

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.128

(06/2008)

SERIES T: TERMINALS FOR TELEMATIC SERVICES
Data protocols for multimedia conferencing

Multipoint application sharing

Recommendation ITU-T T.128

ITU-T T-SERIES RECOMMENDATIONS
TERMINALS FOR TELEMATIC SERVICES

Facsimile – Framework	T.0–T.19
Still-image compression – Test charts	T.20–T.29
Facsimile – Group 3 protocols	T.30–T.39
Colour representation	T.40–T.49
Character coding	T.50–T.59
Facsimile – Group 4 protocols	T.60–T.69
Telematic services – Framework	T.70–T.79
Still-image compression – JPEG-1, Bi-level and JBIG	T.80–T.89
Telematic services – ISDN Terminals and protocols	T.90–T.99
Videotext – Framework	T.100–T.109
Data protocols for multimedia conferencing	T.120–T.149
Telewriting	T.150–T.159
Multimedia and hypermedia framework	T.170–T.189
Cooperative document handling	T.190–T.199
Telematic services – Interworking	T.300–T.399
Open document architecture	T.400–T.429
Document transfer and manipulation	T.430–T.449
Document application profile	T.500–T.509
Communication application profile	T.510–T.559
Telematic services – Equipment characteristics	T.560–T.649
Still-image compression – JPEG 2000	T.800–T.849
Still-image compression – JPEG-1 extensions	T.850–T.899

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

Recommendation ITU-T T.128

Multipoint application sharing

Summary

Recommendation ITU-T T.128 defines a protocol that supports multipoint application sharing.

The T.128 protocol supports multipoint computer application sharing by allowing a view onto a computer application executing at one site to be advertised within a session to other sites. Each site can, under specified conditions, take control of the shared computer application by sending remote keyboard and pointing device information. This style of application sharing does not require and does not make provision for synchronizing multiple instances of the same computer application running at multiple sites. Instead, it enables remote viewing and control of a single application instance to provide the illusion that the application is running locally.

This Recommendation uses services provided by Recommendations ITU-T T.122 (MCS) and T.124 (GCC).

This revised version of Recommendation ITU-T T.128 introduces a number of modifications to legacy mode ASN.1 definition of the application sharing protocol.

Source

Recommendation ITU-T T.128 was approved on 13 June 2008 by ITU-T Study Group 16 (2005-2008) under Recommendation ITU-T A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

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

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2009

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

1	Scope	1
2	References.....	2
3	Definitions	3
4	Abbreviations and acronyms	4
5	Overview	4
5.1	Legacy and base modes	4
5.2	AS concepts.....	5
6	Use of MCS	10
6.1	MCS channel usage	11
6.2	Use of MCS data services.....	11
7	Use of GCC	13
8	Protocol specification	14
8.1	AS sessions.....	14
8.2	Capabilities	14
8.3	ASPDU formats.....	36
8.4	ASCE activation	40
8.5	Flow control.....	47
8.6	Synchronization.....	52
8.7	Remote sharing.....	56
8.8	Fonts	57
8.9	Application management.....	69
8.10	Window list management	70
8.11	Window activation.....	76
8.12	Control.....	80
8.13	Mediated control.....	83
8.14	Pointers	87
8.15	Palette updates	91
8.16	Order updates.....	92
8.17	Bitmap updates	133
8.18	Input.....	139
8.19	Conducted mode operation.....	147
9	ASPDU definitions	148
9.1	Legacy mode ASN.1 definition.....	148
9.2	Base mode ASN.1 definition.....	168
9.3	Legacy mode encoding rules	190
9.4	Base mode non-collapsing capabilities encoding rules	190
	Annex A – Static channel ID assignments.....	192

	Page
Annex B – Legacy application protocol key	193
Annex C – Object identifier assignments	194
Appendix I – Informative values	195
I.1 Flow control.....	195
I.2 Bitmap caching.....	196
I.3 ColorTable caching	196
I.4 Pointer caching	196
I.5 Desktop save cache.....	196
I.6 General compression	196

