source rectangle is to be copied. Its value may span the range (0..N – 1) where N is the number of planes in the source workspace.
sourceRegion	This parameter is a pair of points which define the rectangular region of the source plane from which information is to be copied. Each component of this parameter may span the range (–21845..43690). In the case of a permanent plane, the source information is the set of pixel values within the source rectangle. In the case of an editable plane, the source information is the set of drawing or bitmap objects which lie within the rectangle. If no boundingRectangle had been defined for the object, the object is considered within the source region if, in the case of a drawing element, all control points lie within the source region, and in the case of a bitmap, if the bitmap is fully contained within the source region. If the boundingRectangle parameter had been specified for the object, the object is considered within the source region if the bounding rectangle lies completely within the source region.
destinationWorkspaceIdentifier	This specifies the destination workspace to which a portion of a plane is to be copied. This may be either an active or archived workspace, although this shall not be an archived workspace if the Archive-Supported capability is not present in the negotiated capability set. In the case of an active workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace. In the case of an archived workspace, this shall equal the value of the workspaceIdentifier in the WorkspaceCreatePDU of the corresponding workspace except the modificationTime parameter shall correspond to the current time rather than the time of creation.
destinationPlane	This parameter indicates the plane to which the source rectangle is to be copied. Its value may span the range (0..N – 1) where N is the number of planes in the destination workspace. The destination plane shall have the same editability flag setting and the same usage designator setting as the source plane.
destinationRegion	This parameter is a destination rectangle to specify the location that the region is copied in the destination plane. If the Soft-Copy-Scaling capability is not present in the negotiated capability set, the size of this rectangle shall exactly equal the size of the source rectangle. Otherwise, the source information is scaled by the ratio of the destination dimensions to the source dimensions prior to placing the information in the destination workspace. For a permanent workspace, all of the pixels in the destination region are overwritten by values from the source region. For an editable plane, the objects from the source region are overlaid onto the destination plane with a z-order above any existing objects.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-22/T.126

WorkspaceRefreshStatusPDU

Parameter	Description
refreshStatus	A flag which if TRUE, indicates that the SICE sourcing this PDU is functioning as the conference-wide refresher. If FALSE, this indicates that the SICE sourcing this PDU has ceased to function as the conference-wide refresher.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.3.7 The effect of changes to the application roster

If an SICE receives a GCC-Application-Roster-Report indication from the GCC provider, it shall examine the new application roster. If the application roster indicates that no new SICES have enrolled since the last roster instance, the SICE shall examine the received Application Capabilities List, and apply the rules indicated in Table 8-5 to generate the newly negotiated capability set. If either the capability set has not changed, or the capability set has been expanded (i.e. new capabilities have been added to the negotiated list, MIN capabilities have increased in their negotiated value, or MAX capabilities have decreased in their negotiated value, but the opposite has not occurred for any capability), no change is made to the current workspaces, but the next workspace that is created may make use of the new capability set, and the Application Roster Instance parameter shall be set to the new instance value indicated in the GCC-Application-Roster-Report indication.

If, on the other hand, one or more new SICES have enrolled in the conference, or, due to a node re-enrolling, the negotiated capability set has been contracted (i.e. capabilities have been removed, MIN capabilities have decreased their negotiated value, or MAX capabilities have increased their negotiated value), when the next new workspace is created by any peer SICE, all existing workspaces are automatically deleted. The newly created workspace shall make use of the new capability set, and the Application Roster Instance parameter shall be set to the new instance value indicated in the GCC-Application-Roster-Report indication. If there is an SICE which has indicated itself to be a workspace refresher, that SICE may retransmit some or all of the previously held workspaces using the new capability set (see 8.3.6).

If more than one GCC-Application-Roster-Report indication is received prior to creation of a new workspace, these rules apply cumulatively until a new workspace is created. If the condition arose on any of these indications which would have resulted in deletion of the existing set of workspaces, when a new workspace is finally created, the existing workspaces are deleted even if this was not the result of the last GCC-Application-Roster-Report indication received. In this case, the Application Roster Instance parameter shall be set to the new instance value indicated in the most recently received GCC-Application-Roster-Report indication.

In all cases, workspaces created using a previous Application Roster Instance retain the negotiated capability set of that instance in determining what operations are allowed for that workspace. All references in this Recommendation to the negotiated capability set in describing operations on workspaces imply the capability set for the Application Roster Instance specified in the WorkspaceCreatePDU, which is not necessarily the same as the current negotiated capability set.

When a new GCC-Application-Roster-Report is received which contains new SICES in the conference, after the next creation of a workspace, all other workspaces are automatically deleted. A workspace refresher may recreate these workspaces once deleted (see 8.3.6).

8.3.8 Workspace caching

When an active (non-archived) workspace is created, the total storage required to contain the existing workspaces plus the newly created workspace increases. At each SICE, this increase may or may not exceed the total storage capacity. If the storage capacity is exceeded, each SICE must determine whether to either create the new workspace and delete one or more existing workspaces, or to fail to create the new workspace. In the former case, the SICE shall follow a strict set of rules to determine which previously existing workspaces to delete. This is to attempt to avoid the case that multiple nodes in the conference choose different workspaces to delete, resulting in unnecessary loss of workspaces.

The rules that govern this process depend on several aspects of the workspace to be created as well as the other workspaces currently in existence. One such aspect is the viewState of views associated with the workspace. If any view associated with a workspace is in the focus state, this workspace is referred to as the Focus workspace for the purpose of defining these rules. If no view associated with a workspace is in the focus state, this workspace is referred to as a non-Focus workspace.

If an SICE is capable of storing more than just the Focus workspace, the non-Focus workspaces shall be divided into one of two categories: the previously viewed workspace queue, and the preserved workspace store. Workspaces automatically changed from Focus to non-Focus (due to another workspace becoming the Focus workspace) with their Preserve flag set to FALSE are placed at the top of the previously viewed workspace queue. Workspaces automatically changed from Focus to non-Focus with their Preserve flag set to TRUE, or newly created non-Focus workspaces are placed in the preserved workspace store.

When a new active (non-archived) workspace is created, if it is necessary for the SICE to delete one or more workspaces to make room, it shall delete them from the bottom of the previously viewed image queue. This is done by issuing a WorkspaceDeletePDU for each deleted workspace as described in 8.3.3. In this case, the reason flag is set to insufficientStorage.

In the case of a newly created non-Focus workspace, if an SICE does not have sufficient storage to create the workspace without deletion of other workspaces, but it no longer has any workspaces in the previously viewed workspace queue (or if there are not enough to free up sufficient space for the new workspace), it shall not create the new workspace, but instead shall send a WorkspaceDeletePDU indicating deletion of the newly created workspace. Again, the reason flag is set to insufficientStorage.

In the case of a newly created Focus workspace, if an SICE does not have sufficient storage to create the workspace without deletion of another workspace, but it no longer has any workspaces in the previously viewed workspace queue (or if there are not enough to free up sufficient space for the new workspace), it shall delete the previous Focus workspace, if there was one, regardless of whether the Preserve flag was set or not. If this still does not free up sufficient room for the new workspace, a workspace in the preserved workspace store may be deleted, although an SICE should be designed to avoid this situation by allowing room for a Focus workspace of maximum negotiated size.

If the attributes of a workspace already in the previously viewed workspace queue is edited by its creator using the WorkspaceEditPDU to set the Preserve flag to TRUE, it shall then be considered part of the preserved workspace store and no longer part of the previously viewed workspace queue.

8.4 Bitmaps

Bitmap exchanges are used for several functions in this Recommendation. These include exchanging text (by rendering the strings locally into a bitmap), pointing, annotating with unsupported drawing elements and exchanging photographic and document images.

The value of the destinationAddress parameter of the BitmapCreatePDU indicates the type of bitmap being created. The allowable values for many of the BitmapCreatePDU parameters differ based on the type of bitmap being created. Also note that image bitmap have a specified checkpointing option, which is outlined in 8.4.1. This may be used if the creator desires synchronized incremental display of the bitmap at all SICEs during reception.

8.4.1 Creating bitmaps

Before a bitmap create exchange is initiated, a SICE shall grab the SI-BITMAP-CREATE-TOKEN if the Soft-Copy-Bitmap-No-Token-Protection capability has not been successfully negotiated. Upon completion of the exchange, the token shall be freed if it was grabbed. This token is used to prevent multiple bitmap create exchanges from happening simultaneously in a conference. Note that if the destination workspace is unsynchronized, then some overlap may occur between two sequential bitmap create exchanges due to the mandated use of MCS-SEND-DATA instead of MCS-UNIFORM-SEND-DATA for unsynchronized workspaces.

To initiate the bitmap create exchange within the conference, the SICE shall issue a BitmapCreatePDU to all SICES in the conference by the manner specified in 6.2 with the parameters set according to Table 8-23.

If all of the encoded data fits into one PDU, the moreToFollow flag shall be set to FALSE signalling the completion of the exchange to the receivers, else it shall be set to TRUE and subsequent BitmapCreateContinuePDUs shall be sent to complete the broadcast of the bitmap to the conference. All receivers should guarantee that they adhere to the acknowledgment style specified in the BitmapCreatePDU even if the complete exchange fits into one PDU. It is strongly recommended that applications adjust the maximum amount of data they send in one PDU to avoid latency problems.

TABLE 8-23/T.126

BitmapCreatePDU parameters

Parameter	Description																												
bitmapHandle	Unique Handle returned from GCC-Registry-Allocate-Handle exchange. This handle will be used to reference this bitmap in all future SIPDUs.																												
destinationAddress	<p>This parameter can take several forms depending on the value of the bitmap type and if the destination is intended to be a hard-copy device or soft-copy workspace.</p> <table border="1"> <thead> <tr> <th>Destination type</th> <th>Bitmap type</th> <th>Parameter value</th> <th>Sub-parameters</th> </tr> </thead> <tbody> <tr> <td>hard copy</td> <td>image</td> <td>hardCopyDevice</td> <td>NULL</td> </tr> <tr> <td>hard copy</td> <td>annotation</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td>hard copy</td> <td>pointer</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td>soft copy</td> <td>image</td> <td>softCopyImagePlane</td> <td>destination workspace handle plane ID</td> </tr> <tr> <td>soft copy</td> <td>annotation</td> <td>softCopyAnnotationPlane</td> <td>destination workspace handle plane ID</td> </tr> <tr> <td>soft copy</td> <td>pointer</td> <td>softCopyPointerPlane</td> <td>destination workspace handle</td> </tr> </tbody> </table> <p>The ability to create a bitmap of a certain type and the constraints on the bitmap to be created depend on the negotiated capabilities corresponding to the indicated choice of this parameter value.</p> <p>In the case of a soft-copy image bitmap, the destination plane must have its plane usage set to allow image information. In the case of a soft-copy annotation bitmap, the destination plane must have its plane usage set to allow annotation.</p>	Destination type	Bitmap type	Parameter value	Sub-parameters	hard copy	image	hardCopyDevice	NULL	hard copy	annotation	N/A	N/A	hard copy	pointer	N/A	N/A	soft copy	image	softCopyImagePlane	destination workspace handle plane ID	soft copy	annotation	softCopyAnnotationPlane	destination workspace handle plane ID	soft copy	pointer	softCopyPointerPlane	destination workspace handle
Destination type	Bitmap type	Parameter value	Sub-parameters																										
hard copy	image	hardCopyDevice	NULL																										
hard copy	annotation	N/A	N/A																										
hard copy	pointer	N/A	N/A																										
soft copy	image	softCopyImagePlane	destination workspace handle plane ID																										
soft copy	annotation	softCopyAnnotationPlane	destination workspace handle plane ID																										
soft copy	pointer	softCopyPointerPlane	destination workspace handle																										
attributes (Optional)	Bitmap attributes controlling certain appearance characteristics. See Table 8-24 for details.																												
anchorPoint (Optional)	This parameter is only applicable to bitmaps positioned within workspaces (not bound for a hard-copy device). It specifies the position of the upper left corner of the displayable region of the bitmap (as specified by the bitmapRegionOfInterest) within the destination workspace. This parameter may only be used for softcopy bitmaps (it will be ignored for hardcopy bitmaps). If this parameter is not present for a soft-copy bitmap the anchor point is assumed to be (0,0).																												

TABLE 8-23/T.126 (continued)

BitmapCreatePDU parameters

Parameter	Description
bitmapSize	<p>This parameter specifies the horizontal and vertical size of the bitmap in pixels. Note that the pixel aspect ratio of the bitmap may not be square although the workspace coordinate system assumes a square pixel reference grid. In this case, the number of pixels the bitmap spans in the workspace will be different from the number of pixels in the bitmap itself. In the case of a bitmap format which includes more than one color component, this parameter represents the size of the largest component. The bitmapSize parameter includes a width and height sub-parameter. The allowable range of these parameters is dependent on the destinationAddress selected:</p> <p>Width: destinationAddress = hardCopyDevice: (1..Hard-Copy-Image-Bitmap-Max-Width)</p> <p>destinationAddress = softCopyImagePlane: (1..Soft-Copy-Image-Bitmap-Max-Width)</p> <p>destinationAddress = softCopyAnnotationPlane: (1..Soft-Copy-Annotation-Bitmap-Max-Width)</p> <p>destinationAddress = softCopyPointerPlane: (1..Soft-Copy-Pointing-Bitmap-Max-Width)</p> <p>Height: destinationAddress = hardCopyDevice: (1..Hard-Copy-Image-Bitmap-Max-Height)</p> <p>destinationAddress = softCopyImagePlane: (1..Soft-Copy-Image-Bitmap-Max-Height)</p> <p>destinationAddress = softCopyAnnotationPlane: (1..Soft-Copy-Annotation-Bitmap-Max-Height)</p> <p>destinationAddress = softCopyPointerPlane: (1..Soft-Copy-Pointing-Bitmap-Max-Height)</p> <p>The bitmapSize parameter shall reflect the actual dimensions of the bitmap encoded in the bitmapData parameter of this PDU (along with any additional BitmapCreateContinuePDUs). If a bitmap is received for which a bitmap dimension is less than that indicated by this parameter, the bitmap shall be padded to the indicated size. In the case of a hard-copy destination, it shall be padded with white, in the case of a soft-copy destination, it shall be padded with transparent. If a bitmap is received for which a bitmap dimension is greater than that indicated by this parameter, the bitmap shall be truncated to the indicated size. In any case, the bitmap size indicated by this parameter (if the optional bitmapRegionOfInterest or boundingRectangle parameters are not included) shall determine its size as used in determining whether the bitmap is contained within a source rectangle for the purpose of copying a portion of a workspace plane.</p>
bitmapRegionOfInterest (Optional)	<p>This optional parameter selects the subregion within the associated bitmap that is to be displayed. Note that if the destination is an editable workspace plane, a SICE is required to store the entire transmitted bitmap. The default values for the upper left and lower right offsets if not supplied are (0,0) for the upper left and (bitmap width – 1, bitmap height – 1) for the lower right. If this parameter is used, the anchorPoint and scaling parameters apply to the region of interest rather than to the original bitmap borders.</p>
pixelAspectRatio	<p>Describes the pixel aspect ratio of the bitmap. Different values are allowable depending on the bitmap destination, the bitmap format, and the negotiated capability set. These constraints on pixelAspectRatio are listed in Table 8-5. The range of possible values is shown in Table 8-25.</p>
scaling (Optional)	<p>This optional parameter is only allowed if the Soft-Copy-Scaling is part of the negotiated capability set. This parameter if present indicates the offset from the anchor point, in Workspace coordinates, of the lower right corner of the bitmap within the workspace. This parameter may only be used for soft-copy bitmaps (it will be ignored for hard-copy bitmaps). If this parameter is not present for a soft-copy bitmap the lower right hand corner is determined from the bitmapSize, bitmapRegionOfInterest (if present), and the pixelAspectRatio.</p>

TABLE 8-23/T.126 (continued)

BitmapCreatePDU parameters

Parameter	Description
boundingRectangle (Optional)	<p>This is a parameter which defines the extent of the bitmap relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this bitmap is to be considered within the copy region. If the bounding rectangle is fully contained within the copy region, this bitmap shall be included in the region copied regardless of the actual region spanned by the bitmap. If this parameter is not specified, the actual extent of the bitmap (after any scaling) determines whether the bitmap is enclosed by the copy region. This parameter is mandatory if a non-one-to-one pixel aspect ratio is specified for the bitmap on an editable workspace plane. This is to avoid ambiguities in determining the extent of the bitmap with respect to workspace coordinates.</p>
checkpoints (Optional)	<p>This optional parameter, when present, specifies that checkpointing is in effect for this exchange and correspondingly specifies the set of Token IDs that are to be used for checkpointing. Each Token ID is used to track the status of a portion of the bitmap in transit at all the receivers by the sender. If this parameter is present, receivers should immediately inhibit all the tokens in this set upon receipt of this PDU.</p>
bitmapFormatHeader	<p>Specifies the algorithm used to encode the bitmap data and associated parameters. Note that certain values are applicable only to certain bitmap destinations. Note also that some of the formats listed shall not be used unless the corresponding capability has been negotiated. See Table 8-5 for required capabilities. See 8.4.4 for the details of each specific encoding format supported by this Recommendation.</p> <p>If destinationAddress = (hardCopyDevice)</p> <p>Choice of:</p> <ol style="list-style-type: none"> 1) bitmapHeaderUncompressed; 2) bitmapHeaderT4; 3) bitmapHeaderT6; 4) bitmapHeaderT82; 5) bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list). <p>If destinationAddress = softCopyImagePlane</p> <p>Choice of:</p> <ol style="list-style-type: none"> 1) bitmapHeaderUncompressed; 2) bitmapHeaderT81; 3) bitmapHeaderT82; 4) bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list). <p>If destinationAddress = (softCopyAnnotationPlane)</p> <p>Choice of:</p> <ol style="list-style-type: none"> 1) bitmapHeaderUncompressed; 2) bitmapHeaderT82; 3) bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list). <p>If destinationAddress = softCopyPointerPlane</p> <p>Choice of:</p> <ol style="list-style-type: none"> 1) bitmapHeaderUncompressed; 2) bitmapHeaderT82; 3) bitmapHeaderNonStandard (Only valid if the capability corresponding to the specific bitmapHeaderNonStandard is in the negotiated capability list).

TABLE 8-23/T.126 (concluded)

BitmapCreatePDU parameters

Parameter	Description
bitmapData (optional)	Encoded pixel data representing the bitmap (See Table 8-26).
moreToFollow	TRUE signals that more BitmapCreateContinuePDUs will follow, carrying additional data to complete the bitmap transaction. FALSE signals that the transaction is complete with this PDU and no more will follow.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-24/T.126

Bitmap attributes

Attribute	Default value	Description
viewState	unselected	This attribute is only applicable to bitmaps positioned within workspaces (not bound for a hard-copy device) that are resident in editable workspace planes or are of type pointer. It can take on one of the following values: <ul style="list-style-type: none"> unselected: Bitmap should be displayed normally. selected: Bitmap should be displayed with some unspecified visual highlighting to indicate that the bitmap is selected and that edit or delete operations to the bitmap may be imminent (not applicable to pointer bitmaps). hidden: Bitmap should be removed from view but remain in the local database. nonStandardViewState
zOrder	front	This attribute is only applicable to bitmaps positioned within workspaces (not bound for a hard-copy device) that are resident in an editable workspace plane. It specifies the position in the object stack within the plane that the bitmap is to initially be at. Only one object within an editable plane can be at the front, therefore the operation of setting this attribute to front for a bitmap moves the previous front object behind this one in the stacking order. Only one object within an editable plane can be at the back, therefore the operation of setting this attribute to back for a bitmap moves the previous back object in front of this one in the stacking order.
nonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation. An arbitrary number of different nonStandardAttributes may be included in the attributes list.

TABLE 8-25/T.126

Pixel aspect ratios

Pixel aspect ratio	Description
square	The horizontal extent of a pixel as compared to its vertical extent is given by the ratio, 1:1.
cif	The horizontal extent of a pixel as compared to its vertical extent is given by the ratio, 12:11 (horizontal : vertical pixel size).
fax1	The horizontal extent of a pixel as compared to its vertical extent is given by the ratio, 385:800 (horizontal : vertical pixel size).
fax2	The horizontal extent of a pixel as compared to its vertical extent is given by the ratio, 770:800 (horizontal : vertical pixel size).
general	SICEs wishing to use pixel aspect ratios not specifically defined by this Recommendation can specify this type and its associated sub-parameters if the appropriate capability has been negotiated. <code>Soft-Copy-Image-Bitmap-Any-Aspect-Ratio</code> must be successfully negotiated for soft-copy image bitmaps and <code>Hard-Copy-Image-Bitmap-Any-Aspect-Ratio</code> must be negotiated for hard-copy bitmap exchanges. This method of aspect ratio specification may also be used for specifying non-standard aspect ratios given a successfully negotiated non-standard capability allowing this representation.
nonStandardAspectRatio	SICEs wishing to use pixel aspect ratios not specifically defined by this Recommendation can specify a successfully negotiated non-standard capability to represent the desired pixel aspect ratio.

TABLE 8-26/T.126

bitmapData parameters

Parameter	Description
dataCheckpoint (Optional)	This list of token IDs can only be present if the checkpoints parameter is present in the <code>BitmapCreatePDU</code> and if the data parameter carries in it encoded bitmap information that, when displayed, constitutes passing one or more checkpoints as determined by the transmitter. The receiver shall uninhibit these tokens when the bitmap data carried by the corresponding data parameter is ready for display locally.
padbits (Optional)	If present, it signals the receivers to ignore a specified number of bits at the end of the data field. Allowable values are (1..256).
data	This field contains the portion of encoded bitmap bitstream contained within one <code>BitmapCreatePDU</code> or <code>BitmapCreateContinuePDU</code> . The format of this data is dictated by the sub-parameters contained within the <code>bitmapFormatHeader</code> field of the start parameter in addition to the coding standard specification that may apply. See 8.4.4 for details on the format of this field for each bitmap format. The allowable size in octets is (1..8192). Note that transmitters should limit the size of this field to an appropriate value so as to minimize latency problems that can occur with large PDUs.

In order to facilitate the synchronization of the display of bitmaps at all sites in a conference, the transmitter of a bitmap can mandate that checkpointing be used for the exchange. This process is initiated by supplying the optional checkpoints parameter in the BitmapCreatePDU with the value set to the sequence of Token IDs to be used to mark each checkpoint. All receivers shall inhibit all tokens in the sequence immediately upon receipt in the order that they appear in the sequence. Each transmitter is responsible for allocating a set of dynamic tokens for this purpose some time before their use. The number of dynamic tokens allocated is determined by the individual applications wishing to originate checkpointed bitmap exchanges and should be proportional to the product of the number of checkpoints desired per bitmap and the maximum number of simultaneously sourced bitmap create exchanges the application allows. Each token in the set shall be allocated by issuing a GCC-Registry-Assign-Token request to the local GCC provider. The parameters used in this request are as shown in Table 8-27. If the Result parameter returned in the GCC-Registry-Assign-Token confirm is “successful,” the Token ID contained in the returned confirm primitive can be used as a checkpointing token by the application.

TABLE 8-27/T.126

Parameters for GCC-Registry-Assign-Token-request used to allocate checkpoint tokens

Parameter	Contents
ConferenceID	Provided by GCC-Application-Permission-To-Enroll indication
RegistryKey	Registry keys are composed of an abstract type consisting of the ApplicationKey for this instance of the application (the same key used in the application enrollment) with a unique ID string that is T.50 compliant. This string is formed by concatenating the one octet string “K” with a string formed by converting a unique handle allocated using GCC-Registry-Allocate-Handle primitive to a decimal string of at least one octet in length with no leading zeros.

For bitmap create exchanges involving bitmap data streams that exceed the maximum number of octets allowable per data parameter or that are purposely broken into smaller payloads for latency minimization, multiple PDUs must be used. BitmapCreateContinuePDUs shall be issued in a manner described in 6.2 until all encoded pixel data has been broadcast to the conference. The parameters of this PDU are described in Table 8-28. These PDUs shall be issued in order such that concatenation of their data payload at all receivers less pad bits forms an exact copy of the bitmap data stream at the transmitter. The last such PDU shall have the moreToFollow flag set to FALSE to signal the end of the transaction.

TABLE 8-28/T.126

BitmapCreateContinuePDU parameters

Parameter	Description
bitmapHandle	This parameter must be specified with the same value used in the BitmapCreatePDU for this exchange.
bitmapData	Encoded pixel data representing the bitmap (see Table 8-26).
moreToFollow	A value of TRUE signals that more BitmapCreateContinuePDUs will follow carrying additional data to complete the bitmap transaction. A value of FALSE signals that the transaction is complete with this PDU and no more will follow.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

If present, a checkpoint token shall be uninhibited by each receiver as its corresponding bitmapData payload is made ready for display locally. Note that a checkpoint token (delivered in the optional checkpoint parameter of the BitmapCreatePDU and BitmapCreateContinuePDU) is considered to correspond to a bitmapData payload if it was delivered via the same PDU.

During a checkpointed transmission, the SICE sourcing the bitmap shall poll the status of the tokens specified in the checkpoints parameter of the BitmapCreatePDU at an unspecified rate by testing the checkpoint token that has been outstanding the longest and has not yet been uninhibited by all receiving SICEs. Upon determining that one or more checkpoint tokens are free, the source SICE shall issue a BitmapCheckpointPDU with the parameters set according to Table 8-29 to indicate to all receiving SICEs that the bitmap data corresponding to the tokens specified can be displayed.

Note that a race condition exists with this mechanism in that a transmitting SICE may test a checkpoint before any receiving SICEs in the conference have received the BitmapCreatePDU. In this case, the checkpointing mechanism fails safe in that the transmitting SICE broadcasts a BitmapCheckpointPDU for the checkpoint prematurely. To minimize the probability of this occurring, the transmitting SICE can wait for some period of time before beginning to poll the first checkpoint token. A simple way of determining this timeout period, in the case of a synchronized destination workspace, is to postpone polling of a checkpoint token until the transmitting SICE receives the BitmapCreatePDU or BitmapCreateContinuePDU with which the checkpoint token was issued back via a MCS-UNIFORM-SEND-DATA indication.

Also in the case of a synchronized destination workspace, the transmitting SICE should not inhibit or uninhibit its own checkpoint tokens when they are returned in the appropriate PDUs via an MCS-UNIFORM-SEND-DATA indication.

TABLE 8-29/T.126

BitmapCheckpointPDU parameters

Parameter	Description
bitmapHandle	This parameter references the bitmap and shall be specified with the same value used in the BitmapCreatePDU for this exchange.
passedCheckpoint	This parameter is a list of tokens corresponding to passed checkpoints by the transmitting SICE. Note that a checkpoint is considered passed if all the receiving terminals have uninhibited it. Receiving terminals can display the portion of the bitmap in transit corresponding to the passed checkpoints specified by this parameter.
percentComplete	This parameter's value reflects the cumulative percentage of the bitmap exchange that is complete inclusive of the bitmap information inferred by the passedCheckpoint parameter. The value range is (1..100).
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

Should a receiving SICE wish to cause the transmitting SICE to abort the bitmap create exchange in progress, it shall issue a BitmapAbortPDU with the parameters set to values specified in Table 8-30 to the UserID channel of the transmitting SICE. Upon receipt of the BitmapAbortPDU, the transmitting SICE may choose to not issue any additional BitmapCreateContinuePDUs and broadcast a BitmapAbortPDU to all SICEs in the conference via the SI-CHANNEL to signal the premature end of the exchange. Because SICEs may have begun to display the bitmap locally before the transaction was aborted, the cleanup rules outlined in Table 8-31 should be adhered to in order to maintain display consistency within the conference. If the bitmap is destined for a workspace whose resource is reallocated via a subsequent workspace create, a bitmap abort request is implied and the BitmapAbortPDU shall be issued in the manner described above.

TABLE 8-30/T.126

BitmapAbortPDU parameters

Parameter	Description
bitmapHandle	This parameter references the bitmap and shall be specified with the same value used in the BitmapCreatePDU for this exchange.
userID (Optional)	This parameter can be optionally supplied. It is used to indicate the MCS User ID of the SICE that requested the termination of the bitmap create exchange.
reason (Optional)	This parameter is set to one of unspecified, outOfPaper, noResources or a nonStandardReason. It is used to indicate the reason for the abort request to the transmitter.
message (Optional)	This parameter can be optionally supplied. It is a Unicode string suitable for display to a user.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-31/T.126

Cleanup rules for aborting bitmap creation

Destination	Abort Cleanup Rule
Soft-copy editable workspace	Delete the bitmap from the display.
Soft-copy permanent workspace or Hard-copy device	Commit all bitmap data received from transmitter to the display up to the last decodable full scan line or block of scan lines.

Normal termination of the bitmap create exchange is signalled by the transmitting SICE by setting the moreToFollow parameter to FALSE in either the BitmapCreatePDU in the case of a single PDU exchange or in the BitmapCreateContinuePDU in the case of a multi-PDU exchange. All receiving SICES should be sure to have uninhibited any remaining checkpoint tokens that have not already been uninhibited during the exchange.

8.4.2 Deleting Bitmaps

To delete bitmaps that are members of an editable plane or that are pointers, a SICE shall issue a BitmapDeletePDU whose parameters are described in Table 8-32. A SICE can only delete a pointer if it was the creator.

TABLE 8-32/T.126

BitmapDeletePDU parameters

Parameter	Description
bitmapHandle	This parameter references the bitmap to be deleted.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.4.3 Editing bitmaps

To edit the attributes of bitmaps that are members of an editable plane or that are pointers, a SICE shall issue a BitmapEditPDU whose parameters are described in Table 8-33.

TABLE 8-33/T.126

BitmapEditPDU parameters

Parameter	Description
bitmapHandle	This parameter references the bitmap to be edited.
attributeEdits (Optional)	List of bitmap attributes and associated values to be modified. The allowable values for list members are described in Table 8-24.
anchorPointEdit (Optional)	This parameter specifies the position of the upper left corner of the displayable region of the bitmap (as specified by the bitmapRegionOfInterest) within the destination workspace. If this parameter is not present the anchor point is not modified.
bitmapRegionOfInterestEdit (Optional)	This optional parameter selects the subregion within the associated bitmap that is to be displayed. Note that a SICE is required to store the entire transmitted bitmap if the destination plane is editable and the destination is not a hard-copy device. If this parameter is not present, the region of interest is not modified.
scalingEdit (Optional)	This optional parameter is only allowed if the Soft-Copy-Scaling is part of the negotiated capability set. This parameter if present indicates the offset from the anchor point, in Workspace coordinates, of the lower right corner of the bitmap within the workspace. If this parameter is not present the scaling is not modified.
boundingRectangle (Optional)	This is a parameter which defines the extent of the bitmap relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this bitmap is to be considered within the copy region.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.4.4 Bitmap color definition

Depending on the bitmap format, bitmaps may be encoded using either palettized color maps or directly mapped color spaces.

For the directly mapped color spaces, Greyscale, RGB, CIElab or YCbCr color spaces are supported. Depending on the bitmap format, use of each of these color spaces require specific capabilities to have been negotiated among all peer SICES (see Table 8-5).

Palettized color mapping may be used for uncompressed and JBIG encoding formats. In this case, a color palette of the maximum size 2^P , where P is the number of bits per pixel, is transmitted as part of the bitmap header. Each entry in the palette represents a color which is to be used when a pixel with the value equal to the index of that palette entry is transmitted. The colors within a palette may be specified using one of several possible color spaces either RGB, CIElab or, YCbCr with their optionally associated enhancement parameters. Because colors with neighbouring indices do not necessarily result in similar colors, palettized color mapping is only sensible to use for lossless bitmap encoding formats. Optionally, one palette entry may be defined to represent the transparent color.

A description of each of the color spaces supported within this Recommendation is shown in Table 8-34.

TABLE 8-34/T.126

Bitmap color spaces

Color space	Description
Greyscale	The greyscale colorspace consists of single component values with the value of zero representing black, and the maximum value representing white. The intermediate values vary monotonically between these two extremes. A ColorAccuracyEnhancementGreyscale parameter is optionally specifiable to allow the gamma value of the source colorspace to be indicated.
RGB	The RGB colorspace consists of three component values representing the intensity of the red, green, and blue primaries. For each component, the value of zero represents no contribution of that component, and the maximum value represents the full-intensity primary color. The intermediate values of each component vary monotonically between these two extremes. A ColorAccuracyEnhancementRGB parameter is optionally specifiable to allow the gamma value, color temperature (in degrees Kelvin) and the red, green and blue primary values specified as CIE xy chromaticity coordinates of the source colorspace to be indicated.
YCbCr	The YCbCr colorspace consists of three component values. The interpretation of each color component shall by default be in accordance with ITU-R Recommendation 601-1. A ColorAccuracyEnhancementYCbCr parameter is optionally specifiable to allow the gamma value, color temperature (in degrees Kelvin) and the red, green and blue primary values specified as CIE xy chromaticity coordinates of the source colorspace to be indicated. Alternatively, this parameter can be set to indicate an interpretation of the color components in accordance with ITU-R Recommendation 709.
CIELab	The CIELab colorspace consists of three component values. The interpretation of each color component shall by default be in accordance with Recommendation T.42. A ColorAccuracyEnhancementCIELab parameter is optionally specifiable to indicate the color temperature (in degrees Kelvin) and the CIELab gamut to be used. For each component, the gamut is specified as a span and offset pair. The span represents the range in $L^*a^*b^*$ (continuous) space corresponding to the full dynamic range in the quantized Lab space. The offset represents the value in the quantized Lab space that represents the zero value for that component in the corresponding $L^*a^*b^*$ space.
nonStandardColorSpace	If the appropriate non-standard capability has been negotiated among peer SICEs, a bitmap may be transmitted using a non-standard colorspace. The definition of any non-standard color-spaces is beyond the scope of this Recommendation.

8.4.5 Bitmap color component sampling ratios

For each directly mapped colorspace which includes multiple image components (all except greyscale), the resolution of the pixel arrays for each of the color components depends on the resolution mode specified. Some of the components of the bitmap may be encoded at a resolution lower than the resolution corresponding to the bitmap size indicated in the BitmapCreatePDU depending on the resolution mode specified. A description of the resolution modes supported by this Recommendation is shown in Table 8-35. All modes are not applicable to all bitmap encoding formats.

8.4.6 Bitmap formats

This Recommendation supports multiple bitmap coding formats, not all of which are applicable to all bitmap destinations. Each of the supported bitmap formats are described in the following subclauses.

8.4.6.1 Uncompressed

A bitstream which is encoded in uncompressed format is represented as a packed one or three channel pixel array depending on the selected colorspace and color resolution mode. Each available color mapping mode may or may not be allowed depending on the negotiated capability set, and the bitmap destination. See Table 8-5 for details on capability dependencies.

For an uncompressed bitmap, the bitmapFormatHeader is set to bitmapHeaderUncompressed. The content of this header is shown in Table 8-36.

TABLE 8-35/T.126

Bitmap resolution mode

Resolution mode	Description
4:4:4	For a 4:4:4 bitmap, each color component is maintained at the same resolution in both horizontal and vertical dimensions equal to the size indicated by the bitmapSize parameter in the BitmapCreatePDU. This resolution mode applies to Greyscale, RGB, YCbCr and CIELab colorspace.
4:2:2	For a 4:2:2 bitmap, the luminance component is encoded at the full resolution indicated by the bitmapSize parameter in the BitmapCreatePDU. Each of the two chrominance components are encoded with half the number of pixels in the horizontal dimension. Successive chrominance pixels along a scan-line represent the position corresponding to the location mid-way between successive pairs of luminance pixels. This resolution mode applies to YCbCr and CIELab colorspace.
4:2:0	For a 4:2:0 bitmap, the luminance component is encoded at the full resolution indicated by the bitmapSize parameter in the BitmapCreatePDU. Each of the two chrominance components are encoded with one half the number of pixels in the horizontal dimension and one half the number of pixels in the vertical dimension. Successive chrominance pixels along a scan-line represent the position corresponding to the location mid-way between successive blocks of 2 × 2 of luminance pixels. This resolution mode applies to YCbCr and CIELab colorspace.
NonStandardResolutionMode	If the appropriate non-standard capability has been negotiated among peer SICEs, a bitmap may be transmitted using a non-standard resolution mode. The definition of any non-standard resolution mode is beyond the scope of this Recommendation.

TABLE 8-36/T.126

Uncompressed bitmap format header

Parameter	Description
colorMappingMode	<p>This parameter is a choice of either direct-mapped or palettized-color mapping.</p> <p>Directed mapped</p> <p>The combination of the colorSpace and resolutionMode sub-parameter values shall be limited to those allowed by the appropriate base capability corresponding to the bitmap type implied by the destinationAddress parameter of the BitmapCreatePDU (see Table 8-5). Additional colorspace and resolution modes are valid if the corresponding capabilities have been appropriately negotiated. The appropriate optional accuracyEnhancement parameter of the ColorSpaceSpecifier may also be included to more precisely define the colorspace.</p> <p>Palettized</p> <p>This parameter's value is limited to those allowed by the appropriate base capability corresponding to the bitmap type implied by the destinationAddress parameter of the BitmapCreatePDU (see Table 8-5). This parameter value includes as sub-components a single color palette and a specifier indicating the number of bits per pixel – either 1, 4, or 8. The maximum length of the color palette is determined by the number of bits per pixel (up to 2, 16, or 256 entries for 1, 4, or 8 bits per pixel, respectively). The color palette may be specified in any of the allowable colorspace, and may include the optional accuracyEnhancement parameter. It may also include an entry to indicate one palette entry to represent transparent.</p>

The encoding of an uncompressed bitmap for each of the possible values of the color mapping mode parameter is defined in Table 8-37. In all cases, the image pixel array is encoded from top to bottom with a left to right line scanning. No gaps are left at scan-line boundaries.

TABLE 8-37/T.126

Encoding of uncompressed bitmaps

Color mapping	Description
1-bit palettized	The 1-bit pixel array is encoded in scanning order as described above with each 8 successive bits packed into an octet. The bits are packed beginning with the most significant bit of an octet and proceeding to fill towards the least significant bit.
4-bit palettized	The 4-bit pixel array is encoded in scanning order as described above. The 4-bit pixels are encoded together (they are not encoded as separate bitplane arrays). Pairs of successive 4-bit pixel values are packed into an octet. The first 4-bit value is placed in the most significant four bits, and the second is placed in the least significant four bits.
8-bit palettized	The 8-bit pixel array is encoded in scanning order as described above. The 8-bit pixels are encoded together (they are not encoded as separate bitplane arrays). Successive encoded 8-bit pixel values are placed into successive octets.
8-bit greyscale	The 8-bit pixel array is encoded in scanning order as described above. Successive encoded 8-bit pixel values are placed into successive octets.
RGB 4:4:4	This format is encoded with the three color components interleaved. The image is divided into locations consisting of one pixel from each of the three color components with the same coordinate offset. For each such location, the pixels are encoded in the following order: the R component, the G component, the B component. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.
YCbCr 4:4:4	This format is encoded with the three color components interleaved. The image is divided into locations consisting of one pixel from each of the three color components with the same coordinate offset. For each such location, the pixels are encoded in the following order: the Y component, the Cb component, the Cr component. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.
YCbCr 4:2:2	This format is encoded with the three color components interleaved. The image is divided into locations consisting of a horizontal pair of Y pixels and one pixel from each of the two chrominance components that are co-located within the image. For each such location, the pixels are encoded in the following order: the left-most Y pixel, the right-most Y pixel, the Cb pixel, the Cr pixel. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.
YCbCr 4:2:0	This format is encoded with the three color components interleaved. The image is divided into locations consisting of a 2×2 block of Y pixels and one pixel from each of the two chrominance components that are co-located within the image. For each such location, the pixels are encoded in the following order: the upper-left Y pixel, the upper-right Y pixel, the lower-left Y pixel, the lower-right Y pixel, the Cb pixel, the Cr pixel. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.
CIELab 4:4:4	This format is encoded with the three color components interleaved. The image is divided into locations consisting of one pixel from each of the three color components with the same coordinate offset. For each such location, the pixels are encoded in the following order: the <i>L</i> component, the <i>a</i> component, the <i>b</i> component. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.

TABLE 8-37/T.126 (concluded)

Encoding of uncompressed bitmaps

Color mapping	Description
CIELab 4:2:2	This format is encoded with the three color components interleaved. The image is divided into locations consisting of a horizontal pair of L pixels and one pixel from each of the two chrominance components that are co-located within the image. For each such location, the pixels are encoded in the following order: the left-most L pixel, the right-most L pixel, the A pixel, the B pixel. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.
CIELab 4:2:0	This format is encoded with the three color components interleaved. The image is divided into locations consisting of a 2 × 2 block of L pixels and one pixel from each of the two chrominance components that are co-located within the image. For each such location, the pixels are encoded in the following order: the upper-left L pixel, the upper-right L pixel, the lower-left L pixel, the lower-right L pixel, the A pixel, the B pixel. Successive encoded 8-bit pixels are placed into successive octets. Successive locations are dealt with in scanning order.

8.4.6.2 Recommendation T.4 (G3)

T.4 (G3) encoding of 1-bit per pixel bitmaps is a mandatory capability for terminals supporting the Hard-Copy capability and disallowed otherwise.

For a T.4 encoded bitmap, the bitmapFormatHeader is set to bitmapHeaderT4. The content of this header is shown in Table 8-38.

TABLE 8-38/T.26

T.4 bitmap format header

Parameter	Description
twoDimensionalEncoding	This flag, when set to TRUE indicates that the two-dimensional encoding scheme defined in Recommendation T.4 shall be used. When set to FALSE, the one-dimensional encoding scheme defined in Recommendation T.4 shall be used.

For T.4 encoding, only the one-dimensional coding scheme of 4.1/T.4 and the two-dimensional coding scheme of 4.2/T.4 are supported. Extended two-dimensional coding as well as error limiting mode, error correcting mode, character mode, mixed mode, and file transfer mode are not supported.

The T.4 encoded bitstream is packed into the data field of the bitmapData parameter by filling in successive bits into each octet beginning with the most significant bit of each octet and filling toward the least significant bit.

8.4.6.3 Recommendation T.6 (G4)

T.6 (G4) encoding of 1-bit per pixel bitmaps is a mandatory capability for terminals supporting the Hard-Copy-Image-Bitmap-Format-T.6 capability and disallowed otherwise.

For a T.6 encoded bitmap, the bitmapFormatHeader is set to bitmapHeaderT6. There are no parameters to be specified within this header – all encoding parameters are contained within the T.6 encoded data.

All T.6 encoding options are allowable except for any variable length document options.

The T.6 encoded bitstream is packed into the data field of the bitmapData parameter by filling in successive bits into each octet beginning with the most significant bit of each octet and filling toward the least significant bit.

8.4.6.4 Recommendation T.81 (JPEG)

Recommendation T.81 (JPEG) is an image compression standard optimized for encoding continuous tone color images at a variety of quality levels. The composition of a JPEG bitstream and the associated encoding and decoding algorithm is specified in Recommendation T.81.

For a JPEG bitmap, the `bitmapFormatHeader` is set to `bitmapHeaderT81`. The content of this header is shown in Table 8-39. Because JPEG omits the specification of the colorspace of the coded image, a colorspace specifier is provided as the only parameter to the JPEG Bitmap Format Header.

TABLE 8-39/T.126

JPEG bitmap format header

Parameter	Description
<code>colorSpace</code>	This parameter's value shall be limited to those allowed by the appropriate base capability corresponding to the bitmap type implied by the <code>destinationAddress</code> parameter of the <code>BitmapCreatePDU</code> (see Table 8-5). Additional colorspace are valid if the corresponding capabilities have been appropriately negotiated. The appropriate optional <code>accuracyEnhancement</code> parameter of the <code>ColorSpaceSpecifier</code> may also be included to more precisely define the colorspace.
<code>resolutionMode</code>	This parameter's value shall be limited to those allowed by the appropriate base capability corresponding to the bitmap type implied by the <code>destinationAddress</code> parameter of the <code>BitmapCreatePDU</code> (see Table 8-5). Additional resolution modes are valid if the corresponding capabilities have been appropriately negotiated.

The JPEG encoded bitstream is packed into the data field of the `bitmapData` parameter by filling in successive bits into each octet beginning with the most significant bit of each octet and filling toward the least significant bit.

Note that SI capabilities for JPEG bitmaps couple component interleave ratios and colorspace due to their strong correlation.

See Table 8-5 (Soft-Copy-Image-Bitmap capability) for the limits on JPEG parameters.

All other JPEG modes are available as negotiable capabilities (see Table 8-5). For each negotiated capability, the allowable range of parameters which can be specified in the JPEG Frame Header is shown in Table 8-40.

Allowable JPEG bitstreams for this Recommendation have the restrictions of full interchange format only (all quantizers and Huffman tables must be specified within the image bitstream).