iv

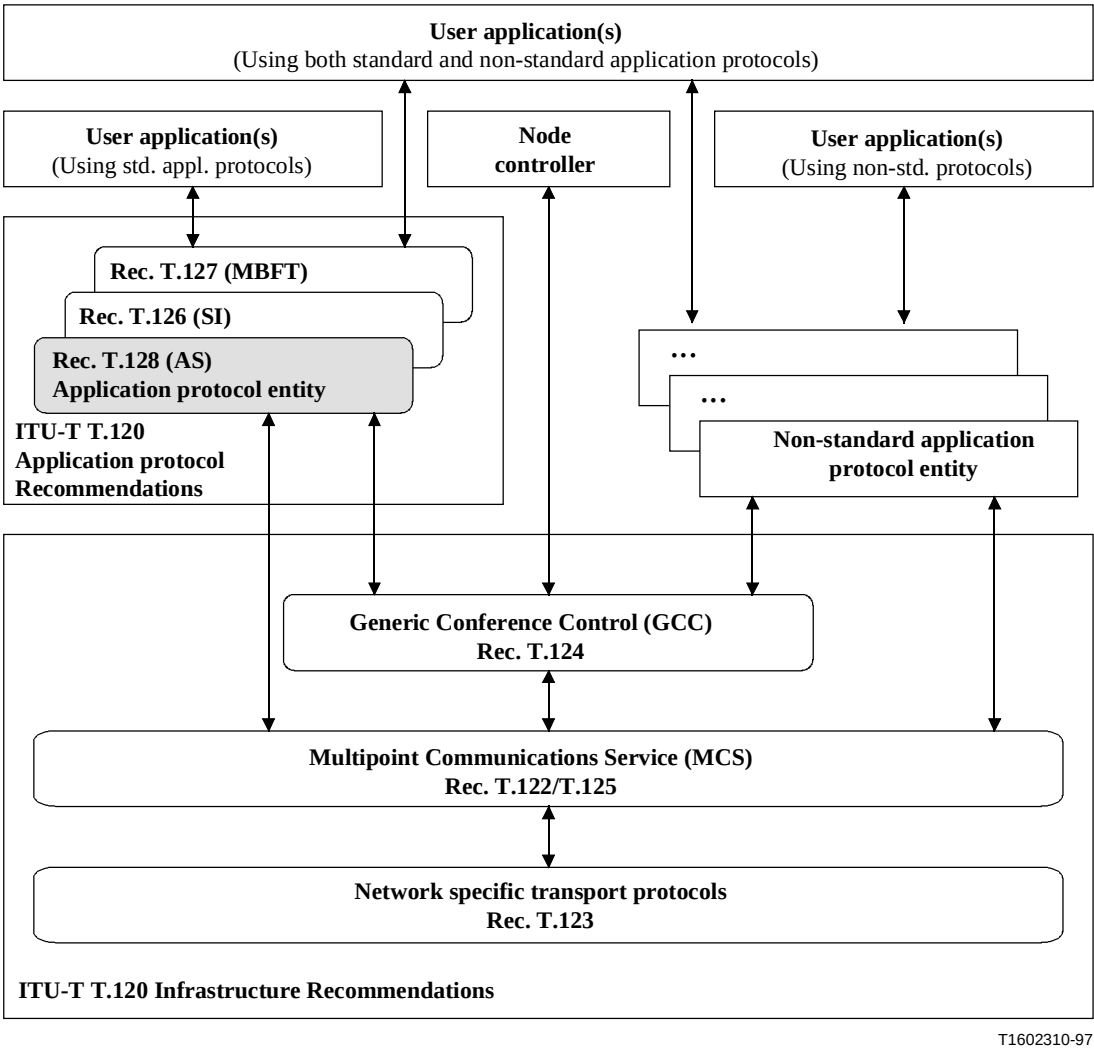
Multipoint application sharing

1 Scope

This Recommendation defines a protocol that supports multipoint application sharing. It uses services provided by [ITU-T T.122] (MCS) and [ITU-T T.124] (GCC).

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 the implementer. Therefore, this Recommendation makes no assumption that these input and output devices and user interfaces are of any specific architecture.

Figure 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.