TABLE 8-40/T.126

JPEG frame header parameters

Capability	SOF _n	P	Y	X	Nf	C _i	H _i	V _i	Tq _i
Soft-Copy-Image	SOF ₀	8	(1..image bitmap max. width)	(1..image bitmap max. height)	1 or 3	(0..255)	H ₀ = 1 or H ₀ = 2 H ₁ = 1 H ₂ = 1	V ₀ = 1 or V ₀ = 1 V ₁ = 1 V ₂ = 1	(0..3)
Soft-Copy-Image-Bitmap- Format-T.81- Extended-Sequential-DCT	+SOF ₁	+12	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81-Progressive-DCT	+SOF ₂	+12	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81-Spatial-DPCM	+SOF ₃	+(2..16)	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Differential-Sequential-DCT	+SOF ₅	=	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Differential-Progressive-DCT	+SOF ₆	+12	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Differential-Spatial-DPCM	+SOF ₇	=	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Extended-Sequential-DCT-Arithmetic	+SOF ₉	+12	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap-Format- T.81-Progressive-DCT-Arithmetic	+SOF ₁₀	+12	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Spatial-DPCM-Arithmetic	+SOF ₁₁	+(2..16)	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Differential-Sequential-DCT- Arithmetic	+SOF ₁₃	=	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81- Differential-Progressive-DCT- Arithmetic	+SOF ₁₄	+12	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81-Differential-Spatial- DPCM-Arithmetic	+SOF ₁₅	+(2..16)	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81-YCbCr-4:2:0	=	=	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap- Format-T.81-YCbCr-4:4:4	=	=	=	=	=	=	+ H ₀ = 1 H ₁ = 1 H ₂ = 1	=	=
Soft-Copy-Image-Bitmap- Format-T.81-RGB-4:4:4	=	=	=	=	=	=	+ H ₀ = 1 H ₁ = 1 H ₂ = 1	=	=
Soft-Copy-Image-Bitmap- Format-T.81-CIELab-4:2:0	=	=	=	=	=	=	=	+ V ₀ = 2 V ₁ = 1 V ₂ = 1	=

TABLE 8-40/T.126 (concluded)

JPEG frame header parameters

Capability	SOF _n	P	Y	X	Nf	C _i	H _i	V _i	Tq _i
Soft-Copy-Image-Bitmap-Format-T.81-CIELab-4:2:2	=	=	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap-Format-T.81-CIELab-4:4:4	=	=	=	=	=	=	+ H ₀ = 1 H ₁ = 1 H ₂ = 1	=	=

= Allowable options remain unchanged over those that have been established by other negotiated capabilities or default values.
 + X Add "X" to the allowable options set that has already been established by other capabilities negotiations or default value assumptions.
 SOF_n Start of Frame Marker. The subscript n identifies the encoding process used.
 P Sample precision. Specifies the number of bits per sample of each component.
 Y Number of lines. Specifies the number of lines in the image component with the largest number of lines.
 X Number of samples per line. Specifies the number of samples per line in the image component with the largest number of samples per line.
 Nf Number of image components in the frame.
 C_i Component identifier. Assigns a label to the ith component in the sequence of frame component specification parameters.
 H_i Horizontal sampling factor. Specifies the relationship between the horizontal dimension of each image component.
 V_i Vertical sampling factor. Specifies the relationship between the vertical dimension of each image component.
 Tq_i Quantization table destination selector.

For each of the color component indicated in the JPEG Frame Header, the relationship between the index i and the actual color component for each of the supported colorspace is shown in Table 8-41.

TABLE 8-41/T.126

Order of JPEG color components

Colorspace	C ₀	C ₁	C ₂
YCbCr	Y	Cb	Cr
CIELab	L	A	B
RGB	R	G	B
Greyscale	Y	-	-

The capability Soft-Copy-Bitmap-Format-T.81-Non-Interleaved indicates the ability to support non-interleaved encoding of the color components. Without this capability being present in the negotiated capability set, only fully interleaved encoding is allowed. In this case the parameter N_s in a JPEG Scan Header shall be equal to the number of components in the frame N_f. If the Soft-Copy-Bitmap-Format-T.81-Non-Interleaved is present in the negotiated capability set, the value of N_s may be less than the value of N_f.

8.4.6.5 Recommendation T.82 (JBIG)

Recommendation T.82 (JBIG) is an image compression standard optimized for lossless encoding of text, half-tone and line art images. The composition of a JBIG bitstream and the associated encoding and decoding algorithm is specified in Recommendation T.82.

Because the JBIG Recommendation does not specify the colorspace of the encoded pixel data, an out-of-band bit-stream header is provided in the bitmap format header parameter of the BitmapCreatePDU. For a JBIG bitmap, the bitmapFormatHeader is set to bitmapHeaderT82. The content of this header is shown in Table 8-42. Because JBIG is lossless, it is possible to code palettized as well as greyscale pixel data. Moreover, it is possible to send the pixel data in many scanning orders. To enable the progressive display of palettized images before all bitplanes are known by the displaying entity, intermediate palettes for some or all of the intermediate bitplanes can be optionally provided by the

receiver such that the display is intelligible before all bitplanes are received. These are only useful for lowest resolution layer modes of JBIG that have stripe orderings that transfer all pixel data from one bitplane before moving on to the next. This style of bitmap exchange can be used to progressively build an image in bitplane order allowing a complete raster to be initially covered faster than if all bitplanes are transmitted for each image stripe before moving to the next. See Appendix I for one possible algorithm that can be used to generate intermediate palettes for this purpose.

TABLE 8-42/T.126

JBIG bitmap format header

Parameter	Description
colorMappingMode	<p>This parameter is a choice of either direct-mapped or palettized-color mapping.</p> <p>Direct mapped</p> <p>The combination of the colorSpace and resolutionMode sub-parameter values shall be limited to those allowed by the appropriate base capability corresponding to the bitmap type implied by the destinationAddress parameter of the BitmapCreatePDU (see Table 8-5). Additional colorspace and resolution modes are valid if the corresponding capabilities have been appropriately negotiated. The appropriate optional accuracyEnhancement parameter of the ColorSpaceSpecifier may also be included to more precisely define the colorspace.</p> <p>This parameter's value shall be limited to the those allowed by the appropriate base capability corresponding to the bitmap type implied by the destinationAddress parameter of the BitmapCreatePDU (see Table 8-5). For the direct-mapped case, this parameter includes a ColorSpaceSpecifier used to select the bitmap colorspace from the available set. The appropriate optional accuracyEnhancement sub-parameter of the ColorSpaceSpecifier may also be included to more precisely define the colorspace.</p> <p>Direct mapped colorspace shall have their bitplane data (per color component) coded and transmitted in the following order:</p> <p>Coding order $(C1_{msb}, C2_{msb}, \dots, CN_{msb}), (C1_{msb-1}, C2_{msb-1}, \dots, CN_{msb-1}), \dots, (C1_{lsb}, C2_{lsb}, \dots, CN_{lsb})$ where each component, CN, is an 8-bit quantity in the form $(CN_{msb}, CN_{msb-1}, \dots, CN_{lsb})$. In the case of the RGB colorspace, the red component shall be considered C1, green C2, and blue C3. A specific component ordering must accompany the definition of any new allowable standard or non-standard direct mapped colorspace in the future.</p> <p>Palette mapped</p> <p>This parameter's value is limited to those allowed by the appropriate base capability corresponding to the bitmap type implied by the destinationAddress parameter of the BitmapCreatePDU (see Table 8-5). This parameter includes as a sub-component a single color palette (see 8.4.4) and, optionally, a selfProgressive flag indicating whether or not the bitmapPalette is suitable for the display of bitmap data as it is incrementally decoded and displayed or a series of additional color index tables (progressivePalettes) used to form intermediate palettes for each successive bitplane by referencing colors in the color palette. If the number of color index tables is less than the total number of bitplanes (as indicated within the JBIG encoded bitstream), the bitmapPalette shall be used for bitplanes after the last indicated progressive palette. The length of the bitmapPalette is bounded by the total number of bitplanes. For the progressive palettes, the length shall be equal to 2^P where P is the bitplane number (most significant bitplane = 1) for which it is intended to be used.</p> <p>NOTE – If an equivalent number of color index tables as bitplanes are present, the bitmap palette is not utilized in its transmitted order. Though sending an equivalent number of color index tables as image bitplanes is allowable, the same results can be obtained more efficiently by including no color index table for the final bitplane so long as the bitmap palette is reordered at the transmitter such that the color positions are one-to-one with respect to the color index table that would have been used for the final bitplane.</p>

The JBIG encoded bitstream is packed into the data field of the bitmapData parameter by filling in successive bits into each octet beginning with the most significant bit of each octet and filling toward the least significant bit.

All parameters that control encoding options and limits for JBIG are contained within the BIH (Bi-level Image Header) which is present at the head of all JBIG bitstreams. Allowable ranges for these parameters when JBIG is used in conjunction with this Recommendation are shown in Table 8-43. This table also specifies the additional parameters and ranges that can be negotiated.

8.4.6.6 Non-standard bitmap format

Additional coding formats not in the list of supported bitmap formats are allowed if successfully negotiated. The bitmapFormatHeader, in this case, is encoded as a nonStandardIdentifier. The interpretation of this is beyond the scope of this Recommendation.

8.5 Pointers

Pointers may be applied to any workspace in a conference if the Soft-Copy-Pointing capability is present in the negotiated capability set. Pointers are created by creating a bitmap using the BitmapCreatePDU with the destinationAddress parameter set to softCopyPointerPlane, possibly in conjunction with a BitmapCreateContinuePDU (see 8.4). When a pointer bitmap is created, its destination is not one of the ordinary workspace planes, but a virtual plane which is interpreted to be in front of all other planes in the workspace.

Once created, the position or other attributes of a pointer bitmap may be changed by making use of the BitmapEditPDU or may be deleted using the BitmapDeletePDU as described in 8.4. Unlike other bitmap types, a pointer bitmap may be edited or deleted without the need to have the Soft-Copy-Plane-Editing capability present in the negotiated capability list. Also unlike other bitmap types, editing or deletion of a pointer bitmap is only allowed by the SICE which created it. A SICE shall ignore any received BitmapEditPDU or BitmapDeletePDU which refers to a pointer bitmap but is not sourced from the same SICE which issued the BitmapCreatePDU for that pointer. If a SICE receives a new Application Roster in which the SICE that owns a pointer is no longer present, that pointer shall be considered deleted.

As with other bitmap types, a receiving node may wish to abort a pointer bitmap. It does so by issuing a BitmapAbortPDU as described in 8.4. The response to this request is exactly the same as that of other bitmap types in the case of an editable workspace plane as also described in 8.4. (the aborted bitmap is deleted in its entirety).

With the Soft-Copy-Pointing capability in the negotiated capability set, and no other pointing capabilities, a pointing bitmap may be up to 32 by 32 pixels in size, and may be transmitted in uncompressed format. A larger bitmap size may be used only if the capabilities Soft-Copy-Pointing-Bitmap-Max-Width or Soft-Copy-Pointing-Bitmap-Max-Height are negotiated to a larger value in the capability set. JBIG encoding format may be used if the Soft-Copy-Pointing-Bitmap-Format-T.82 capability is present in the negotiated capability set. The limitations of these two encoding formats for the case of pointer bitmaps is described in 8.4.

8.6 Drawing

Drawing information may be transmitted to any workspace plane which has the annotation flag set in its usage designator. This may only be set if the Soft-Copy-Annotation capability is included in the negotiated capability set (the number of SICEs with this capability need be greater than one for this capability to appear in the negotiated capability set).

Drawing information may be sent to either permanent or editable workspace planes. In the case of permanent planes, the drawing commands overwrite the pixel values over which the drawing objects pass. In the case of editable planes, the drawing elements are treated as separate editable objects. The attributes of these objects may, in this case, be edited as long as the workspace and object continue to exist. Drawing elements may also be deleted from an editable workspace plane.

TABLE 8-43/T.126

JBIG BIH parameters

Capability	D _L	D	P	X _D	Y _D	L ₀	M _x	M _y	Order byte	Options byte
Hard-Copy-Image-Bitmap-Format-T.82	0	0	(1)	(1..image bitmap max. width)	(1..image bitmap max. height)	(1.. Y _D)	(0..127)	(0..255)	HITOLO = 0 SEQ = (0.1) ILEAVE = (0.1) SMID = (0.1)	See Table 8-44
Soft-Copy-Pointing-Bitmap-Format-T.82	0	0	(1..8)	(1..image bitmap max. width)	(1..image bitmap max. height)	(1.. Y _D)	(0..127)	(0..255)	HITOLO = 0 SEQ = (0.1) ILEAVE = (0.1) SMID = (0.1)	See Table 8-44
Soft-Copy-Annotation-Bitmap-Format-T.82	0	0	(1..8)	(1..image bitmap max. width)	(1..image bitmap max. height)	(1.. Y _D)	(0..127)	(0..255)	HITOLO = 0 SEQ = (0.1) ILEAVE = (0.1) SMID = (0.1)	See Table 8-44
Soft-Copy-Image	0	0	(1..8)	(1..image bitmap max. width)	(1..image bitmap max. height)	(1.. Y _D)	(0..127)	(0..255)	HITOLO = 0 SEQ = (0.1) ILEAVE = (0.1) SMID = (0.1)	See Table 8-44
Soft-Copy-Image-Bitmap-Format-T.82-12 Bit	=	=	+12	=	=	=	=	=	=	See Table 8-44
Soft-Copy-Image-Bitmap-Format-T.82-Differential	=	(0..255)	=	=	=	(1..Y _D /2 ^D)	=	=	HITOLO = (0.1) SEQ = (0.1) ILEAVE = (0.1) SMID = (0.1)	See Table 8-44
Soft-Copy-Image-Bitmap-Format-T.82-Differential-Deterministic-Prediction	=	=	=	=	=	=	=	=	=	See Table 8-44
=	Allowable options remain unchanged over those that have been established by other negotiated capabilities or default values.									
+ X	Add "X" to the allowable options set that has already been established by other capabilities negotiations or default value assumptions.									
DL	Lowest resolution layer to be specified in the associated BIE (Bi-level Image Entity).									
D	Final differential layer specified.									
P	Number of bitplanes.									
X _D	Horizontal picture dimension (in pixels) at the highest resolution layer.									
Y _D	Vertical picture dimension (in pixels) at the highest resolution layer.									
L ₀	Number of lines per stripe at the lowest resolution layer.									
M _x	Maximum horizontal offset allowed (in pixels) for AT pixel processing.									
M _y	Maximum vertical offset allowed (in pixels) for AT pixel processing.									
Order byte	Set of parameters in the BIH that specifies the order in which stripe data is concatenated to form BID (Bi-level Image Data).									
HITOLO	Component of the Order Byte that indicates whether or not stripe data is sent from highest to lowest resolution layer when using resolution reduction.									
SEQ	Component of the Order Byte that indicates whether or not corresponding stripes from all resolution layers are sent before proceeding to the next stripe, or whether all stripes within a resolution layer are sent before proceeding to the next resolution layer.									
ILEAVE	Component of the Order Byte that indicates whether or not stripes from multiple bitplanes should be interleaved.									
SMID	Component of the Order Byte that indicates how stripe data is interleaved.									
Options byte	Set of options parameters.									

TABLE 8-44/T.126

JBIG BIH options byte

Capability	LRLTWO	VLENGTH	TPDON	TPBON	DPON	DPPRIV	DPLAST
Hard-Copy-Image-Bitmap-Format-T.82	(0.1)	0	0	(0.1)	0	0	0
Soft-Copy-Pointing-Bitmap-Format-T.82	(0.1)	0	0	(0.1)	0	0	0
Soft-Copy-Annotation-Bitmap-Format-T.82	(0.1)	0	0	(0.1)	0	0	0
Soft-Copy-Image	(0.1)	0	0	(0.1)	0	0	0
Soft-Copy-Image-Bitmap-Format-T.82-12-Bit	=	=	=	=	=	=	=
Soft-Copy-Image-Bitmap-Format-T.82-Differential	=	=	(0.1)	=	=	=	=
Soft-Copy-Image-Bitmap-Format-T.82-Differential-Deterministic-Prediction	=	=	=	=	(0.1)	(0.1)	=
LRLTWO	Component of the Options Byte that indicates whether or not two line templates shall be used for the image.						
VLENGTH	Component of the Options Byte that indicates whether or not the encoded images length (in scan lines) will be determined by floating marker codes in the coded bitstream.						
TPDON	Component of the Options Byte that indicates whether or not differential layer typical prediction shall be used for the coded bitstream.						
TPBON	Component of the Options Byte that indicates whether or not lowest resolution layer typical prediction shall be used for the coded bitstream.						
DPON	Component of the Options Byte that indicates whether or not differential layer typical prediction shall be used for the coded bitstream.						
DPPRIV	Component of the Options Byte that indicates whether or not a private Deterministic Prediction table is specified for the image.						
DPLAST	Component of the Options Byte that indicates whether or not the last Deterministic Prediction table sent shall be used.						

8.6.1 Drawing shapes

Drawing elements may be one of several basic shapes: a point (or series of points), an open polyline, a closed polyline, a rectangle, or an ellipse. The characteristics of these basic shape types are described in the following subclauses.

8.6.1.1 Point

A drawing element of type Point is a sequence of individual points. The anchor point, and the points in the point list correspond to the centers of the points to be drawn. Each point is created from the specified nib shape of the specified thickness. The control point definition is shown in Table 8-45.

TABLE 8-45/T.126

Definition of point drawing shape

Control point index	Description
–	Anchor point/First point to be drawn
0	Second point to be drawn relative to anchor point
1	Third point to be drawn relative to the previous control point
...	
N = (0..254)	Last point to be drawn relative to the previous control point

8.6.1.2 Open polyline

A drawing element of type Open Polyline is a sequence of straight lines connecting the anchor point and successive points in the point list specified in the DrawingCreatePDU. For an open polyline, there is no line connected between the anchor point and the last point in the point list. In the case of an open polyline, there shall be at least one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with less than one point is received, the drawing element shall be ignored. The control point definition is shown in Table 8-46.

TABLE 8-46/T.126

Definition of open polyline drawing shape

Control point index	Description
–	Anchor point
0	First line segment endpoint relative to anchor point
1	Second line segment endpoint relative to the previous control point
...	
N = (1..254)	Last line segment endpoint relative to the previous control point

8.6.1.3 Closed polyline

A drawing element of type Closed Polyline is a sequence of straight lines connecting the anchor point and successive points in the point list specified in the DrawingCreatePDU. For a closed polyline, a line is also connected between the anchor point and the last point in the point list. In the case of a closed polyline, there shall be at least one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with less than one point is received, the drawing element shall be ignored. The control point definition is shown in Table 8-47.

TABLE 8-47/T.126

Definition of closed polyline drawing shape

Control point index	Description
–	Anchor point
0	First line segment endpoint relative to anchor point
1	Second line segment endpoint relative to the previous control point
...	
N = (1..254)	Last line segment endpoint relative to the previous control point. The closing line segment is drawn to the starting point

8.6.1.4 Rectangle

A drawing element of type Rectangle is a rectangular region whose upper left corner and lower right corner are specified by the anchor point and a single point in the point list, respectively. In the case of a rectangle, there shall be exactly one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with more than one point is received, the additional points shall be ignored. If a list with less than one point is received in the case of a DrawingCreatePDU, the drawing element shall be ignored. The control point definition is shown in Table 8-48.

TABLE 8-48/T.126

Definition of rectangle drawing shape

Control point index	Description
–	Anchor point/Upper Left Corner
0	Lower Right Corner relative to anchor point

8.6.1.5 Ellipse

A drawing element of type Ellipse is defined by a bounding rectangle whose upper left corner and lower right corner are specified by the anchor point and a single point in the point list, respectively. The size of this rectangle corresponds to the length of the two axes of the ellipse. The ellipse is positioned so that no part of it protrudes beyond the bounding rectangle. The ellipse drawing type shall not be used unless the Soft-Copy-Annotation-Drawing-Ellipse capability is present in the negotiated capability set. In the case of an ellipse, there shall be exactly one point specified in the point list of a DrawingCreatePDU (in addition to the anchor point). If a list with more than one point is received, the additional points shall be ignored. If a list with less than one point is received in the case of a DrawingCreatePDU, the drawing element shall be ignored. The control point definition is shown in Table 8-49.

NOTE – The bounding rectangle used in defining this drawing element does not correspond to an actual area of the workspace overwritten by this drawing element. Only the line which forms the ellipse itself (and optionally the fill area within the ellipse) is modified in the case of a non-editable workspace plane, or opaque in the case of an editable workspace plane.

TABLE 8-49/T.126

Definition of ellipse drawing shape

Control point index	Description
–	Anchor point/Upper Left Corner of bounding rectangle
0	Lower Right Corner of bounding rectangle relative to anchor point

8.6.1.6 Non-standard

A drawing element of Non-Standard shape is permitted to have from 1 to 256 control points in the point list. The meaning of these control points is beyond the scope of this Recommendation. The control point definition is shown in Table 8-50.

NOTE – Non-Standard drawing shapes should be specified such that the drawing element does not extend beyond the smallest size rectangle which can enclose all of the control points. This allows the decision of whether the rectangular copy region for the WorkspacePlaneCopyPDU surrounds the control points (to determine whether the drawing element is to be included in the copy) to be consistent with whether that copy region surrounds the actual drawing element.

TABLE 8-50/T.126

Definition of non-standard drawing shape

Control point index	Description
–	Anchor point/Non-Standard Control Point 0
0	Non-Standard Control Point 1 relative to anchor point
1	Non-Standard Control Point 2 relative to the previous control point
...	
N = (0..254)	Non-Standard Control Point N relative to the previous control point

8.6.2 Drawing element attributes and parameters

When a drawing element is created, it is defined to have a number of characteristics: display sample rate, rotation, pen color, pen thickness, pen nib, line style, fill color, solid or highlighted, view state, and depth order. Of these, all but the first are attributes which may be modified after the drawing element was created if the drawing element is placed on an editable workspace plane.

8.6.2.1 Sample rate

The sample rate is an optional parameter to the DrawingCreatePDU which only applies for certain drawing types. For a point, open polyline, or closed polyline, the sample rate parameter provides a recommended rate at which the sequence of control points (and corresponding connecting lines in the case of polylines) should be displayed. This may be useful for preserving the apparent drawing rate of the user at receiving sites. The value of this parameter is specified in units of samples per second. A receiving terminal may choose to ignore this parameter. For drawing types other than point, open polyline, or closed polyline, or non-standard, this parameter shall not be included in the DrawingCreatePDU. If received, in such a case, this parameter shall be ignored. The interpretation of this parameter in the case of a non-standard drawing type is beyond the scope of this Recommendation.

8.6.2.2 Rotation

This is an optional parameter which determines the rotation to be applied to the specified drawing element. This consists of two components: the angle of rotation, and the axis of rotation. The angle of rotation is specified in integer units of minutes of arc within the range (0..21599). The direction of rotation is defined to be counterclockwise. The rotation is performed relative to a specified axis of rotation. This axis is specified as a relative offset from the anchor point. Its coordinates are bounded to the range (-32768..32767).

8.6.2.3 Bounding rectangle

This is a parameter which defines the extent of the drawing element relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this drawing element is to be considered within the copy region. If the bounding rectangle is fully contained within the copy region, this drawing element shall be included in the region copied. This is regardless of the actual region spanned by the control points of the object. If this parameter is not specified, the control points of the object determine whether the drawing element is enclosed by the copy region. This parameter is mandatory if a non-zero rotation is specified for the drawing element on an editable workspace plane. This is to avoid ambiguities in determining the location of rotated control points.

8.6.2.4 Pen color

The pen color parameter is an optional parameter (part of the drawing attribute set) which determines the color applied to the line portion of a drawing element (as opposed to the fill region). The pen color may be specified in the colorspace allowed given the negotiated capability set for the designated workspace (see 8.3.1.4). If the true color capability has been negotiated, the color value may be specified either using the palette or as a true-color value. The pen color may also be specified as Transparent. In the case of a permanent plane, drawing using the Transparent color sets the modified pixels to Transparent, erasing what was on that plane. In the case of an editable plane, the Transparent color simply makes the drawing object invisible – objects in the same plane which are below it can still be seen through. If the pen color is not specified, a value of Black is used.

8.6.2.5 Pen thickness

The pen thickness parameter is an optional parameter (part of the drawing attribute set) which determines the thickness of the line. The pen thickness specifies the width of pen nib used to draw the line portion of a drawing element in units of pixels. If drawing is supported in a conference, the pen thickness range of 3 to 16 pixels must be supported. A wider range than this may be negotiated via the Soft-Copy-Annotation-Drawing-Pen-Min-Thickness and Soft-Copy-Annotation-Drawing-Pen-Max-Thickness capabilities. The minimum thickness may be negotiated as low as one pixel, and the maximum as high as 255 pixels. If this parameter is not specified, a default value of 3 shall be used.

8.6.2.6 Pen nib

The pen nib parameter is an optional parameter (part of the drawing attribute set) which determines the shape of the nib used to draw the line portion of drawing elements. All lines are composed of a nib continuously translated along the path of the line (or curve). In the case of a line style with dots or dashes, the nib is translated along the path of the line with periodic gaps where the nib is effectively raised and lowered again on the other side of the gap. If the pen nib parameter is not present, a circular nib shall be used. The circular nib is defined to be a solid circle of diameter equal to the pen thickness. If the Soft-Copy-Annotation-Drawing-Pen-Square-Nib parameter is present in the negotiated capability set for the designated workspace, a square nib may be used instead by specifying square as the pen nib parameter. The square nib is defined to be a solid square region with the pen thickness as the length of each side.

8.6.2.7 Line style

The line style is an optional parameter (part of the drawing attribute set) which determines the type of line to be drawn. The line style is a choice of one of the following:

- Solid;
- Dashed;
- Dotted;
- Dash-Dot;
- Dash-Dot-Dot;
- Two-Tone.

For a solid line, all pixels along the path of the line are drawn using the designated pen color. For a dashed, dotted, dash-dot, or dash-dot-dot line style, the line is drawn by periodically leaving gaps across which the nib is effectively raised and lowered again on the other side of the gap. The pattern of switching is determined by which of these styles was selected. If this parameter is not specified, a solid line style shall be used as the default.

A Two-Tone line style is a solid line which is drawn using the designated pen color for the inner 50 percent of the line's width, and the complementary color for the 25 percent of the line width along each edge.

A non-standard line style may also be used if the corresponding non-standard capability has been negotiated.

8.6.2.8 Fill color

The fill color parameter is an optional parameter (part of the drawing attribute set) which determines whether or not a drawing element will be filled, and if so, the color of the filled region. If the attribute is not present in the attribute list, the drawing element is not filled. If the attribute is present, it specifies the color of the fill region using a color valid given the negotiated capability set (the same color representations valid for the pen color). In the case of a permanent workspace plane, if the Transparent color is specified, the fill region in the plane is set to the Transparent color. For an editable plane, this is equivalent to choosing not to fill, in that the fill region will have no effect on the resulting image.

If the attribute is present, the fill region is determined by the drawing shape. For a point type, the fill color has no effect (the fill region is the null set). For a closed polyline type, the fill region is the set of pixels which are enclosed by the polyline. If the polyline crosses over itself one or more times, the fill region is the set of regions enclosed by each closed loop formed by the polyline. For an open polyline, the fill region is the same region that would have been enclosed by a closed polyline with the same set of control points. For a rectangle and ellipse, the fill region is the interior of the two shapes.

8.6.2.9 Highlighting

The highlighting flag is an optional parameter (part of the drawing attribute set) which determines whether the drawing element is treated as a solid colored image, or if it is treated as a semi-transparent highlight. The use of highlighting is permitted only if the Soft-Copy-Annotation-Drawing-Highlight capability is present in the negotiated capability set. If the attribute is set, the highlighting effect applies to the line portion of the drawing element as well as to the fill region, if any.

When the highlighting attribute is set, the effect depends somewhat on whether the drawing element is directed at a permanent or editable workspace plane. In either case, the result should appear the same, but the means to achieve this are somewhat different.

In the case of an editable plane, the attributes of the drawing element are set to indicate the use of highlighted colors. When the image is rendered, instead of occluding objects and planes behind the highlighted object, the color which would have resulted from these objects is modified in such a way that it appears that the objects and planes behind the drawing element is covered by a semi-transparent object of the designated color. The detailed rule for making the object appear this way shall be locally defined.

In the case of a permanent plane, when the drawing element is created in a plane, it modifies the content of the plane to semi-transparent rather than solid colors. The semi-transparent colors affect the planes behind the designated plane in the manner described for the editable case (also, see 8.3.1.1). In this case, however, the values of the semi-transparent color at each pixel on the designated plane over which the drawing object passes are not necessarily all the same. For pixels in the designated plane which had previously been Transparent, the color of the drawing object at that pixel is used as is. For pixels which were previously set to a color (either solid or semi-transparent), the pixel value is modified to a new semi-transparent color which is chosen to make it appear as if the previous color were overwritten with the semi-transparent color specified for the drawing element. The detailed rule for choosing this color shall be locally defined.

8.6.2.10 View state

The view state is an optional parameter (part of the drawing attribute set) which only affects drawing elements on editable workspace planes. For a permanent plane, this parameter shall be ignored. This parameter may be set to either selected, unselected, hidden, or nonStandardViewState. If the parameter is not present in the attribute list, the unselected setting shall be assumed. If the attribute is set to the selected state, the form of the drawing element may be locally modified to indicate that the object is in the selected state. The method used to indicate the selected state shall be locally defined. If the attribute is set to the hidden state, the drawing element should be removed from view but remain in the local database so that it may later be restored to view. A nonStandardViewState may be specified only if the corresponding non-standard capability is part of the negotiated capability set.

8.6.2.11 Depth order

Depth order is an optional parameter (part of the drawing attribute set) which is used to determine the stacking order of objects within an editable plane. For a permanent plane, this parameter shall be ignored. This parameter may be set to either Front or Back. If set to Front, the drawing element is placed in front of all other objects in the plane. If set to Back, the drawing element is placed in back of all other objects on the plane.

Unlike other attributes, this attribute does not necessarily persist as new drawing elements are created, or as existing drawing elements are edited. If a drawing element is placed at the front, for example, if the depth order attribute of another drawing element is set to Front, the previous drawing element is no longer at the front of all objects on the plane.

8.6.3 Creating drawing elements

A drawing element is created by sending a DrawingCreatePDU to all peer SICEs in a conference. This is done using MCS data primitives as described in Table 6-3. The parameters of the DrawingCreatePDU are shown in Table 8-51. This SIPDU shall only be sent if the Soft-Copy-Annotation capability is included in the negotiated capability set.

TABLE 8-51/T.126

DrawingCreatePDU

Parameter	Description
drawingHandle (Optional)	Unique Handle allocated by GCC using the GCC-Registry-Allocate-Handle primitive. This handle is used to identify this drawing element in all future references. It is needed only if the drawing element is destined for an editable workspace plane.
destinationAddress	This parameter consists of a SoftCopyDataPlaneAddress which consists of a workspaceHandle identifying the handle of the destination workspace and a plane ID indicating the plane to which the drawing element shall be created.
drawingType	This parameter indicates the shape of the drawing element, either point, openPolyline, closedPolyline, rectangle, ellipse, or nonStandardDrawingType. The Ellipse setting may only be used if the Soft-Copy-Annotation-Drawing-Ellipse capability is present in the negotiated capability set. The nonStandardDrawingType may only be used if the corresponding non-standard capability is present in the negotiated capability set. See 8.6.1 for a description of this parameter.
attributes (Optional)	Drawing element attributes controlling certain appearance characteristics. See Table 8-52 for details.
anchorPoint	This parameter specifies the origin of the drawing element within the workspace. All other points including the remaining control points and axis of rotation are specified relative to this control point. The anchor point may be specified in the range (-21845..43690).
rotation (Optional)	This parameter indicates an angle of rotation to be applied to the drawing element and the axis of rotation relative to the anchorPoint. The angle is specified in integer units of minutes of arc in the range (0..21599). The angle is specified as a counterclockwise rotation. The axis of rotation is a positional offset from the anchor point specified in the range (-32768..32767). If this parameter is not specified, an angle of zero degrees (no rotation) is assumed. See 8.6.2.2 for a description of this parameter.
boundingRectangle (Optional)	This is a parameter which defines the extent of the drawing element relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this drawing element is to be considered within the copy region. If the bounding rectangle is fully contained within the copy region, this drawing element shall be included in the region copied regardless of the actual region spanned by the control points of the object. If this parameter is not specified, the control points of the object determine whether the drawing element is enclosed by the copy region. This parameter is mandatory if a non-zero rotation is specified for the drawing element on an editable workspace plane. This is to avoid ambiguities in determining the location of rotated control points.
sampleRate (Optional)	This is an optional parameter that indicates the rate at which it is recommended that the display device draw successive points in the point list. This parameter shall be ignored in the case of Rectangle or Ellipse drawing types. See 8.6.2.1 for a description of this parameter.
pointList	This parameter is a list of control points if there are any needed in addition to the anchor point. The first control point is specified relative to the anchor point. Successive control points following this are encoded relative to the previous point in the point list. When received, however, the value of each control point shall be locally translated into offsets from the anchor point so that later editing of intermediate points does not effect the location of subsequent points in this list. Later editing of the anchor point, however, does directly effect the positions of the other points. The coordinates of these points are bounded to either (-8..7), (-128..127), or (-32768..32767). Points in the point list which extend beyond the (-21845..43690) range of a workspace coordinate point shall be ignored. In the case of drawing types Point, Open Polyline, or Closed Polyline, there may be from 0 to 255 points in the list. In the case of a Rectangle or an Ellipse, there shall be exactly one point. See 8.6.1 for a description of how the control points in the point list are interpreted for each case.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-52/T.126

Drawing attributes

Attribute	Default Value	Description
penColor	black	This attribute indicates the color to be used to draw the line portion of this drawing element. If the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable, then the pen color can later be changed using the DrawingEditPDU. See 8.6.2.4 for a description of this parameter.
fillColor	no fill	This attribute indicates the color to be used to fill the interior region of this drawing element. If the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable, then the fill color can later be changed using the DrawingEditPDU. See 8.6.2.8 for a description of this parameter.
penThickness	3 pixels	This attribute indicates the thickness of the line portion of this drawing element in units of pixels. If the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable, then the pen thickness can later be changed using the DrawingEditPDU. See 8.6.2.5 for a description of this parameter.
penNib	round	This attribute indicates the shape of the nib used to draw the line portion of this drawing element. The nib shape may be set to either round or square. A square nib may only be used if the Soft-Copy-Annotation-Drawing-Square-Nib capability is present in the negotiated capability set. If the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable, then the pen nib can later be changed using the DrawingEditPDU. See 8.6.2.6 for a description of this parameter.
lineStyle	solid	This attribute indicates the line style used to draw the line portion of this drawing element. If the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable, then the line style can later be changed using the DrawingEditPDU. See 8.6.2.7 for a description of this parameter.
highlight	FALSE (not highlighted)	This attribute indicates whether the pen and fill colors should be solid or highlighted (semi-transparent). The highlighted setting is only allowed if the Soft-Copy-Annotation-Drawing-Highlight capability is present in the negotiated capability set. If the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable, then the highlight flag can later be changed using the DrawingEditPDU. See 8.6.2.9 for a description of this parameter.
viewState	unselected	<p>This attribute indicates the state of view for this drawing element. This parameter is allowed only if the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable. The viewState can later be changed using the DrawingEditPDU. It can take on one of the following values:</p> <ul style="list-style-type: none"> • unselected: Drawing element should be displayed normally. • selected: Drawing element should be displayed with some unspecified visual highlighting to indicate that the bitmap is selected and that edit or delete operations to the bitmap may be imminent (not applicable to pointer bitmaps). • hidden: Drawing element should be removed from view but remain in the local database. • nonStandardViewState. <p>See 8.6.2.10 View state for a description of this parameter.</p>

TABLE 8-52/T.126 (concluded)

Drawing attributes

Attribute	Default Value	Description
zOrder	front	This attribute indicates whether this drawing element shall initially be placed at the front or rear of all other objects on the plane. This parameter is allowed only if the Soft-Copy-Plane-Edit capability is present in the negotiated capability list and if the workspace plane has been designated editable. The depth order can later be changed using the DrawingEditPDU. See 8.6.2.11 for a description of this parameter.
nonStandardAttribute	–	This attribute is specified as a nonStandardIdentifier. To be used, it must have been successfully negotiated by a corresponding non-standard capability. Its interpretation is not specified by this Recommendation. An arbitrary number of different nonStandardAttributes may be included in the attributes list.

8.6.4 Deleting drawing elements

A drawing element may be deleted by sending a DrawingDeletePDU to all peer SICEs in a conference. This is done using MCS data primitives as described in Table 6-3. The parameters of the DrawingDeletePDU are shown in Table 8-53. This SIPDU shall only be sent if the Soft-Copy-Annotation and Soft-Copy-Editing capabilities are included in the negotiated capability set, and shall only be directed at a workspace plane which has been designated as editable.

TABLE 8-53/T.126

DrawingDeletePDU

Parameter	Description
drawingHandle	Handle corresponding to the drawing element to be deleted. This shall be the same value as the drawingHandle specified in the DrawingCreatePDU which created this drawing element.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.6.5 Editing Drawing Elements

The attributes of a drawing element, or series of drawing elements may be modified by sending a DrawingEditPDU to all peer SICEs in a conference. This is done using MCS data primitives as described in Table 6-3. The parameters of the DrawingEditPDU are shown in Table 8-54. This SIPDU shall only be sent if the Soft-Copy-Annotation and Soft-Copy-Plane-Editing capabilities are included in the negotiated capability set, and shall only be directed at a workspace plane which has been designated as editable.

TABLE 8-54/T.126

DrawingEditPDU

Parameter	Description
drawingHandle	Handle corresponding to the drawing element to be edited. This shall be the same value as the drawingHandle specified in the DrawingCreatePDU which created this drawing element.
attributeEdits (Optional)	List of drawing attributes and associated values to be modified. The allowable values for list members are described in Table 8-52.
anchorPointEdit (Optional)	This parameter specifies the position of the anchor point within the destination workspace relative to which all other control points are defined. If this parameter is not present, the anchor point is not modified.
rotationEdit (Optional)	This parameter indicates an angle of rotation to be applied to the drawing element and the axis of rotation relative to the anchor Point. The angle is specified in integer units of minutes of arc in the range (0..21599). The angle is specified as a counterclockwise rotation. The axis of rotation is a positional offset from the anchor point specified in the range (-32768..32767). If this parameter is not specified, the rotation angle is not modified. See 8.6.2.2 for a description of this parameter.
boundingRectangle Edit (Optional)	This is a parameter which defines the extent of the drawing element relative to the anchor point. If this parameter is specified, this region is used when copying an editable workspace plane region in determining whether this drawing element is to be considered within the copy region.
pointListEdits (Optional)	This is a parameter that may be used to alter the control points of the drawing element. If present, this specifies a list of one or more edits. Each edit consists of an initial index referencing a control point to be edited, the new value of that control point specified relative to the anchor point of the drawing element, as well as an optional sequence of control point values to be applied to those control points with successive indices following the initially specified index. The new value of each control point in this list is encoded relative to the previous one in the list, with the first being encoded relative to the value of the control point referenced by the initial index. When received, the value of each control point shall be locally translated into offsets from the anchor point so that later editing of intermediate points does not effect the location of subsequent points in this list. The additional sequence of control points, if present, are specified bounded to either (-8..7), (-128..127), or (-32768..32767). Points in the point list which extend beyond the (-21845..43690) range of a workspace coordinate point shall be ignored. In the case of drawing types Point, Open Polyline, or Closed Polyline, there may be up to 255 point list edits. In these cases, if the index value corresponds to a control point already included in the drawing element, the position of that control point is modified to the new position. If the index refers to a value which was not present in the current drawing element, a new control point is added. In this way, point sequence or polylines may be appended after creation. The ability to append additional control points to an existing drawing element shall only be allowed by the SICE which created the drawing element. When appending new control points, the new indices shall be contiguously specified from the last control point index previously defined. That is, there shall be no unspecified indices between control points. If control points are defined which are not contiguous, control points with a higher number than the ones which are not defined shall be ignored by the receiver. In the case of Rectangles or Ellipses, there shall be no more than one control point in this list.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.7 Remote events

When a workspace is created, it may be designated to be capable of accepting remote keyboard or pointing device events from other SICEs in the conference by setting the Accept Keyboard Events and/or the RemotePointingDeviceEvents flag in the WorkspaceCreatePDU. For any workspace which has indicated either or both of these abilities, any SICE in the conference may send corresponding remote events to this workspace if the creator of the workspace is still in the conference.

To do this, the SICE is required to first request permission to issue the corresponding event PDUs by issuing a RemoteEventPermissionRequestPDU with the parameters set according to Table 8-55. The SICE shall then wait for a RemoteEventPermissionGrantPDU from the creator of the workspace with the destinationUserID parameter set to its MCS User ID. See Table 8-56. The event permissions list contained within this PDU defines the types of events the SICE is allowed to source without being ignored. Receipt of a RemoteEventPermissionGrantPDU at a later time with a different permission list indicates that the SICEs permissions have been changed. The RemoteEventPermissionGrantPDU is sent on the SI-CHANNEL to allow all SICEs in the conference to be aware of the activity.

A SICE wishing to relinquish its ability to source a type of remote event may do so by issuing a RemoteEventPermissionGrantPDU with the RemoteEventPermissionList parameter set to the values indicating the event privileges wishing to be relinquished.

The administration of remote event privileges for a workspace is at the discretion of the workspace creator. The conductor privileges mechanism, when in conducted mode, form an additional constraint on the ability to source remote events. The source SICE must have remote event privileges granted by both the conductor and the workspace creator.

NOTE – It is recommended that workspaces created for the purpose of accepting remote events be specified to have the preserve flag set in the workspace attributes. This helps to ensure (although it does not guarantee) that the workspace will not be deleted if a SICE should create a new workspace with a view in the Focus state.

TABLE 8-55/T.126

RemoteEventPermissionRequestPDU

Parameter	Description
destinationAddress	This parameter shall be set to the value softCopyWorkspace with its sub-parameter set to a Unique Handle indicating the workspace to which the sending SICE wishes to issue remote events.
remoteEventPermissionList	Set of one or more of the following values: keyboardEvent, pointingDeviceEvent, nonStandardEvent.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-56/T.126

RemoteEventPermissionGrantPDU

Parameter	Description
destinationAddress	This parameter shall be set to the value softCopyWorkspace with its sub-parameter set to a Unique Handle indicating the workspace to which the SICE indicated by the destinationUserID has been granted the permissions indicated in the remoteEventPermissionList.
destinationUserID	MCS User ID of destination node.
remoteEventPermissionList	Set of one or more of the following values: keyboardEvent, pointingDeviceEvent, nonStandardEvent In the case of the pointingDeviceEvent permission, this entry may optionally include a Handle which corresponds to the pointer bitmap which corresponds to the controlled pointing device from this SICE for this workspace.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.7.1 Remote keyboard events

If a SICE has been granted the keyboardEvent permission by the mechanism described in clause 8.7, the SICE may send a keyboard event, it shall do so by sending a RemoteKeyboardEventPDU to the SICE which created the workspace (the issuer of the WorkspaceCreatePDU). This is done in the manner indicated in Table 6-3. The content of the RemoteKeyboardEventPDU is shown in Table 8-57.

The action taken by the workspace owner on receipt of a RemoteKeyboardEventPDU is a locally defined matter beyond the scope of this Recommendation.

TABLE 8-57/T.126

RemoteKeyboardEventPDU

Parameter	Description
destinationAddress	This parameter shall be set to the value softCopyWorkspace with its sub-parameter set to a Unique Handle indicating the workspace to which to direct the pointing device event. This shall equal the value of the Workspace Handle in the WorkspaceCreatePDU of the corresponding workspace.
keyModifierStates (Optional)	This is a list of optional key modifiers indicating whether or not the keyCode should be conditioned by any combination modifiers. The possible modifiers are leftAlt, rightAlt, leftShift, rightShift, leftControl, rightControl, leftSpecial, rightSpecial, numberPad, scrollLock, or nonStandardModifier. There may be an arbitrary number of different nonStandardModifiers included in the list. A nonStandardModifier may only be used if the corresponding non-standard event permission has been granted in the RemoteEventPermissionGrantPDU. Left and right correspond to left and right handed versions of each basic modifier. The numberPad modifier indicates whether the key code was actuated from a numeric keypad rather than a primary keypad.
keyPressState	This parameter is used to indicate the action of a key indicated by this SIPDU. It may be one of none, keyPress, keyDown, keyUp, or nonStandardKeyPressState. A nonStandardKeyPressState may only be used if the corresponding non-standard event permission has been granted in the RemoteEventPermissionGrantPDU. The none case may be used for a key which is already in the down state if the keyModifierStates have changes. The keyPress action indicates an instantaneous key press action. This may be used, for example, to signal key-presses due to auto-repeat if the key is already in the key-down state. The keyDown and keyUp actions represent a state change of the key to a pressed or non-pressed position, respectively. Successive keyDown actions without and intervening keyUp action for the same key shall be ignored by the receiver. Successive keyUp actions without intervening keyDown actions for the same key shall be ignored by the receiver. Before relinquishing keyboard event privileges for this workspace, it is recommended that a SICE set all keys in the keyDown state to the keyUp state.
keyCode	This parameter is a choice of either a single Unicode encoded character, a choice of one of 32 function (F) keys, or a choice of one of the following edit keys: upArrow, downArrow, leftArrow, rightArrow, pageUp, pageDown, home, end, insert, delete or nonStandardKey. A nonStandardKey may only be used if the corresponding non-standard capability has been negotiated.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.7.2 Remote pointing device events

When a SICE that has been granted pointingDeviceEvent permission wishes to send a pointing device event, it shall do so by sending a RemotePointingDeviceEventPDU to the SICE which created the workspace (the issuer of the WorkspaceCreatePDU). This is done in the manner indicated in Table 6-3. The content of the RemotePointingDeviceEventPDU is shown in Table 8-58. The initial remote pointer position shall be undefined and all button states shall be assumed to be buttonUp by the SICE granting the pointingDeviceEvent permission until receipt of the first RemotePointingDeviceEventPDU. Note that a SICE may grant the pointingDeviceEvent permission to multiple other SICES. How multiple pointing device states are simultaneously interpreted or merged is beyond the scope of this Recommendation. Note, though, that care must be taken to properly handle the case of a SICE with pointingDeviceEvent permission leaving the conference or having its pointingDeviceEvent permission revoked with any button not in a buttonUp state.

The action taken by the workspace creator on receipt of a RemotePointingDeviceEventPDU is a locally defined matter beyond the scope of this Recommendation. Before relinquishing the pointingDeviceEvent permission for this workspace, it is recommended that the SICE send an event setting the state of all buttons to buttonUp if the last RemotePointingDeviceEventPDU sent by the SICE had any button states set to anything other than buttonUp.

If the creator of the workspace had indicated the Handle of the pointer bitmap corresponding to this SICE, the SICE which is controlling that pointing device may optionally track incoming edits to the location of the pointer bitmap with this Handle value. This SICE may optionally choose to alter its local display so that the corresponding pointer bitmap is positioned locally, ignoring the incoming position changes. This could be used to provide more rapid response time to movement of the local pointing device.

TABLE 8-58/T.126

RemotePointingDeviceEventPDU

Parameter	Description
destinationAddress	This parameter shall be set to the value softCopyWorkspace with its sub-parameter set to a Unique Handle indicating the workspace to which to direct the pointing device event. This shall equal the value of the Workspace Handle in the WorkspaceCreatePDU of the corresponding workspace.
leftButtonState	This parameter indicates the state of the left button of the remote pointing device. Allowable values are detailed in Table 8-59.
middleButtonState	This parameter indicates the state of the middle button of the remote pointing device. Allowable values are detailed in Table 8-59.
rightButtonState	This parameter indicates the state of the right button of the remote pointing device. Allowable values are detailed in Table 8-59.
initialPoint	This parameter shall indicate the initial position within the workspace of the pointing device at the time of the event. The point is represented in workspace coordinates (see 8.3.1.2).
sampleRate (Optional)	This is an optional parameter that indicates the rate at which the remote pointing device acquired the successive pointing device coordinates expressed in this PDU. The units are in samples/second (1..255).
pointList (Optional)	This parameter is a list of differential coordinates that are specified relative to their predecessors (the first one is specified relative to the initialPoint parameter). This list is used to collapse multiple sequential pointing device translation events that have the same button states into one transaction.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-59/T.126

Pointing device button events

Button Event	Description
buttonUp	Button is up.
buttonDown	Button is down.
buttonDoubleClick	Button was pressed within a locally determined time interval that constitutes a double click. If the receiving SICE has no need to distinguish double clicks, then it can be interpreted as a buttonDown.
buttonTripleClick	Button was pressed within a locally determined time interval that constitutes a triple click. If the receiving SICE has no need to distinguish triple clicks, then it can be interpreted as a buttonDown.
buttonQuadClick	Button was pressed within a locally determined time interval that constitutes a quad click. If the receiving SICE has no need to distinguish quad clicks, then it can be interpreted as a buttonDown.
nonStandardButtonEvent	A non-standard button event has occurred. A nonStandardButtonEvent may only be used if the corresponding non-standard event permission has been granted in the RemoteEventPermissionGrantPDU.

8.7.3 Remote printing events

At any time, a SICE may request that a workspace be printed at all nodes in a conference which are capable of doing so by issuing a RemotePrintPDU to all peer SICES in the conference. This is done in the manner described in Table 6-3. The content of the RemotePrintPDU is shown in Table 8-60. In the case of an unsynchronized workspace, this SIPDU shall be transmitted on the High priority data channel. In the case of a synchronized workspace, a SICE shall send this SIPDU three separate times on each of the three priority channels: High, Medium, and Low. This is done to ensure that the same information is applied to the workspace before printing so that it is consistently printed at all nodes.

TABLE 8-60/T.126

RemotePrintPDU

Parameter	Description
destinationAddress	This parameter shall be set to the value softCopyWorkspace with its sub-parameter set to a Unique Handle indicating the workspace to be printed. This shall equal the value of the Workspace Handle in the WorkspaceCreatePDU of the corresponding workspace.
numberOfCopies (Optional)	Optional indication of the number of copies desired to be printed at the remote sites. If this parameter is not present, it is assumed that one copy is to be printed.
portrait (Optional)	TRUE specifies a portrait paper orientation. FALSE specifies a landscape paper orientation. If the parameter is not present, no particular orientation is preferred.
regionOfInterest (Optional)	Optionally specifies a rectangular region to be printed within the designated workspace. If this parameter is not present, it is implied that the entire workspace is to be printed.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

When a SICE receives a RemotePrintPDU, if it does not have a copy of the designated workspace (if it joined the conference after this workspace was created), it shall ignore this SIPDU. If the SICE is not capable of printing, it may also ignore this SIPDU.

In the case of an unsynchronized workspace, a RemotePrintPDU received on any priority channel except High shall be ignored. Upon receipt of a RemotePrintPDU on the High priority channel, a SICE which wishes to respond to print requests shall print immediately upon receiving this SIPDU.

In the case of a synchronized workspace, on receiving a RemotePrintPDU from a requesting node on any one of the three priority channels (High, Medium, or Low), a SICE shall stop applying updates that are received on that channel to the designated workspace. It shall continue applying workspace updates to the workspace from each of the other two priority channels until it receives a RemotePrintPDU from the same requesting node specifying the same workspace on each of these priority channels. Upon receiving each of these RemotePrintPDUs, it shall stop applying updates to the workspace received on that channel until the workspace has been printed or copied to an area to ready it for printing. While waiting for the remaining RemotePrintPDUs, if the SICE receives a GCC-Application-Roster-Report from the GCC provider which indicates that the SICE which originated the print request is no longer enrolled in the conference, the SICE shall stop waiting for the remaining requests and continue applying workspace updates from all priority channels to this workspace (unless another print request is pending for the same workspace).

8.8 Archives

An archive is a collection of workspaces which may be saved beyond the extent of a single conference. If an archive is present at every peer SICE in a conference, it may be opened by a SICE if the Archive-Support capability is present in the negotiated capability set. Once opened, that SICE may perform workspace operations on workspaces contained within. An archive may be opened for reading, writing, creation (of a new archive), or any combination of these. Multiple SICES may simultaneously open an archive for reading, but only one SICE may open the same archive for creation or writing. To open an archive, the ArchiveOpenPDU shall be broadcast to all peer SICES. This is done in the manner described in Table 6-3. The content of the ArchiveOpenPDU is shown in Table 8-61. Upon receiving the ArchiveOpenPDU, all SICES shall send an ArchiveAcknowledgePDU to the SICE requesting the open on its Used ID channel. See Table 8-62. The SICE requesting the open shall wait until all peer SICES marked as Active in the current application roster have responded to the open request before proceeding with archive operations. If not all Active peer SICES acknowledge, the archive shall not be considered open. The SICE shall also monitor any roster changes as indicated from GCC so as to be able to recognize when an Active peer SICE that has not responded has left the conference. If the roster changes with any new SICES being added to the conference since the roster instance that was valid at the time the archive was opened, the archive is assumed to be automatically closed and that another ArchiveOpenPDU must be transmitted in order to perform any further archive operations.

TABLE 8-61/T.126

ArchiveOpenPDU

Parameter	Description
archiveHandle	Unique Handle used to reference the archive volume once it has been opened.
mode	This parameter indicates the action to be performed on the archive. This parameter may be set to Read, Write, Create, or a combination of these. The Read setting indicates that the archive shall be opened for reading. The Write setting indicates that the archive shall be opened for writing. The create setting indicates that a new archive shall be created. In this case, if an archive of the same name already exists, this operation shall fail.
header	The header parameter is the identifier by which the archive shall be referenced for the duration of its existence. The header is formed from an archiveName, archiveCreationTime which indicates the creation time of the archive, and archiveModificationTime which indicates the time that the contents of the archive were last modified.
maxEntries (Optional)	This parameter shall be present if the Archive Mode is set to Create, and shall not be present otherwise. This parameter, which may span the range (1..65535) indicates the maximum number of entries which may be placed in the archive. This can be used to allow the receiving nodes to attempt to allocate space for the archive, allowing them to indicate an error condition if they do not have sufficient space.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-62/T.126

ArchiveAcknowledgePDU

Parameter	Description
archiveHandle	Unique Handle used to reference the archive volume once it has been opened.
result	Result of the archive open operation. Either archiveOpenSuccessful, archiveNotFound (when opening for reading or writing), archiveTimeIncorrect (when opening for reading or writing), archiveExists (when opening for creation), archiveOpenForWriting (when opening for writing), storageExceeded, unspecifiedError, or nonStandardResult. In the case of archiveTimeIncorrect, the actual header of the archive that was found is included in the result to allow the SICE requesting the open to determine which version of the archive was present at the acknowledging SICE.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

Once an archive has been opened, it may be operated on by the SICE which opened it using any of the workspace PDUs. A SICE which has not opened an archive shall not perform any archive operations on that archive. If an archive has been opened for reading but not writing (or creation), workspaces may not be created, deleted, or edited. Only workspace plane copy operations are allowed where the source is a workspace in this archive and the destination is an active workspace or a workspace in another archive. If an archive has been opened for writing (or creation) but not for reading, workspaces may be created, deleted, or edited, and workspace plane copies may be performed where the source workspace is an active workspace or a workspace in another archive and the destination is a workspace in this archive. If an archive is opened for both reading and writing (or creation), any workspace operation may be performed. When performing a workspace operation on an archived workspace, the workspaceIdentifier parameter is set to the Handle of the archive as indicated in the ArchiveOpenPDU, along with the entry name of the particular workspace within the archive.

Also to be included in the workspaceIdentifier for some workspace operations to archives is a modificationTime parameter. This parameter shall be included in the case that the archive is modified in any way. This includes workspace creation, editing, deletion, as well as workspace plane copies for the destination workspace. This parameter shall not be included in the workspaceIdentifier in the case of workspace creation acknowledge or ready indications, or for the source workspace in a workspace plane copy operation. When an archive is successfully modified in some way, the archive header used to identify the workspace is changed to include the new modification time as the archiveModificationTime parameter. If an archive modification fails for any reason, the modification time is not updated and the archive shall remain as it was before the failed operation was attempted. When the workspace is to be opened at a later time, the header used to reference it shall include the most recent modification time as the archiveModificationTime parameter. When opening a workspace, if the modification time indicated in the open request does not match that of the local archive at a SICE, the resulting result code in the ArchiveAcknowledgePDU shall include the actual header contents including the modification time. This may allow the requesting SICE to identify the differences between the available archive and the desired archive.

When the SICE which opened an archive has completed all operations on that archive, it may be closed by broadcasting an ArchiveClosePDU to all peer SICES. This is done in the manner described in Table 6-3. The content of the ArchiveClosePDU is shown in Table 8-63. The SICE shall send this PDU three separate times on each of the three priority channels: High, Medium, and Low. This is done to ensure that all archive modifications made prior to closing the archive are processed. At receiving SICES, only those operations resulting from PDUs received on a given priority channel prior to the ArchiveClosePDU on that channel are applied to the archive. This PDU is only valid if received from the SICE which opened the specified archive. From any other SICE, this PDU shall be ignored.

TABLE 8-63/T.126

ArchiveClosePDU

Parameter	Description
archiveHandle	Unique Handle used to reference the archive volume to be closed.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

If an error occurs in an archive as a result of an archive operation, receiving SICEs shall send an ArchiveErrorPDU to the node which originated the archive operation. This is done in the manner described in Table 6-3. The content of the ArchiveErrorPDU is shown in Table 8-64.

TABLE 8-64/T.126

ArchiveErrorPDU

Parameter	Description
archiveHandle	Unique Handle used to reference the archive volume to which this PDU refers.
entryName (Optional)	Specifies the archive entry associated with the error if applicable.
errorCode	This parameter indicates the type of error that occurred. It is either entryNotFound (when reading, modifying, or deleting an entry), entryExists (when creating an entry), storageExceeded, archiveNoLongerAvailable, unspecifiedError, or nonStandardError.
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

8.9 Conducted mode operation

When a conference is in conducted mode, the ability for a SICE to perform many of the operations described in clause 8 may be restricted by the conducting node. The GCC permission mechanism is used to determine whether these operations are allowed – if the node is given GCC conducted-mode permission, all restricted SI operations are allowed. If there is a peer SICE at the conducting node (i.e. one that has enrolled with the same Session Key), that SICE may also determine the ability to perform specific restricted operations by each other peer SICE. If there is no peer SICE at that node, or if no peer SICE at that node chooses to act as an arbiter of SI privileges, only the GCC permission mechanism is used. If there is more than one SICE at the conducting node which is capable of acting as a conducting SICE, it is up to that node locally to determine which of these will take on the conducting function.

When a conference switches to conducted mode, or if conductorship switches from one node to another, all SICEs are notified by receiving a GCC-Assign-Conductor indication from the GCC provider. This indication includes the GCC User ID of the node which has become the conductor. Each SICE may determine the User ID of the peer SICE at that node which has indicated itself as capable of arbitrating SI privileges. This is done by searching the current Application Roster for the entries corresponding to this GCC User ID and determining which, if any, of these has indicated itself as capable of arbitrating SI privileges. When a conference switches to non-conducted mode, all SICEs are notified by receiving a GCC-Conductor-Release indication from the GCC provider.

When a conference is first switched to conducted mode, or when a new conductor is assigned while already in conducted mode, no privileges are granted to any SICE to perform any restricted operation. While in conducted mode, if no GCC-Conductor-Permission-Grant indication has been received since the last change of conductorship or if the most recently received GCC-Conductor-Permission-Grant indication since the last change of conductorship had the Permission Flag set to FALSE, a SICE shall have no SI privileges unless specifically granted by the peer SICE at the conducting node. If the most recently received GCC-Conductor-Permission-Grant indication since the last change of conductorship had the Permission Flag set to TRUE, a SICE shall have all SI privileges granted.

If there is a peer SICE at the conducting node which has indicated itself as capable of arbitrating SI privileges, a SICE may request to be granted one or more privileges from the conductor. The following privileges may be requested:

- Privilege to create, edit or delete a workspace.
- Privilege to modify an annotation plane (a plane with the annotation usage designator set).
- Privilege to modify an image plane (a plane with the image usage designator set).
- Privilege to create, edit, or delete pointer.
- Privilege to send remote keyboard or pointing device events.
- Privilege to request remote printing.
- Privilege to open an archive for writing or creation.

A SICE may request particular privileges or, by not including a privilege list in the request, may request all available privileges. Privileges are requested from the conductor by sending a ConductorPrivilegeRequestPDU to the conducting node. This is done in the manner described in Table 6-3. The content of the ConductorPrivilegeRequestPDU is shown in Table 8-65.

NOTE – The presence of a particular privilege allows only those actions which would otherwise have been available in non-conducted mode based on the negotiated capability set.

On receipt of a ConductorPrivilegeRequestPDU, the SICE at the conducting node may grant some or all of the requested privileges by sending a ConductorPrivilegeGrantPDU to the requesting SICE. The ConductorPrivilegeGrantPDU includes the User ID of the SICE to which the privilege has been granted so that it can be broadcast to all other peer SICES to inform them of the privilege granted to the requesting SICE. If no privileges are granted (beyond those that the requesting SICE may already have) no ConductorPrivilegeGrantPDU need be sent. If the SICE at the conducting node receives a GCC-Application-Roster-Report indication from the GCC provider and if the report indicates that new nodes have been added to the list of enrolled SICES, the SICE at the conducting node shall rebroadcast the ConductorPrivilegeGrantPDU for each SICE which has some level of privilege. The SICE may at any time revoke some or all privileges or add privileges to any SICE by sending, unsolicited, a ConductorPrivilegeGrantPDU which indicates the new privilege list for the designated node in the same manner as used for granting privileges. The content of the ConductorPrivilegeGrantPDU is shown in Table 8-66.

Table 8-67 shows the effect of each of the privileges on the ability to transmit each SIPDU. The presence or absence of a privilege has no effect on the operation of receivers. If a SICE receives a PDU from another SICE which does not have the privilege to transmit that PDU, it shall process it as normal.

NOTE – The above restriction requiring receivers to process PDUs even if they were in violation of their conducted mode privilege is necessary to avoid race conditions during transitions when privileges are granted or removed.

TABLE 8-65/T.126

ConductorPrivilegeRequestPDU

Parameter	Description
privilegeList	<p>A set of values that indicate the privileges the requesting SICE wishes to be granted by the conductor.</p> <p>workspacePrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to create, edit, or delete workspaces (although deletion is allowed for the reason of insufficientStorage).</p> <p>annotationPrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to create, edit, or delete annotation bitmaps or drawing elements on workspace planes designated for annotation usage.</p> <p>imagePrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to create, edit, or delete image bitmaps on workspace planes designated for image usage.</p> <p>pointingPrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to create, edit, or delete pointers.</p> <p>remoteKeyEventPrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to send remote keyboard events. Note that permission from the workspace creator is still required before these events can be issued.</p> <p>remotePointingEventPrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to send remote pointing device events. Note that permission from the workspace creator is still required before these events can be issued.</p> <p>remotePrintingPrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to send remote printing requests.</p> <p>archiveCreateWritePrivilege This flag indicates that the requesting SICE wishes to be granted the privilege to open archives for creation or writing.</p> <p>nonStandardPrivilege This is a nonStandardIdentifier which represents a privilege understood by peer SICEs based on successful negotiation of the corresponding non-standard capability. The privilegeList may contain an arbitrary number of different nonStandardPrivileges.</p>
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-66/T.126

ConductorPrivilegeGrantPDU

Parameter	Description
destinationUserID	MCS User ID of the SICE to which the privileges are being granted or revoked.
privilegeList	<p>A set of values that indicate the privileges granted to a SICE by the conductor.</p> <p>workspacePrivilege This flag indicates that the destination SICE has been granted the privilege to create, edit, or delete workspaces (although deletion is allowed for the reason of insufficientStorage).</p> <p>annotationPrivilege This flag indicates that the destination SICE has been granted the privilege to create, edit, or delete annotation bitmaps or drawing elements on workspace planes designated for annotation usage.</p> <p>imagePrivilege This flag indicates that the destination SICE has been granted the privilege to create, edit, or delete image bitmaps on workspace planes designated for image usage.</p> <p>pointingPrivilege This flag indicates that the destination SICE has been granted the privilege to create, edit, or delete pointers.</p> <p>remoteKeyEventPrivilege This flag indicates that the destination SICE has been granted the privilege to send remote keyboard events. Note that permission from the workspace creator is still required before these events can be issued.</p> <p>remotePointingEventPrivilege This flag indicates that the destination SICE has been granted the privilege to send remote pointing device events. Note that permission from the workspace creator is still required before these events can be issued.</p> <p>remotePrintingPrivilege This flag indicates that the destination SICE has been granted the privilege to send remote printing requests.</p> <p>archiveCreateWritePrivilege This flag indicates that the destination SICE has been granted the privilege to open archives for creation or writing.</p> <p>nonStandardPrivilege This is a nonStandardIdentifier which represents a privilege understood by peer SICEs based on successful negotiation of the corresponding non-standard capability. The privilegeList may contain an arbitrary number of different nonStandardPrivileges.</p>
nonStandardParameters (Optional)	An optional list of non-standard parameters allowed only if the corresponding non-standard capabilities are present in the negotiated capability set.

TABLE 8-67/T.126

Conducted mode operation summary

SIPDU	Privilege required
ArchiveAcknowledgePDU	None required
ArchiveClosePDU	None required
ArchiveErrorPDU	None required
ArchiveOpenPDU	None required to open for reading. Archive Create or Write Privilege Flag to open for creation or writing
BitmapAbortPDU	Annotation Privilege, Image Privilege, or Pointer Privilege depending on the bitmap destination when sent by a bitmap transmitter None required when sent by a bitmap receiver
BitmapCheckpointPDU	Annotation Privilege, Image Privilege, or Pointer Privilege depending on the bitmap destination
BitmapCreatePDU	Annotation Privilege, Image Privilege, or Pointer Privilege depending on the bitmap destination
BitmapCreateContinuePDU	Annotation Privilege, Image Privilege, or Pointer Privilege depending on the bitmap destination
BitmapDeletePDU	Annotation Privilege, Image Privilege, or Pointer Privilege depending on the bitmap destination
BitmapEditPDU	Annotation Privilege, Image Privilege, or Pointer Privilege depending on the bitmap destination
ConductorPrivilegeGrantPDU	Only allowed by the conductor
ConductorPrivilegeRequestPDU	None required
DrawingCreatePDU	Annotation Privilege
DrawingDeletePDU	Annotation Privilege
DrawingEditPDU	Annotation Privilege
RemoteEventPermissionGrantPDU	None required
RemoteEventPermissionRequestPDU	Remote Keyboard or Pointing Device Privilege
RemoteKeyboardEventPDU	Remote Keyboard or Pointing Device Privilege
RemotePointingDeviceEventPDU	Remote Keyboard or Pointing Device Privilege
RemotePrintPDU	Remote Printing Privilege
SINonStandardPDU	Not defined by this Recommendation

TABLE 8-67/T.126 (concluded)

Conducted mode operation summary

SIPDU	Privilege required
WorkspaceCreatePDU	Workspace Privilege
WorkspaceCreateAcknowledgePDU	None required
WorkspaceDeletePDU	Workspace Privilege, although none required if reason is insufficientStorage
WorkspaceEditPDU	Workspace Privilege
WorkspacePlaneCopyPDU	Annotation Privilege, Image Privilege, or both, depending on the usage designator of the destination plane
WorkspaceReadPDU	None required
WorkspaceRefreshStatusPDU	None required

9 SIPDU definitions

Each SIPDU is transported as one MCSSDU across an MCS connection. A standard ASN.1 data value encoding is used to transfer SIPDUs between peer SICEs. For all PDUs, the ALIGNED variant of the Packed Encoding Rules of Recommendation X.691 shall be used.

-- Begin SI Definitions

SI-PROTOCOL DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- NOTE – All abstract types defined shall be exported

-- ArchiveEntryName

-- Name used to reference an archive entry

ArchiveEntryName ::= BMPString (SIZE (1..256))

-- ArchiveError

-- Specifies the cause of an error at a remote terminal during

-- a workspace archive operation

ArchiveError ::= CHOICE

```
{
  entryNotFound          NULL,
                        -- The terminal does not have the entry
                        -- that matches the archive name being
                        -- accessed for reading, editing, or
                        -- deletion
  entryExists            NULL,
                        -- The terminal already has an archive
                        -- entry that matches the name of the
                        -- archive entry name being created
  storageExceeded        NULL,
                        -- The terminal does not have sufficient
                        -- memory to store the requested
                        -- information
  archiveNoLongerAvailable NULL,
                        -- The archive indicated is no longer
                        -- available
  unspecifiedError        NULL,
                        -- A general error that is not
                        -- previously defined has occurred
  nonStandardError        NonStandardIdentifier,
                        -- Non standard error code
  ...
}
```

-- ArchiveHeader

-- This type specifies the parameters used to address archives

-- stored at remote terminals

ArchiveHeader ::= SEQUENCE

```
{
  archiveName            ArchiveName,
                        -- Name of the archive
  archiveCreationTime      GeneralizedTime,
                        -- Time and date of the creation of
                        -- the archive
  archiveModificationTime  GeneralizedTime,
                        -- Time and date of the most recent
                        -- modification of the archive
  ...
}
```

```

-- ArchiveMode
-- One of the following set of access modes must be indicated
-- when an archive is opened
ArchiveMode ::= SEQUENCE
{
  create      BOOLEAN,
              -- TRUE indicates that the archive shall be created. If
              -- an archive with the same name exists the operation
              -- should fail
  read        BOOLEAN,
              -- TRUE indicates that the archive shall be opened for
              -- reading only
  write       BOOLEAN,
              -- TRUE indicates that the archive shall be opened for
              -- writing
  ...
}

-- ArchiveName
-- Name used to reference an archive
ArchiveName ::= BMPString (SIZE (1..256))

-- ArchiveOpenResult
-- Specifies the result of an archive open request
ArchiveOpenResult ::= CHOICE
{
  archiveOpenSuccessful      NULL,
                              -- The requested archive was successfully
                              -- opened
  archiveNotFound            NULL,
                              -- An archive to be opened for reading
                              -- or writing was not found to exist
  archiveTimeIncorrect       ArchiveHeader,
                              -- An archive to be opened for reading
                              -- or writing was found, but with
                              -- incorrect creation or modification
                              -- time. The actual header is included
                              -- in the error response in this case
  archiveExists              NULL,
                              -- An archive to be opened for creation
                              -- already exists and will not be
                              -- overwritten
  archiveOpenForWriting      NULL,
                              -- An archive to be opened for writing
                              -- is already open for writing
  storageExceeded            NULL,
                              -- The terminal does not have sufficient
                              -- memory to store the requested
                              -- archive
  unspecifiedError           NULL,
                              -- An unspecified error has occurred
                              -- preventing the archive from being
                              -- opened
  nonStandardResult          NonStandardIdentifier,
                              -- Non-standard result code
  ...
}

-- BitmapAbortReason
-- These values represent the possible reason codes
-- for the BitmapAbortPDU

```

```

BitmapAbortReason ::= CHOICE
{
    unspecified                NULL,
                                -- Bitmap aborted for an unspecified
                                -- reason

    noResources               NULL,
                                -- Bitmap creation failed due to local
                                -- resource management problems

    outOfPaper                 NULL,
                                -- Bitmap creation failed because the
                                -- receiving terminal is out of paper

    nonStandardReason         NonStandardParameter,
    ...
}

-- BitmapAttribute
-- This CHOICE represents the list of possible bitmap attributes
BitmapAttribute ::= CHOICE
{
    viewState                 ViewState,
                                -- Indicates the state

    zOrder                    ZOrder,
                                -- Used to set the bitmap to the front
                                -- or back of the display list within an
                                -- addressable plane

    nonStandardAttribute     NonStandardParameter,
    ...
}

-- BitmapData
-- All or part of a bitmap bitstream
BitmapData ::= SEQUENCE
{
    dataCheckpoint           SEQUENCE (SIZE (1..100)) OF TokenID OPTIONAL,
                                -- Tokens to uninhibit when the corresponding
                                -- data is ready for display if checkpointing
                                -- is enabled for the exchange

    padBits                   INTEGER (1..256) OPTIONAL,
                                -- Count of bits at the end of the data octets that
                                -- are not part of the image bitstream and
                                -- are to be ignored

    data                       OCTET STRING (SIZE (1..8192)),
                                -- The compression-format-specific bitmap data
    ...
}

-- BitmapDestinationAddress
-- Destination address for bitmap exchanges
BitmapDestinationAddress ::= CHOICE
{
    hardCopyDevice            NULL,
    softCopyImagePlane        SoftCopyDataPlaneAddress,
    softCopyAnnotationPlane   SoftCopyDataPlaneAddress,
    softCopyPointerPlane     SoftCopyPointerPlaneAddress,
    ...
}

-- BitmapHeaderUncompressed
-- This type specifies the parameters of uncompressed bitmap
-- bitstreams
BitmapHeaderUncompressed ::= SEQUENCE
{
    colorMappingMode         CHOICE
    {

```

```

    directMap                SEQUENCE
    {
        colorSpace           ColorSpaceSpecifier,
        resolutionMode       ColorResolutionModeSpecifier
    },
    paletteMap               SEQUENCE
    {
        colorPalette         ColorPalette,
        bitsPerPixel         INTEGER (1 | 4 | 8)
    },
    ...
}

-- BitmapHeaderT4
-- Bitmap header for T.4 (G3) encoding
BitmapHeaderT4 ::= SEQUENCE
{
    twoDimensionalEncoding   BOOLEAN,
                            -- 2-D encoding if TRUE,
                            -- 1-D encoding if FALSE
    ...
}

-- BitmapHeaderT6
-- Bitmap header for T.6 (G4) encoding
BitmapHeaderT6 ::= SEQUENCE
{
    ...
}

-- BitmapHeaderT81
-- This type is used to specify the parameters necessary to
-- decode and display a T.81 (JPEG) image that are not specified
-- within the T.81 bitstream
BitmapHeaderT81 ::= SEQUENCE
{
    colorSpace               ColorSpaceSpecifier,
    resolutionMode           ColorResolutionModeSpecifier,
    ...
}

-- BitmapHeaderT82
-- This type is used to specify the parameters necessary to
-- decode and display a T.82 (JBIG) image that are not specified
-- within the T.82 bitstream
BitmapHeaderT82 ::= SEQUENCE
{
    colorMappingMode         CHOICE
    {
        directMap           ColorSpaceSpecifier,
                            -- Only greyscale and RGB colorspaces are
                            -- allowed
        paletteMap           SEQUENCE
        {
            bitmapPalette   ColorPalette,
            progressiveMode  CHOICE
            {
                progressivePalettes  SEQUENCE (SIZE (1..8)) OF ColorIndexTable,
                selfProgressive       NULL,
                ...
            } OPTIONAL
        }
    },
    ...
}

```

```

-- BitmapRegion
-- This type specifies a rectangular subregion within a bitmap
BitmapRegion ::= SEQUENCE
{
  upperLeft                SEQUENCE
  {
    xCoordinate            INTEGER (0..65535),
                          -- X component of a Cartesian address
    yCoordinate            INTEGER (0..65535)
                          -- Y component of a Cartesian address
  },
  lowerRight               SEQUENCE
  {
    xCoordinate            INTEGER (0..65535),
                          -- X component of a Cartesian address
    yCoordinate            INTEGER (0..65535)
                          -- Y component of a Cartesian address
  }
}

```

```

-- BitmapSize
-- The size of a bitmap in pixels
BitmapSize ::= SEQUENCE
{
  width                    INTEGER (1..65536),
                          -- The number of pixels horizontally
  height                   INTEGER (1..65536)
                          -- The number of pixels vertically
}

```

```

-- BoundingRectangle
-- Defines a rectangular region relative to an anchor point
BoundingRectangle ::= SEQUENCE
{
  upperLeft                PointDiff16,
  lowerRight              PointDiff16
}

```

```

-- ButtonEvent
-- Describes pointing device button events
ButtonEvent ::= CHOICE
{
  buttonUp                 NULL,
                          -- The button is up
  buttonDown               NULL,
                          -- The button is down
  buttonDoubleClick       NULL,
                          -- A button down event occurred
                          -- within the double click time window
  buttonTripleClick       NULL,
                          -- A button down event occurred
                          -- within the triple click time window
  buttonQuadClick         NULL,
                          -- A button down event occurred within
                          -- the quad click time window
  nonStandardButtonEvent   NonStandardIdentifier,
  ...
}

```

```

-- ColorAccuracyEnhancementCIELab
ColorAccuracyEnhancementCIELab ::= CHOICE
{
  predefinedCIELabSpace    CHOICE
  {

```

```

        nonStandardCIELabSpace      NonStandardParameter,
        ...
    },
    generalCIELabParameters          SEQUENCE
    {
        colorTemperature             INTEGER (0..MAX) OPTIONAL,
                                     -- Color temperature of the white point assumed by
                                     -- the color space (in degrees Kelvin)

        gamut                        SEQUENCE
        {
            lSpan                    INTEGER (-32768..32767),
                                     -- max L* - min L*

            lOffset                  INTEGER (-32768..32767),
                                     -- offset of the zero point for L

            aSpan                    INTEGER (-32768..32767),
                                     -- max a* - min a*

            aOffset                  INTEGER (-32768..32767),
                                     -- offset of the zero point for a

            bSpan                    INTEGER (-32768..32767),
                                     -- max b* - min b*

            bOffset                  INTEGER (-32768..32767),
                                     -- offset of the zero point for b
        } OPTIONAL,
        ...
    },
    ...
}

-- ColorAccuracyEnhancementGreyscale
ColorAccuracyEnhancementGreyscale ::= CHOICE
{
    predefinedGreyscaleSpace         CHOICE
    {
        nonStandardGreyscaleSpace    NonStandardParameter,
        ...
    },
    generalGreyscaleParameters       SEQUENCE
    {
        gamma                        REAL (0..MAX) OPTIONAL,
                                     -- Gamma value of the color space
        ...
    },
    ...
}

-- ColorAccuracyEnhancementRGB
ColorAccuracyEnhancementRGB ::= CHOICE
{
    predefinedRGBSpace               CHOICE
    {
        nonStandardRGBSpace          NonStandardParameter,
        ...
    },
    generalRGBParameters             SEQUENCE
    {
        gamma                        REAL (0..MAX) OPTIONAL,
                                     -- Gamma value of the color space

        colorTemperature             INTEGER (0..MAX) OPTIONAL,
                                     -- Color temperature of the white point assumed by
                                     -- the color space (in degrees Kelvin)

        primaries                    SEQUENCE
        {
            red                      ColorCIExyChromaticity,
                                     -- CIE xy chromaticity coordinate of the red
                                     -- primary

```

```

        green          ColorCIExyChromaticity,
                       -- CIE xy chromaticity coordinate of the green
                       -- primary
        blue          ColorCIExyChromaticity
                       -- CIE xy chromaticity coordinate of the blue
                       -- primary
    } OPTIONAL,
    ...
},
...
}

-- ColorAccuracyEnhancementYCbCr
ColorAccuracyEnhancementYCbCr ::= CHOICE
{
    predefinedYCbCrSpace          CHOICE
    {
        cCIR709                  NULL,
        nonStandardRGBSpace      NonStandardParameter,
        ...
    },
    generalYCbCrParameters       SEQUENCE
    {
        gamma                    REAL (0..MAX) OPTIONAL,
                                   -- Gamma value of the color space
        colorTemperature          INTEGER (0..MAX) OPTIONAL,
                                   -- Color temperature of the white point assumed by
                                   -- the color space (in degrees Kelvin)
        primaries                SEQUENCE
        {
            red                   ColorCIExyChromaticity,
                                   -- CIE xy chromaticity coordinate of the red
                                   -- primary
            green                 ColorCIExyChromaticity,
                                   -- CIE xy chromaticity coordinate of the green
                                   -- primary
            blue                  ColorCIExyChromaticity
                                   -- CIE xy chromaticity coordinate of the blue
                                   -- primary
        } OPTIONAL,
        ...
    },
    ...
}

-- ColorCIELab
-- Definition of a CIELab color
ColorCIELab ::= SEQUENCE
{
    l    INTEGER (0..255),
         -- Perceptually normalized luminance component
    a    INTEGER (0..255),
         -- One of two perceptually normalized chroma components
    b    INTEGER (0..255)
         -- One of two perceptually normalized chroma components
}

-- ColorCIExyChromaticity
-- Definition of a CIE normalized chromaticity value
ColorCIExyChromaticity ::= SEQUENCE
{
    x    REAL (0..one),
         -- CIE normalized x component
    y    REAL (0..one)
         -- CIE normalized y component
}

```

```

-- ColorIndexTable
-- This type is used to specify collections of color
-- values. All entries are references to absolute
-- color palette data.
ColorIndexTable ::= SEQUENCE (SIZE (1..256)) OF INTEGER (0..255)

-- ColorPalette
ColorPalette ::= SEQUENCE
{
  colorLookUpTable          CHOICE
  {
    paletteRGB              SEQUENCE
    {
      palette                SEQUENCE (SIZE (2..256)) OF ColorRGB,
      enhancement            ColorAccuracyEnhancementRGB OPTIONAL,
      ...
    },
    paletteCIELab           SEQUENCE
    {
      palette                SEQUENCE (SIZE (2..256)) OF ColorCIELab,
      enhancement            ColorAccuracyEnhancementCIELab OPTIONAL,
      ...
    },
    paletteYCbCr            SEQUENCE
    {
      palette                SEQUENCE (SIZE (2..256)) OF ColorYCbCr,
      enhancement            ColorAccuracyEnhancementYCbCr OPTIONAL,
      ...
    },
    nonStandardPalette      NonStandardParameter,
    ...
  },
  transparentEntry          INTEGER (0..255)          OPTIONAL,
  -- Index value of transparent color
  ...
}

-- ColorResolutionModeSpecifier
ColorResolutionModeSpecifier ::= CHOICE
{
  resolution4-4-4           NULL,
  -- Indicates single component
  -- 4:4:4
  resolution-4-2-2         NULL,
  -- 4:2:2 chrominance subsampling
  resolution-4-2-0         NULL,
  -- 4:2:0
  -- chrominance subsampling
  nonStandardResolutionMode NonStandardIdentifier,
  ...
}

-- ColorRGB
-- Definition of an RGB color
ColorRGB ::= SEQUENCE
{
  r      INTEGER (0..255),
  -- Red color component
  g      INTEGER (0..255),
  -- Green color component
  b      INTEGER (0..255)
  -- Blue color component
}

```

```

-- ColorSpaceSpecifier
ColorSpaceSpecifier ::= CHOICE
{
    greyscale                SEQUENCE
    {
        accuracyEnhancement  ColorAccuracyEnhancementGreyscale OPTIONAL
    },
    yCbCr                    SEQUENCE
    {
        accuracyEnhancement  ColorAccuracyEnhancementYCbCr OPTIONAL
    },
    rgb                      SEQUENCE
    {
        accuracyEnhancement  ColorAccuracyEnhancementRGB OPTIONAL
    },
    cieLab                   SEQUENCE
    {
        accuracyEnhancement  ColorAccuracyEnhancementCIELab OPTIONAL
    },
    nonStandardColorSpace    NonStandardIdentifier,
    ...
}

```

```

-- ColorYCbCr
-- Definition of a YCbCr color
ColorYCbCr ::= SEQUENCE
{
    y    INTEGER (0..255),
        -- Luminance component
    cb   INTEGER (0..255),
        -- Normalized blue minus luminance component
    cr   INTEGER (0..255)
        -- Normalized red minus luminance component
}

```

```

-- ConductorPrivilege
-- List of privileges that are awarded by the SICE at the conducting
-- node to other SICEs in the conference
ConductorPrivilege ::= CHOICE
{
    workspacePrivilege        NULL,
        -- Privilege to create, edit,
        -- or delete workspaces
    annotationPrivilege       NULL,
        -- Privilege to create, edit,
        -- or delete annotation bitmaps
        -- or drawing elements
    imagePrivilege            NULL,
        -- Privilege to create, edit,
        -- or delete image bitmaps
    pointingPrivilege         NULL,
        -- Privilege to create, edit,
        -- or delete pointers
    remoteKeyEventPrivilege   NULL,
        -- Privilege to send remote
        -- keyboard events
    remotePointingEventPrivilege NULL,
        -- Privilege to send pointing device
        -- events
    remotePrintingPrivilege   NULL,
        -- Privilege to request remote
        -- printing
}

```

```

archiveCreateWritePrivilege      NULL,
                                   -- Privilege to create or
                                   -- append an archive
nonStandardPrivilege             NonStandardIdentifier,
                                   -- non-standard privilege
                                   -- that was successfully
                                   -- negotiated
...
}

-- DataPlaneID
-- This is the identifier of a data plane within a workspace
DataPlaneID ::= INTEGER (0..255)

-- DrawingAttribute
-- The following drawingAttributes are used to specify visual and
-- behavioural properties of a drawing
DrawingAttribute ::= CHOICE
{
  penColor                        WorkspaceColor,
                                   -- Color of drawing pen
  fillColor                       WorkspaceColor,
                                   -- Color used to fill a closed region
  penThickness                     PenThickness,
                                   -- Width of pen
  penNib                           PenNib,
                                   -- Shape of pen nib
  lineStyle                        LineStyle,
                                   -- Style of line
  highlight                         BOOLEAN,
                                   -- Flag indicating whether the drawing
                                   -- element should be of a solid color
                                   -- or a highlight (semi-transparent)
  viewState                         ViewState,
                                   -- Indicates the visibility state
  zOrder                           ZOrder,
                                   -- Used to set the graphical element to
                                   -- the front or back of the display list
                                   -- within an addressable plane
  nonStandardAttribute             NonStandardParameter,
  ...
}

-- DrawingDestinationAddress
-- A DrawingDestinationAddress specifies the destination of drawing
-- elements
DrawingDestinationAddress ::= CHOICE
{
  softCopyAnnotationPlane         SoftCopyDataPlaneAddress,
  ...
}

-- DrawingType
-- A DrawingType specifies the shape of a drawn element
DrawingType ::= CHOICE
{
  point                           NULL,
                                   -- Unconnected points
  openPolyLine                     NULL,
                                   -- Points connected with straight lines.
                                   -- The last point is not connected to the
                                   -- first

```

```

closedPolyLine      NULL,
                    -- Points connected with straight lines.
                    -- The last point is connected to the
                    -- first
rectangle           NULL,
                    -- A rectangle defined by two corners
ellipse            NULL,
                    -- An ellipse
nonStandardDrawingType NonStandardIdentifier,
                    -- Negotiated non-standard type
...
}

-- Handle
-- Unique identifier that is used to address objects to allow edit
-- and or delete operations. These are obtained from GCC via the
-- GCC-Registry-Allocate-Handle request/confirm primitives
Handle ::= INTEGER (0..4294967295)

-- KeyCode
-- Character code that is contained in a RemoteKeyboardEventPDU.
-- This is either a two octet value that uses the UNICODE character
-- representation or special key specifier
KeyCode ::= CHOICE
{
  character          BMPString (SIZE (1)),
                    -- UNICODE character
  fkey              INTEGER (1..32),
                    -- Function key
                    -- Edit and navigation keys
  upArrow           NULL,
  downArrow         NULL,
  leftArrow         NULL,
  rightArrow        NULL,
  pageUp           NULL,
  pageDown          NULL,
  home              NULL,
  end               NULL,
  insert            NULL,
  delete            NULL,

  nonStandardKey    NonStandardIdentifier,
                    -- Non-standard key code
  ...
}

-- KeyModifier
-- Collection of keyboard modifiers
KeyModifier ::= CHOICE
{
  leftAlt           NULL,
                    -- Indicates the left ALT modifier key
                    -- is pressed
  rightAlt          NULL,
                    -- Indicates the right ALT modifier key
                    -- is pressed
  leftShift         NULL,
                    -- Indicates the left SHIFT modifier key
                    -- is pressed
  rightShift        NULL,
                    -- Indicates the right SHIFT modifier key
                    -- is pressed

```

```

leftControl          NULL,
                       -- Indicates the left CONTROL modifier key
                       -- is pressed
rightControl       NULL,
                       -- Indicates the right CONTROL modifier key
                       -- is pressed
leftSpecial        NULL,
                       -- Indicates the left SPECIAL modifier key
                       -- is pressed
rightSpecial       NULL,
                       -- Indicates the right SPECIAL modifier key
                       -- is pressed
numberPad          NULL,
                       -- Indicates the associated keystroke is
                       -- actuated by the numeric keypad
scrollLock         NULL,
                       -- Indicates that the scroll lock is active
nonStandardModifier NonStandardIdentifier,
                       -- Non-standard key modifier
...
}

```

```

-- KeyPressState
-- Set of events for a key, used as part of
-- a RemoteKeyboardEventPDU

```

```

KeyPressState ::= CHOICE

```

```

{
  none              NULL,
                       -- No key event is signaled. This is used
                       -- when only keyboard modifier keys are
                       -- changing state
  keyPress          NULL,
                       -- A key press event has occurred. Note that
                       -- multiple keyPress events may occur as
                       -- a result of a keyDown event and keyboard
                       -- auto-repeat.
  keyDown           NULL,
                       -- A key down transition has occurred. Note
                       -- that this implies a keyUp for an unmatched
                       -- previously received keyDown
  keyUp             NULL,
                       -- A key up transition has occurred
  nonStandardKeyPressState NonStandardIdentifier,
                       -- Non-standard key press state
  ...
}

```

```

-- LineStyle
-- The LineStyle attribute is used during a line draw procedure. It
-- specifies the type of line drawn.

```

```

LineStyle ::= CHOICE

```

```

{
  solid             NULL,
                       -- All pixels between endpoints are to be drawn
  dashed            NULL,
                       -- A dashed pattern is to be applied
  dotted           NULL,
                       -- A dotted pattern is to be applied
  dash-dot         NULL,
                       -- A dash-dot pattern is to be applied
  dash-dot-dot     NULL,
                       -- A dash-dot-dot pattern is to be applied
}

```

```

two-tone                                NULL,
-- Line color is to be applied to 50% of the
-- line width with a complimentary color
-- applied to either side. The width of
-- either side region is to be 25% of the
-- line width

nonStandardStyle                        NonStandardIdentifier,
...
}

-- MCSUserID
-- This type is used to specify MCS User IDs
MCSUserID ::= INTEGER (1001..65535)

-- H221NonStandardIdentifier
-- Used to specify non-standard objects using H.221 numbering.
-- The first four octets shall designate country code and
-- manufacturer code, assigned as specified in
-- Annex A/H.221 for NS-cap and NS-comm.
H221NonStandardIdentifier ::= OCTET STRING (SIZE (4..255))

-- NonStandardIdentifier
-- Unique identifier used to specify non-standard capabilities and
-- parameters either as an ASN.1 OBJECT IDENTIFIER or as an H.221
-- non-standard object
NonStandardIdentifier ::= CHOICE
{
  object                                OBJECT IDENTIFIER,
  h221nonStandard                        H221NonStandardIdentifier
}

-- NonStandardParameter
-- Used to specify non-standard parameters. This includes a
-- data field which may be used to fill in parameter values
-- of the type indicated by the NonStandardIdentifier
NonStandardParameter ::= SEQUENCE
{
  nonStandardIdentifier                  NonStandardIdentifier,
  data                                    OCTET STRING
}

-- one
-- This type provides a real value = 1 for use in this Recommendation
one REAL ::= {mantissa 1, base 2, exponent 0}

-- PenNib
-- This type specifies the shape of the nib of the pen that is
-- used to draw graphical elements
PenNib ::= CHOICE
{
  circular                                NULL,
-- A circle is used for the nib shape
  square                                    NULL,
-- A square is used for the nib shape
  nonStandardNib                            NonStandardIdentifier,
-- A non-standard pen nib
  ...
}

-- PenThickness
-- This type specifies the thickness of the pen that is used to
-- draw graphical elements
PenThickness ::= INTEGER (1..255)

```

```

-- PixelAspectRatio
-- This type specifies the horizontal to vertical ratio of
-- the size of a pixel
PixelAspectRatio ::= CHOICE
{
    square                NULL,
                        -- pixel aspect ratio is 1:1
    cif                   NULL,
                        -- pixel aspect ratio is 12:11 (hor:ver)
    fax1                  NULL,
                        -- 385:800 (hor:ver)
                        -- 8 lines/mm horizontally,
                        -- 3.85 lines/mm vertically
    fax2                  NULL,
                        -- 770:800 (hor:ver)
                        -- 8 lines/mm horizontally,
                        -- 7.7 lines/mm vertically
    general               SEQUENCE
                        -- The following two integers
                        -- specify a rational number that
                        -- is equivalent to a pixel's width
                        -- divided by a pixel's height
                        {
                            numerator      INTEGER (1..65535),
                            denominator    INTEGER (1..65535)
                        },
    nonStandardAspectRatio NonStandardIdentifier,
    ...
}

```

```

-- PlaneAttribute
-- Plane attributes are editable characteristics of workspace planes
PlaneAttribute ::= CHOICE

```

```

{
    protection            PlaneProtection,
                        -- Access restrictions for a plane
    nonStandardAttribute NonStandardParameter,
                        -- Non-standard attribute
    ...
}

```

```

-- PlaneProtection
-- This enumeration identifies the possible access restrictions
-- that can be imposed on a workspace plane
PlaneProtection ::= SEQUENCE

```

```

{
    creatorProtected     BOOLEAN,
                        -- Only the workspace
                        -- creator can submit data to the plane
    ...
}

```

```

-- PlaneUsage
-- This type specifies the usage of a single plane in a workspace
PlaneUsage ::= CHOICE

```

```

{
    annotation           NULL,
                        -- The plane is designated to contain
                        -- annotation data
    image                NULL,
                        -- The plane is designated to contain
                        -- image data
}

```

```

nonStandardPlaneUsage          NonStandardIdentifier,
                                -- The plane is designated to contain
                                -- non-standard plane data
...
}

-- PointList
-- A list of points to define a drawing object using one of
-- three possible encodings depending on how far any point
-- in the list strays from the anchor point
PointList ::= CHOICE
{
  pointsDiff4      SEQUENCE (SIZE (0..255)) OF PointDiff4,
  pointsDiff8      SEQUENCE (SIZE (0..255)) OF PointDiff8,
  pointsDiff16     SEQUENCE (SIZE (0..255)) OF PointDiff16
}

-- PointListEdits
-- A list of points to edit a drawing object using one of
-- three possible encodings depending on how far any point
-- in the list strays from the anchor point
PointListEdits ::= SEQUENCE SIZE (1..255) OF SEQUENCE
{
  initialIndex          INTEGER (0..254),
                        -- Index of the first (or only) point
                        -- to edit
  initialPointEdit      PointDiff16,
                        -- Position of the point specified
                        -- relative to the anchor point
  subsequentPointEdits PointList OPTIONAL,
                        -- Points specified relative to the
                        -- previous point in this list (the
                        -- first one specified relative to
                        -- the initialPointEdit).
                        -- When this list is used, successive
                        -- points are assumed to have indices
                        -- sequentially following the
                        -- initialPointIndex
  ...}

-- PointDiff4
-- A point specified differentially relative to an anchor point
-- with a range from -8 to +7
PointDiff4 ::= SEQUENCE
{
  xCoordinate          INTEGER (-8..7),
                        -- X component of a Cartesian address
  yCoordinate          INTEGER (-8..7)
                        -- Y component of a Cartesian address
}

-- PointDiff8
-- A point specified differentially relative to an anchor point
-- with a range from -128 to +127
PointDiff8 ::= SEQUENCE
{
  xCoordinate          INTEGER (-128..127),
                        -- X component of a Cartesian address
  yCoordinate          INTEGER (-128..127)
                        -- Y component of a Cartesian address
}

```

```

-- PointDiff16
-- A point specified differentially relative to an anchor point
-- with a range from -32768 to +32767
PointDiff16 ::= SEQUENCE
{
    xCoordinate                INTEGER (-32768..32767),
                                -- X component of a Cartesian address
    yCoordinate                INTEGER (-32768..32767)
                                -- Y component of a Cartesian address
}

-- RemoteEventDestinationAddress
-- A RemoteEventDestinationAddress specifies the destination of
-- a remote event
RemoteEventDestinationAddress ::= CHOICE
{
    softCopyWorkspace          Handle,
    ...
}

-- RemoteEventPermission
-- Choice of remote events that can be issued to a workspace
RemoteEventPermission ::= CHOICE
{
    keyboardEvent              NULL,
    pointingDeviceEvent        NULL,
    nonStandardEvent          NonStandardIdentifier,
    ...
}

-- RotationSpecifier
-- Specifies a rotation angle and an axis of revolution
RotationSpecifier ::= SEQUENCE
{
    rotationAngle              INTEGER (0..21599),
                                -- 0 degrees to 359 degrees 59 minutes
                                -- in units of minutes of arc
    rotationAxis                PointDiff16
                                -- Workspace location relative to an
                                -- object's anchor point
}

-- SoftCopyDataPlaneAddress
-- Address of a workspace data plane.
SoftCopyDataPlaneAddress ::= SEQUENCE
{
    workspaceHandle            Handle,
    plane                       DataPlaneID
}

-- SoftCopyPointerPlaneAddress
-- Address of a workspace pointer plane
SoftCopyPointerPlaneAddress ::= SEQUENCE
{
    workspaceHandle            Handle
}

-- SourceDisplayIndicator
-- Indicator of the size and location of a workspace view within
-- the display device of the sourcing terminal
SourceDisplayIndicator ::= SEQUENCE
{
    displayAspectRatio        REAL (0..MAX),
                                -- Aspect ratio of the display;
                                -- horizontal over vertical size.
                                -- Positive real values
}

```

```

horizontalSizeRatio          REAL (0..MAX),
                               -- Ratio of workspace view horizontal
                               -- dimension to display horizontal
                               -- dimension.
                               -- Positive real values

horizontalPosition         REAL,
                               -- Horizontal offset of upper left
                               -- corner of the workspace view from the
                               -- upper left corner of the display
                               -- normalized to the display width
                               -- (where the display spans the
                               -- horizontal range 0.0 to 1.0)

verticalPosition          REAL,
                               -- Vertical offset of upper left
                               -- corner of the workspace view from the
                               -- upper left corner of the display
                               -- normalized to the display height
                               -- (where the display spans the
                               -- vertical range 0.0 to 1.0)

...
}
-- TokenID
-- MCS Token ID
TokenID ::= INTEGER (1..65535)

-- ViewState
-- Controls the visibility state of an object
ViewState ::= CHOICE
{
  unselected                NULL,
  selected                  NULL,
  hidden                    NULL,
  nonStandardViewState     NonStandardIdentifier,
  ...
}

-- WorkspaceAttribute
-- Workspace attributes are editable characteristics of workspace
WorkspaceAttribute ::= CHOICE
{
  backgroundColor          WorkspaceColor,
                               -- This specifies the background
                               -- color of the workspace

  preserve                  BOOLEAN,
                               -- If TRUE, the associated workspace
                               -- resource should not be placed on
                               -- the viewed workspace queue once
                               -- it has been automatically removed
                               -- from the Focus state

  nonStandardAttribute     NonStandardParameter,
  ...
}

-- WorkspaceColor
-- The following defines a generic type for a color, used where a
-- color is required for drawing or workspace backgrounds
WorkspaceColor ::= CHOICE
{
  workspacePaletteIndex    INTEGER (0..255),
  rgbTrueColor             ColorRGB,
  transparent              NULL,
  ...
}

```

```

-- WorkspaceCoordinate
-- A WorkspaceCoordinate is the value of a single axis of
-- a point in a workspace
WorkspaceCoordinate ::= INTEGER (-21845..43690)

-- WorkspaceDeleteReason
-- This value represents the reason codes for the
-- WorkspaceDeletePDU
WorkspaceDeleteReason ::= CHOICE
{
    userInitiated          NULL,
                          -- Workspace deletion initiated by user
    insufficientStorage    NULL,
                          -- Workspace deleted due to insufficient
                          -- storage capacity
    nonStandardReason      NonStandardParameter,
    ...
}

-- WorkspaceIdentifier
WorkspaceIdentifier ::= CHOICE
{
    activeWorkspace        Handle,
                          -- Handle identifying the active workspace
    archiveWorkspace       SEQUENCE
    archiveHandle          Handle,
                          -- Handle identifying the archive in which the
                          -- archived workspace is contained
    entryName              ArchiveEntryName,
                          -- Name of the archived workspace
    modificationTime       GeneralizedTime OPTIONAL
                          -- If the workspace identifier is being
                          -- used for an operation in which the
                          -- workspace is to be modified, this parameter
                          -- shall indicate the time of modification.
                          -- In this case, the archive header is modified
                          -- to reflect the most recent modification time.
                          -- Otherwise, this parameter shall not
                          -- be included

    },
    ...
}

-- WorkspacePoint
-- A WorkspacePoint is a two dimensional address of a location in a
-- workspace plane including points in the invisible border areas
WorkspacePoint ::= SEQUENCE
{
    xCoordinate            WorkspaceCoordinate,
                          -- X component of a Cartesian address
    yCoordinate            WorkspaceCoordinate
                          -- Y component of a Cartesian address
}

-- WorkspaceRegion
-- This type can be used to describe both the size and
-- position of a rectangular region within a workspace
WorkspaceRegion ::= SEQUENCE
{
    upperLeft              WorkspacePoint,
    lowerRight             WorkspacePoint
}

```

-- *WorkspaceSize*
-- *The size of a workspace in pixels*

WorkspaceSize ::= SEQUENCE

```
{  
  width          INTEGER (1..21845),  
                -- The number of pixels horizontally  
  height         INTEGER (1..21845)  
                -- The number of pixels vertically  
}
```

-- *WorkspaceViewAttribute*
-- *View attributes are editable characteristics of workspace views*

WorkspaceViewAttribute ::= CHOICE

```
{  
  viewRegion     CHOICE  
  {  
    fullWorkspace NULL,  
                -- View the entire workspace  
    partialWorkspace WorkspaceRegion  
                -- Rectangle defining the region of the  
                -- workspace to view. The view shall  
                -- not extend beyond the boundaries of  
                -- the workspace  
  },  
  viewState      WorkspaceViewState,  
                -- Visibility state of the view  
  updatesEnabled BOOLEAN,  
                -- If this attribute is set to FALSE  
                -- (the default is TRUE), it is an indication that  
                -- subsequent updates to the workspace corresponding  
                -- to this view not be shown until this attribute  
                -- is set to TRUE  
  sourceDisplayIndicator SourceDisplayIndicator,  
                -- Indicates the characteristics of  
                -- the view within the source  
                -- display device  
  nonStandardAttribute NonStandardParameter,  
  ...  
}
```

-- *WorkspaceViewState*
-- *A view's state indicates how the local terminal should*
-- *display the view*

WorkspaceViewState ::= CHOICE

```
{  
  hidden         NULL,  
                -- This workspace should not be  
                -- shown to the user  
  background     NULL,  
                -- The display of this workspace is  
                -- optional  
  foreground     NULL,  
                -- The display of this workspace is  
                -- desirable  
  focus          NULL,  
                -- The display of this workspace is  
                -- mandatory. Only one workspace may  
                -- be set to this state  
  nonStandardState NonStandardIdentifier,  
  ...  
}
```

```

-- ZOrder
-- This enumerated type is used to specify a transition to front
-- or back of an object within an addressable plane
ZOrder ::= ENUMERATED
{
  front                (0), -- Move object to the front of the plane display list
  back                 (1), -- Move object to the back of the plane display list
  ...
}

-- Begin SI PDU Definitions

-- ArchiveAcknowledgePDU
-- The ArchiveAcknowledgePDU is used to acknowledge that an archive
-- has been successfully opened
ArchiveAcknowledgePDU ::= SEQUENCE
{
  archiveHandle       Handle, -- Unique handle that references the archive
  result              ArchiveOpenResult, -- Indicates whether or not the
  -- archive was opened successfully
  nonStandardParameters SET OF NonStandardParameter OPTIONAL, -- Allowed only if the corresponding
  -- non-standard capabilities are present
  -- in the negotiated capability set
  ...
}

-- ArchiveClosePDU
-- The ArchiveClosePDU is used to close an archive that was
-- previously opened during an SI conference
ArchiveClosePDU ::= SEQUENCE
{
  archiveHandle       Handle, -- Unique handle that references the archive
  nonStandardParameters SET OF NonStandardParameter OPTIONAL, -- Allowed only if the corresponding
  -- non-standard capabilities are present
  -- in the negotiated capability set
  ...
}

-- ArchiveErrorPDU
-- The ArchiveErrorPDU is used by a terminal receiving an
-- archive PDU to signal error conditions to the sender
ArchiveErrorPDU ::= SEQUENCE
{
  archiveHandle       Handle, -- Unique handle that references the archive
  entryName          ArchiveEntryName OPTIONAL, -- Specifies the archive entry associated with
  -- the error if applicable
  errorCode          ArchiveError, -- Specifies the cause of the error at the
  -- remote terminal
  nonStandardParameters SET OF NonStandardParameter OPTIONAL, -- Allowed only if the corresponding
  -- non-standard capabilities are present
  -- in the negotiated capability set
  ...
}

```

```

-- ArchiveOpenPDU
-- The ArchiveOpenPDU is used to open an archive at a
-- remote terminal that supports this capability
ArchiveOpenPDU ::= SEQUENCE
{
    archiveHandle          Handle,
                           -- Unique handle that is used to reference
                           -- this archive during the conference
    mode                   ArchiveMode,
                           -- Indicates the access restrictions placed
                           -- on the archive
    header                 ArchiveHeader,
                           -- Specifies information used to identify
                           -- the archive. If the archive
                           -- is being created, this is the information
                           -- that is used to identify the archive in
                           -- the future
    maxEntries            INTEGER (1..65535) OPTIONAL,
                           -- This parameter allows remote terminals
                           -- to estimate the local resource usage for
                           -- the specified archive so they can
                           -- signal an error early in the archiving
                           -- process. It is only to be specified if
                           -- the archive open mode is set to "create"
    nonStandardParameters SET OF NonStandardParameter OPTIONAL,
                           -- Allowed only if the corresponding
                           -- non-standard capabilities are present
                           -- in the negotiated capability set
    ...
}

-- BitmapAbortPDU
-- This PDU is used by both the transmitting SICE to signal that a
-- bitmap exchange is being aborted and by a SICE requesting that
-- a bitmap exchange in progress be aborted
BitmapAbortPDU ::= SEQUENCE
{
    bitmapHandle          Handle,
                           -- Handle referring to the bitmap being created
    userID                MCSUserID OPTIONAL,
                           -- Optionally provided by the transmitter
                           -- if identification of the source of
                           -- the abort is desired
    reason                BitmapAbortReason OPTIONAL,
    message               BMPString (SIZE (1..256)) OPTIONAL,
    nonStandardParameters SET OF NonStandardParameter OPTIONAL,
                           -- Allowed only if the corresponding
                           -- non-standard capabilities are present
                           -- in the negotiated capability set
    ...
}

-- BitmapCheckpointPDU
-- This PDU is used by a terminal that is transmitting a bitmap
-- when it wants to notify receiving terminals that they should
-- display previously received data
BitmapCheckpointPDU ::= SEQUENCE
{
    bitmapHandle          Handle,
                           -- Handle used to reference this bitmap

```

```

passedCheckpoints      SET (SIZE (1..100)) OF TokenID,
                        -- List of checkpoints that have been
                        -- uninhibited by all nodes

percentComplete        INTEGER (1..100),
                        -- Cumulative portion of the bitmap
                        -- completed as a result of all
                        -- passed checkpoints so far

nonStandardParameters SET OF NonStandardParameter OPTIONAL,
                        -- Allowed only if the corresponding
                        -- non-standard capabilities are present
                        -- in the negotiated capability set

...
}

-- BitmapCreatePDU
-- This PDU is used to initiate a bitmap transmission
BitmapCreatePDU ::= SEQUENCE
{
  bitmapHandle          Handle,
                        -- Handle to be used to reference this object
                        -- in the future

  destinationAddress    BitmapDestinationAddress,
                        -- Destination address of the bitmap

  attributes            SET OF BitmapAttribute OPTIONAL,
                        -- List of editable attributes
                        -- of the bitmap

  anchorPoint          WorkspacePoint OPTIONAL,
                        -- Point of origin of the bitmap with
                        -- respect to the destination workspace.
                        -- Only needed for softcopy bitmaps.
                        -- Default is (0,0)

  bitmapSize           BitmapSize,
                        -- Width and height of the total bitmap
                        -- represented in the bitstream. For a
                        -- multi-component bitmap, this is the
                        -- size of the largest component

  bitmapRegionOfInterest BitmapRegion OPTIONAL,
                        -- Region of interest within the bitmap
                        -- to be applied to the workspace
                        -- Default is full bitmap

  pixelAspectRatio     PixelAspectRatio,
                        -- Pixel aspect ratio of the bitmap

  scaling              PointDiff16 OPTIONAL,
                        -- Offset in workspace coordinates
                        -- of the lower right hand corner of
                        -- the bitmap relative to the anchor point.
                        -- Default is no scaling.
                        -- Only needed for softcopy bitmaps

  boundingRectangle    BoundingBox OPTIONAL,
                        -- Bounding rectangle of bitmap
                        -- specified relative to the anchor point.
                        -- Used in determining the extent of the bitmap
                        -- when a workspace region is being copied.
                        -- This parameter is mandatory if a non-one-to-one
                        -- pixel aspect ratio is specified for a bitmap
                        -- on an editable workspace plane

  checkpoints          SEQUENCE (SIZE (1..100)) OF TokenID OPTIONAL,
                        -- Tokens to be used for checkpointing the
                        -- bitmap create exchange

```

```

bitmapFormatHeader
{
  bitmapHeaderUncompressed
  bitmapHeaderT4
  bitmapHeaderT6
  bitmapHeaderT81
  bitmapHeaderT82
  bitmapHeaderNonStandard
  ...
},
bitmapData
moreToFollow
nonStandardParameters
...
}

-- BitmapCreateContinuePDU
-- This PDU is used by the transmitting SICE to continue
-- a bitmap transmission begun by a BitmapCreatePDU
BitmapCreateContinuePDU ::= SEQUENCE
{
  bitmapHandle
  bitmapData
  moreToFollow

```

CHOICE

- *The following headers provide image*
- *bitstream parameters that*
- *are outside the scope of the corresponding*
- *coding standard but*
- *are necessary for image decompression.*
- *NOTE – Some bitmap formats*
- *are disallowed depending on the value of*
- *the destinationAddress parameter.*

BitmapHeaderUncompressed,

- *Parameters for the*
- *uncompressed pixel*
- *representation*

BitmapHeaderT4,

- *Parameters for T4 (G3)*
- *encoded bitstreams outside*
- *the T.4 standard's scope*

BitmapHeaderT6,

- *Parameters for T6 (G4)*
- *encoded bitstreams outside*
- *the T.6 standard's scope*

BitmapHeaderT81,

- *Parameters for T.81 (JPEG)*
- *encoded bitstreams outside*
- *the T.81 standard's scope*

BitmapHeaderT82,

- *Parameters for T.82 (JBIG)*
- *encoded bitstreams outside*
- *the T.82 standard's scope*

NonStandardParameter,

BitmapData OPTIONAL,

- *Compression format specific*
- *bitmap data*
- *padded to be byte aligned*

BOOLEAN,

- *Indicates whether or not this is the last*
- *block of data for the bitmap*

SET OF NonStandardParameter OPTIONAL,

- *Allowed only if the corresponding*
- *non-standard capabilities are present*
- *in the negotiated capability set*

Handle,

- *Handle referring to the bitmap being created*

BitmapData,

- *Bitmap data*

BOOLEAN,

- *Indicates whether or not this is the last*
- *block of data for the bitmap*

```

nonStandardParameters          SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

-- BitmapDeletePDU
-- This PDU is used to delete bitmaps.
BitmapDeletePDU ::= SEQUENCE
{
    bitmapHandle                Handle,
                                -- Handle used to reference this bitmap
    nonStandardParameters      SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
    ...
}

-- BitmapEditPDU
-- This PDU is used to change bitmap attributes
BitmapEditPDU ::= SEQUENCE
{
    bitmapHandle                Handle,
                                -- Handle used to reference this bitmap
    attributeEdits              SET OF BitmapAttribute OPTIONAL,
                                -- List of attributes to be edited
    anchorPointEdit            WorkspacePoint OPTIONAL,
                                -- Point of origin of the bitmap with
                                -- respect to the destination workspace.
    bitmapRegionOfInterestEdit BitmapRegion OPTIONAL,
                                -- Region of interest within the bitmap
                                -- to be applied to the workspace
    scalingEdit                PointDiff16 OPTIONAL,
                                -- Offset in workspace coordinates
                                -- of the lower right hand corner of
                                -- the bitmap relative to the anchor point
    boundingRectangleEdit      BoundingBox OPTIONAL,
                                -- Bounding rectangle of bitmap
                                -- specified relative to the anchor point.
                                -- Used in determining the extent of the bitmap
                                -- when a workspace region is being copied
    nonStandardParameters      SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
    ...
}

-- ConductorPrivilegeGrantPDU
-- This PDU is used by the conductor to grant or revoke privileges
-- when the conference is in conducted mode.
ConductorPrivilegeGrantPDU ::= SEQUENCE
{
    destinationUserID          MCSUserID,
                                -- MCS User ID of the destination node
    privilegeList              SET OF ConductorPrivilege,
                                -- A particular privilege shall appear
                                -- in this list no more than once

```

```

nonStandardParameters          SET OF NonStandardParameter OPTIONAL,
-- Allowed only if the corresponding
-- non-standard capabilities are present
-- in the negotiated capability set
...
}

-- ConductorPrivilegeRequestPDU
-- This PDU is used to request privileges from the conductor
-- when the conference is in conducted mode
ConductorPrivilegeRequestPDU ::= SEQUENCE
{
  privilegeList                SET OF ConductorPrivilege,
-- A particular privilege shall appear
-- in this list no more than once

  nonStandardParameters       SET OF NonStandardParameter OPTIONAL,
-- Allowed only if the corresponding
-- non-standard capabilities are present
-- in the negotiated capability set
  ...
}

-- DrawingCreatePDU
-- A drawingCreate PDU is used to deposit one or more
-- drawing elements to a workspace plane
DrawingCreatePDU ::= SEQUENCE
{
  drawingHandle               Handle OPTIONAL,
-- Handle to be used to reference this
-- drawing object in future exchanges. Note
-- that editing and deleting objects is
-- only valid if the target plane is
-- of type "editable"

  destinationAddress          DrawingDestinationAddress,
-- Destination of drawing

  drawingType                 DrawingType,
-- Which basic drawing shape this element
-- represents

  attributes                  SET OF DrawingAttribute OPTIONAL,
-- Attributes of the drawing object.
-- NOTE – All attributes have default
-- values that are assumed if the
-- attribute is not specified

  anchorPoint                WorkspacePoint,
-- Point of origin of the drawing element.
-- This forms the first of the control points
-- and is the point from which all other control
-- points are defined relative to

  rotation                   RotationSpecifier OPTIONAL,
-- Specifies a rotation angle and
-- point of revolution for the drawing element

  boundingRectangle          BoundingRectangle OPTIONAL,
-- Bounding rectangle of drawing element
-- specified relative to the anchor point.
-- Used in determining the extent of the object
-- when a workspace region is being copied.
-- This parameter is mandatory if a non-zero
-- rotation is specified for a drawing element
-- on an editable workspace plane

```

```

sampleRate                INTEGER (1..255) OPTIONAL,
                             -- For applicable types, this indicates
                             -- the rate at which points were acquired
                             -- by the transmitting terminal (in samples
                             -- per second) so they can be
                             -- replayed at a similar rate if desired

pointList                 PointList,
                             -- List of control points that define the
                             -- drawing shape. The interpretation
                             -- of the control point list is dependent on
                             -- the value of the "type" parameter.
                             -- Note that the control points in the list
                             -- are differentially encoded from
                             -- the previous

nonStandardParameters    SET OF NonStandardParameter OPTIONAL,
                             -- Allowed only if the corresponding
                             -- non-standard capabilities are present
                             -- in the negotiated capability set
...
}

-- DrawingDeletePDU
-- A DrawingDeletePDU is used to delete one
-- graphical element from a workspace plane
DrawingDeletePDU ::= SEQUENCE
{
  drawingHandle           Handle,
                             -- Drawing object to delete

  nonStandardParameters  SET OF NonStandardParameter OPTIONAL,
                             -- Allowed only if the corresponding
                             -- non-standard capabilities are present
                             -- in the negotiated capability set
...
}

-- DrawingEditPDU
-- A DrawingEditPDU is used to alter one or more of
-- a drawing elements attributes or parameters
DrawingEditPDU ::= SEQUENCE
{
  drawingHandle           Handle,
                             -- Identifier of item to be edited

  attributeEdits         SET OF DrawingAttribute OPTIONAL,
                             -- List of attribute changes

  anchorPointEdit       WorkspacePoint OPTIONAL,
                             -- Point of origin of the drawing element

  rotationEdit          RotationSpecifier OPTIONAL,
                             -- Specifies a rotation angle and
                             -- point of revolution for the drawing element

  boundingRectangleEdit BoundingRectangle OPTIONAL,
                             -- Bounding rectangle of drawing element
                             -- specified relative to the anchor point.
                             -- Used in determining the extent of the object
                             -- when a workspace region is being copied

  pointListEdits       PointListEdits OPTIONAL,
                             -- List of control point changes.
                             -- Note that the index refers to
                             -- the point list not including
                             -- the anchor point

```

```

nonStandardParameters          SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

-- RemoteEventPermissionGrantPDU
-- This PDU is used to grant permission to issue remote
-- events
RemoteEventPermissionGrantPDU ::= SEQUENCE
{
  destinationAddress          RemoteEventDestinationAddress,
                                -- Address to which remote event
                                -- permission is being granted

  destinationUserID          MCSUserID,
                                -- MCS User ID of the destination node

  remoteEventPermissionList  SET OF RemoteEventPermission,
                                -- A particular permission shall not be
                                -- included in this list more than once

  nonStandardParameters      SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

-- RemoteEventPermissionRequestPDU
-- This PDU is used to request permission to issue remote
-- events from the workspace creator
RemoteEventPermissionRequestPDU ::= SEQUENCE
{
  destinationAddress          RemoteEventDestinationAddress,
                                -- Address to which remote event
                                -- permission is being requested

  remoteEventPermissionList  SET OF RemoteEventPermission,
                                -- A particular permission shall not be
                                -- included in this list more than once

  nonStandardParameters      SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

-- RemoteKeyboardEventPDU
-- This PDU signals keyboard event
RemoteKeyboardEventPDU ::= SEQUENCE
{
  destinationAddress          RemoteEventDestinationAddress,
                                -- Destination address of remote event

  keyModifierStates          SET OF KeyModifier OPTIONAL,
                                -- Set of key modifiers. Only modifiers
                                -- in this list are assumed to be active.
                                -- A particular key modifier shall not be
                                -- included in this set more than once

  keyPressState              KeyPressState,
                                -- This item specifies keyboard event that
                                -- is being signalled

  keyCode                    KeyCode,
                                -- Character corresponding to the pressed
                                -- key or function key
}

```

```

nonStandardParameters          SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

-- RemotePointingDeviceEventPDU
-- This PDU is used to signal pointing device events
RemotePointingDeviceEventPDU ::= SEQUENCE
{
  destinationAddress          RemoteEventDestinationAddress,
                                -- Destination address of remote event
  leftButtonState            ButtonEvent,
                                -- This specifies the left button state
  middleButtonState         ButtonEvent,
                                -- This specifies the middle button state
  rightButtonState         ButtonEvent,
                                -- This specifies the right button state
  initialPoint             WorkspacePoint,
                                -- This specifies the initial pointing device
                                -- position
  sampleRate               INTEGER (1..255) OPTIONAL,
                                -- This parameter indicates the rate at which
                                -- points were acquired by the transmitting
                                -- terminal (in samples per second) so they
                                -- can be replayed at a similar rate if desired
  pointList                 PointList OPTIONAL,
                                -- Additional coordinates that are
                                -- each differentially encoded with respect
                                -- to the initialPoint parameter
  nonStandardParameters     SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

-- RemotePrintPDU
-- This PDU is used to instruct a remote terminal to print
-- the specified workspace.
RemotePrintPDU ::= SEQUENCE
{
  destinationAddress       RemoteEventDestinationAddress,
                                -- Destination address of remote event
  numberOfCopies         INTEGER (1..65536) OPTIONAL,
                                -- number of copies to be printed
  portrait               BOOLEAN OPTIONAL,
                                -- TRUE specifies a portrait paper
                                -- orientation
                                -- FALSE specifies a landscape paper
                                -- orientation
  regionOfInterest       WorkspaceRegion OPTIONAL,
                                -- Optionally defines rectangular
                                -- region of interest within the
                                -- workspace to be printed.
                                -- If not present, it is implied that
                                -- the entire workspace is to be printed..
  nonStandardParameters   SET OF NonStandardParameter OPTIONAL,
                                -- Allowed only if the corresponding
                                -- non-standard capabilities are present
                                -- in the negotiated capability set
...
}

```

```

-- SINonStandardPDU
-- This PDU allows any non-standard information to be transmitted.
SINonStandardPDU ::= SEQUENCE
{
    nonStandardTransaction          NonStandardParameter,
    ...
}

-- WorkspaceCreatePDU
-- This PDU causes a workspace to be created and its
-- attributes to be set
WorkspaceCreatePDU ::= SEQUENCE
{
    workspaceIdentifier             WorkspaceIdentifier,
                                     -- Identifier that will be used to reference
                                     -- this workspace in the future
    appRosterInstance              INTEGER (0..65535),
                                     -- Indicates which application roster
                                     -- instance (returned in the
                                     -- GCC-Application-Roster-Report
                                     -- indication) was valid when this PDU
                                     -- was issued. This is used to eliminate
                                     -- race conditions that can occur when
                                     -- terminals enter a conference while
                                     -- a workspace is being created
    synchronized                   BOOLEAN,
                                     -- TRUE specifies that the
                                     -- workspace contents
                                     -- stacking order must be consistent
                                     -- everywhere.
                                     -- In many cases, this implies
                                     -- the use of
                                     -- MCS-UNIFORM-SEND-DATA for SIPDU
                                     -- submission.
                                     -- FALSE specifies that the
                                     -- workspace contents
                                     -- do not have to be consistent
                                     -- in stacking
                                     -- order therefore the use of
                                     -- MCS-SEND-DATA is
                                     -- acceptable for all content
                                     -- submitting transactions
    acceptKeyboardEvents           BOOLEAN,
                                     -- If TRUE this workspace can accept
                                     -- remote keyboard events
    acceptPointingDeviceEvents     BOOLEAN,
                                     -- If TRUE this workspace can accept
                                     -- remote pointer device events
    workspaceSize                  WorkspaceSize,
                                     -- This value specifies the width and
                                     -- height of the new workspace in pixels
    workspaceAttributes            SET OF WorkspaceAttribute OPTIONAL,
                                     -- Editable attributes of the workspace
    planeParameters                SEQUENCE (SIZE (1..256)) OF SEQUENCE
                                     -- This sequence contains
                                     -- plane parameters.
                                     -- Its length is the number of planes
                                     -- in the workspace
}

```

```

    editable          BOOLEAN,
                    -- This item specifies whether objects
                    -- created on this plane are editable.
                    -- If not editable, each plane is treated
                    -- as a bitmap image
    usage            SET (SIZE (1..MAX)) OF PlaneUsage,
                    -- This item specifies restrictions
                    -- on the usage of this plane
                    -- (image data or annotation data).
                    -- At least one use shall be included.
                    -- A particular usage designator shall be
                    -- listed no more than once
    planeAttributes SET OF PlaneAttribute OPTIONAL,
                    -- List of attributes.
                    -- A particular attribute shall be
                    -- listed no more than once
    ...
},
viewParameters    SET (SIZE (1..256)) OF SEQUENCE
                  -- Each entry in this list (if any) defines
                  -- a view to be created in association with
                  -- this workspace
{
    viewHandle     Handle,
                  -- Identifier of the view to be created
    viewAttributes SET OF WorkspaceViewAttribute OPTIONAL,
                  -- Editable attributes of the view
    ...
} OPTIONAL,
nonStandardParameters SET OF NonStandardParameter OPTIONAL,
                    -- Allowed only if the corresponding
                    -- non-standard capabilities are present
                    -- in the negotiated capability set
    ...
}

-- WorkspaceCreateAcknowledgePDU
-- This PDU acknowledges the reception of an WorkspaceCreatePDU in
-- the case of unsynchronized workspace
WorkspaceCreateAcknowledgePDU ::= SEQUENCE
{
    workspaceIdentifier WorkspaceIdentifier,
                    -- Workspace being acknowledged
    nonStandardParameters SET OF NonStandardParameter OPTIONAL,
                    -- Allowed only if the corresponding
                    -- non-standard capabilities are present
                    -- in the negotiated capability set
    ...
}

-- WorkspaceDeletePDU
-- This PDU causes a workspace to be destroyed
WorkspaceDeletePDU ::= SEQUENCE
{
    workspaceIdentifier WorkspaceIdentifier,
                    -- Workspace to be deleted
    reason             WorkspaceDeleteReason,
                    -- Reason for deletion of the workspace
    nonStandardParameters SET OF NonStandardParameter OPTIONAL,
                    -- Allowed only if the corresponding
                    -- non-standard capabilities are present
                    -- in the negotiated capability set
    ...
}

```

```

-- WorkspaceEditPDU
-- This PDU allows workspace attributes to be edited
WorkspaceEditPDU ::= SEQUENCE
{
    workspaceIdentifier      WorkspaceIdentifier,
                            -- Workspace to be edited
    attributeEdits           SET OF WorkspaceAttribute OPTIONAL,
                            -- List of attribute changes
    planeEdits              SET (SIZE (1..256)) OF SEQUENCE
    {
        plane                DataPlaneID,
                            -- Plane whose attributes are
                            -- to be edited
        planeAttributes      SET OF PlaneAttribute,
                            -- List of attributes to change
        ...
    } OPTIONAL,
    viewEdits               SET (SIZE (1..256)) OF SEQUENCE
    {
        viewHandle          Handle,
                            -- Identifier of the view to be edited
        action              CHOICE
        {
            createNewView    SET OF WorkspaceViewAttribute,
                            -- Editable attributes of the view
            editView         SET OF WorkspaceViewAttribute,
                            -- List of attributes to change
            deleteView       NULL,
            nonStandardAction NonStandardParameter,
            ...
        },
        ...
    } OPTIONAL,
    nonStandardParameters  SET OF NonStandardParameter OPTIONAL,
                            -- Allowed only if the corresponding
                            -- non-standard capabilities are present
                            -- in the negotiated capability set
    ...
}

```

```

-- WorkspacePlaneCopyPDU
-- This PDU causes a portion of a plane to be copied
-- to another plane (either intra- or inter-workspace).
-- The source and destinations must either both be
-- permanent or both be editable, and they must have the same
-- usage designator otherwise copy for that plane will
-- not take place.
-- If the planes are editable, objects with any of their control
-- points falling totally within the source rectangle are copied.
-- If the Scaling capability has been negotiated in the case of a
-- softcopy workspace then it is not necessary for the source and
-- destination rectangles to be the same size

```

```

WorkspacePlaneCopyPDU ::= SEQUENCE
{
    sourceWorkspaceIdentifier  WorkspaceIdentifier,
                            -- Workspace to be copied
    sourcePlane               DataPlaneID,
                            -- Source plane identifier
    sourceRegion              WorkspaceRegion,
                            -- Source rectangle to be copied
    destinationWorkspaceIdentifier  WorkspaceIdentifier,
                            -- Destination workspace identifier.
                            -- May be the same as the
                            -- source workspace

```

```

destinationPlane
DataPlaneID,
    -- Destination plane identifier.
    -- May be the same as the
    -- source plane

destinationRegion
WorkspaceRegion,
    -- Destination rectangle to be copied.
    -- May be restricted by caps to be the
    -- same size as the source region

nonStandardParameters
SET OF NonStandardParameter OPTIONAL,
    -- Allowed only if the corresponding
    -- non-standard capabilities are present
    -- in the negotiated capability set

...
}

-- WorkspaceReadyPDU
-- This PDU signals that a workspace create is complete (for
-- unsynchronized workspaces)
WorkspaceReadyPDU ::= SEQUENCE
{
    workspaceIdentifier
WorkspaceIdentifier,
    -- Workspace being enabled

    nonStandardParameters
SET OF NonStandardParameter OPTIONAL,
    -- Allowed only if the corresponding
    -- non-standard capabilities are present
    -- in the negotiated capability set

    ...
}

-- WorkspaceRefreshStatusPDU
-- This PDU is used by a SICE to announce or remit its status as
-- the conference refresh SICE for SICEs that join late
WorkspaceRefreshStatusPDU ::= SEQUENCE
{
    refreshStatus
BOOLEAN,
    -- TRUE indicates that the SICE sourcing
    -- this PDU is functioning as the conference
    -- wide refresher.
    -- FALSE indicates that the SICE sourcing
    -- this PDU has ceased to function as the
    -- conference wide refresher

    nonStandardParameters
SET OF NonStandardParameter OPTIONAL,
    -- Allowed only if the corresponding
    -- non-standard capabilities are present
    -- in the negotiated capability set

    ...
}

-- SIPDU
-- The set of all SIPDUs
SIPDU ::= CHOICE
{
    archiveAcknowledgePDU
ArchiveAcknowledgePDU,
archiveClosePDU
ArchiveClosePDU,
archiveErrorPDU
ArchiveErrorPDU,
archiveOpenPDU
ArchiveOpenPDU,

    bitmapAbortPDU
BitmapAbortPDU,
bitmapCheckpointPDU
BitmapCheckpointPDU,
bitmapCreatePDU
BitmapCreatePDU,
bitmapCreateContinuePDU
BitmapCreateContinuePDU,
bitmapDeletePDU
BitmapDeletePDU,
bitmapEditPDU
BitmapEditPDU,

```

conductorPrivilegeGrantPDU	ConductorPrivilegeGrantPDU,
conductorPrivilegeRequestPDU	ConductorPrivilegeRequestPDU,
drawingCreatePDU	DrawingCreatePDU,
drawingDeletePDU	DrawingDeletePDU,
drawingEditPDU	DrawingEditPDU,
remoteEventPermissionGrantPDU	RemoteEventPermissionGrantPDU,
remoteEventPermissionRequestPDU	RemoteEventPermissionRequestPDU,
remoteKeyboardEventPDU	RemoteKeyboardEventPDU,
remotePointingDeviceEventPDU	RemotePointingDeviceEventPDU,
remotePrintPDU	RemotePrintPDU,
siNonStandardPDU	SINonStandardPDU,
workspaceCreatePDU	WorkspaceCreatePDU,
workspaceCreateAcknowledgePDU	WorkspaceCreateAcknowledgePDU,
workspaceDeletePDU	WorkspaceDeletePDU,
workspaceEditPDU	WorkspaceEditPDU,
workspacePlaneCopyPDU	WorkspacePlaneCopyPDU,
workspaceReadyPDU	WorkspaceReadyPDU,
workspaceRefreshStatusPDU	WorkspaceRefreshStatusPDU,
...	
}	
-- End SI Definitions	
END	

Annex A

SI profiles

(This annex forms an integral part of this Recommendation)

A.1 The following profiles are defined as guidelines for terminal vendors wishing to build equipment that is maximally interoperable. Note that the protocol itself explicitly does not support a profile shorthand mechanism and requires explicit advertisement of each capability. This is to ensure proper forward compatibility. A terminal conforms to a profile if it advertises capabilities greater than or equal to the minimums specified. See Table A.1.

TABLE A.1/T.126

SI profiles

Capability name	Hard-Copy-0	Soft-Copy-Image-0	Soft-Copy-Image-1	Soft-Copy-White-Board-0	Soft-Copy-Annotated-Image-0
Hard-Copy-Image	M	O	O	O	O
Soft-Copy-Workspace	O	M	M	M	M
Soft-Copy-Workspace-Max-Width	O	O	≥ 768	O	≥ 768
Soft-Copy-Workspace-Max-Height	O	O	≥ 576	O	≥ 576
Soft-Copy-Workspace-Max-Planes	O	O	O	O	≥ 2
Soft-Copy-Pointing	O	O	O	O	M
Soft-Copy-Annotation	O	O	O	M	M
Soft-Copy-Image	O	M	M	O	M
Soft-Copy-Image-Bitmap-Max-Width	O	O	≥ 768	O	≥ 768
Soft-Copy-Image-Bitmap-Max-Height	O	O	≥ 576	O	≥ 576

The definition of each of these capabilities is listed in Table 8-5. The requirements for specific bitmap encoding algorithms with respect to these capabilities are included in 8.4.6.1 (uncompressed encoding), 8.4.6.2 (T.4 encoding), 8.4.6.4 (T.81 encoding), and 8.4.6.5 (T.82 encoding).

NOTE – Workspace resolutions are measured with respect to a square pixel format but the actual pixel format of exchanged bitmaps can be variable.

Any capabilities which are not listed in Table A.1 are optional for all profiles. Note that some capabilities depend on the presence of others. It is a protocol violation to include any dependent capability in the Application Capabilities List if the capability on which it depends is not also included. These dependencies are listed in Table 8-5.

Annex B

Static channel and token ID assignments

(This annex forms an integral part of this Recommendation)

B.1 Static channel ID assignments

Table B.1 lists the numerical assignment of static channel IDs for the static channels allocated for use by this Recommendation. The numerical assignment of static channel IDs is intended to be centralized in Recommendation T.120, but is included here until T.120 is completed. See Table B.1.

TABLE B.1/T.126

Static channel ID assignments

Symbolic name	Channel ID
SI-CHANNEL-0	8

B.2 Static token ID assignments

Table B.2 lists the numerical assignment of static token IDs for the static tokens allocated for use by this Recommendation. The numerical assignment of static token IDs is intended to be centralized in Recommendation T.120, but is included here until T.120 is completed. See Table B.2.

TABLE B-2/T.126

Static token ID assignments

Symbolic name	Token ID
SI-TOKEN-0	8
SI-TOKEN-1	9

Annex C

Object identifier assignments

(This annex forms an integral part of this Recommendation)

Table C.1 lists the assignment of Object Identifiers defined for use by this Recommendation. See Table C.1.

TABLE C.1/T.126

Assignment of Object Identifiers

Object Identifier value	Description
{itu recommendation t 126 version(0) 1 }	This Object Identifier is used to indicate the version of this Recommendation. At this time there is a single standardized version defined.

Appendix I**Deriving intermediate palettes for bitplane progressive transmission of palettized images**

(This appendix does not form an integral part of this Recommendation)

I.1 It is possible to derive a set of color index tables, one per bitplane sent, that form interim palettes by referencing colors in the final bitmap palette that provide acceptable mappings for each intermediate image built from all the bitplanes transmitted, up to and including the most current. The derivation approach is based on a Kd-tree vector quantizer, which allows binary subdivisions of the color space defined by the original image palette.

One candidate algorithm follows but many variants are possible. These inventions are left to the discretion of the implementer.

Example palette splitting algorithm:

- Initialize the root node of a binary tree to contain all the color entries in the original image palette.
- Find a single suitable representative color for this node from this set. Note that the representative color must be in the original palette. Suitable representatives can be found using techniques such as determining the average value over the set, etc., followed by a matching pass over the bitmap palette to find the closest match.

For each bitplane:

For each leaf:

- Create two children for the current leaf node, each inheriting one part of the constituent palette entries from its parent node. Each color palette entry must be uniquely assigned to one child or the other. This is done by determining median value point along the axis with the maximum error with respect to the selected representative color, and distributing all the constituents that are less than or equal to that value along the split axis to the left child and all those greater than the split point to the right child.
- Calculate a suitable representative color for each node from the nodes' new color set. Note that the representative color must be in the original palette. Suitable representatives can be found using techniques such as determining the average value over the set, etc., followed by a matching pass over the bitmap palette to find the closest match.

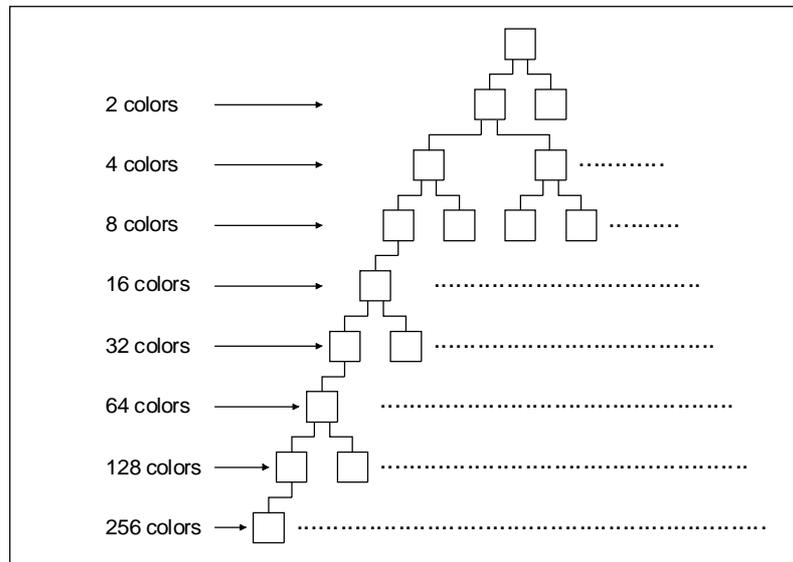
End for each leaf:

- Walk the tree and extract the representative color from each leaf. Each of these leaf colors are then converted to indices by finding the position in the bitmap palette containing the closest matching color to the leaf representative color. Each of these gets placed in the color index table position whose corresponding address has a prefix that matches the string of zeros and ones that describe the path from the root of the tree to that leaf. Zero indicates the taking of a left branch, one indicates a right branch. The string must be populated from MSB to LSB forming the address prefix of the color index table to be filled with the index value referencing the bitmap palette containing the closest match to that leaf representative color.
- Store the color index table formed in the above step to be used for the image formed by bitplanes up to and including the bitplane corresponding to the current depth level of the tree.

End for each bitplane:

NOTE – The final bitplane color index table can be omitted if the bitmap palette is reordered locally by permuting it as specified by the color index table derived for the last bitplane. This new bitmap palette would become the bitmap palette for the image. If this is done, all other color index table values for previous bitplanes must be changed to reference the new reordered bitmap palette. Also note that the pixel data must be reordered by the transmitter before coding such that it maps properly to either the final bitplane’s color index table or the bitmap palette depending on which of the previously mentioned approaches is chosen.

Figure I.1 illustrates how the treewise palette splitting will ultimately look after all split iterations are complete.



T0824490-95/d08

FIGURE I.1/T.126

Palette splitting



Printed in Switzerland
Geneva, 1996