T1602310-97

Figure 1 – Scope of this Recommendation

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T F.710] Recommendation ITU-T F.710 (1991), *General principles for audiographic conference service*.
- [ITU-T H.221] Recommendation ITU-T H.221 (1997), *Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices*.
- [ITU-T T.35] Recommendation ITU-T T.35 (1991), *Procedure for the allocation of CCITT defined codes for non-standard facilities*.
- [ITU-T T.50] Recommendation ITU-T T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange*.
- [ITU-T T.120] Recommendation ITU-T T.120 (2007), *Data protocols for multimedia conferencing*.
- [ITU-T T.121] Recommendation ITU-T T.121 (1996), *Generic application template*.
- [ITU-T T.122] Recommendation ITU-T T.122 (1993), *Multipoint communication service for audiographics and audiovisual conferencing service definition*.
- [ITU-T T.123] Recommendation ITU-T T.123 (1996), *Network-specific data protocol stacks for multimedia conferencing*.
- [ITU-T T.124] Recommendation ITU-T T.124 (1995), *Generic Conference Control*.
- [ITU-T T.125] Recommendation ITU-T T.125 (1994), *Multipoint communication service protocol specification*.
- [ITU-T V.42 bis] Recommendation ITU-T V.42 bis (1990), *Data compression procedures for data circuit-terminating equipment (DCE) using error correcting procedures*.
- [ITU-T X.509] Recommendation ITU-T X.509 (1997) | ISO/IEC 9594-8:1998, *Information technology – Open Systems Interconnection – The Directory: Authentication framework*.
- [ITU-T X.680] Recommendation ITU-T X.680 (1994) | ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.1) – Specification of basic notation*.
- [ITU-T X.691] Recommendation ITU-T X.691 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules – Specification of Packed Encoding Rules (PER)*.
- [ISO/IEC 8859-1] ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*.
<http://iso.org/iso/iso_catalogue_tc/catalogue_details.htm?csnumber=28245>
- [ISO/IEC 10646] ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*.
<http://iso.org/iso/iso_catalogue_tc/catalogue_details.htm?csnumber=39921>

3 Definitions

This Recommendation defines the following terms:

3.1 application sharing (AS): A process whereby two or more terminals cooperate to share the output of applications running on one or more terminals to the other terminals and to provide input to the applications.

3.2 application sharing conference entity (ASCE): An application protocol entity that interacts with a user application above and with the local multipoint communication service (MCS) and local generic conference control (GCC) providers below to implement application sharing. Data are exchanged between peer ASCEs using application sharing protocol data units (ASPDUs).

3.3 bitmap: A rectangular area described by a two-dimensional array of pixels. These pixels can be coded using a variety of encoding methods.

3.4 ColorTable: A finite set of colours defined by at least three linearly-independent colour primaries. This is a synonym for palette (see definition below), but is used in this Recommendation to reference the particular cached palette associated with cached bitmap data.

3.5 desktop: The logical or physical display area for a particular terminal or window manager advertised by the application sharing conference entity (ASCE) in the application sharing (AS) capabilities. An ASCE may choose to advertise a size corresponding to the actual terminal display area or some other logical display area.

3.6 handle: An application sharing (AS) session-wide unique number used to identify an addressable item.

3.7 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 session participants.

3.8 palette: A finite set of colours defined by at least three linearly-independent colour primaries.

3.9 palettized: A term used to describe protocol elements (such as bitmap data) comprised of palettized pixels. The colour of a palettized pixel is specified by the colour value at the location in a palette referenced by the pixel value.

3.10 pointer: A bitmap that is moveable over the virtual desktop that is used as an indicator of position.

3.11 standard capability: The capability is defined within the scope of this Recommendation but is not required for all application sharing conference entity (ASCE) implementations. Note that all standard capabilities must be negotiated before use.

3.12 unicode: Multilingual text string format as defined in [ISO/IEC 10646].

3.13 virtual desktop: A logical desktop that is the largest size of all desktops of hosting application sharing conference entities (ASCEs).

3.14 window: A rectangular area on the desktop corresponding to a user interface display area managed by the terminal window manager.

3.15 window manager: A program executing on the terminal, which is responsible for managing a collection of user interface windows on the terminal desktop.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

APE	Application Protocol Entity
AS	Application Sharing
ASCE	Application Sharing Conference Entity
ASPDU	Application Sharing Protocol Data Unit
GCC	Generic Conference Control
ID	Identifier
MCS	Multipoint Communication Service
SAP	Service Access Point

5 Overview

The AS protocol enables multipoint computer application sharing by allowing a view onto a computer application executing at one site to be advertised within a session to other sites. Each site can, under specified conditions, take control of the shared computer application by sending remote keyboard and pointing device information. This style of application sharing does not require and does not make provision for synchronizing multiple instances of the same computer application running at multiple sites. Instead, it enables remote viewing and control of a single application instance to provide the illusion that the application is running locally.

An AS session consists of one or more ASCEs which cooperate via the AS protocol to share one or more applications within the session. The AS protocol defines interactions between ASCEs. It does not define interactions between an ASCE and the operating system or input and output devices on the local terminal.

5.1 Legacy and base modes

The AS protocol supports two modes of operation. Legacy mode is provided for interoperability with an existing installed base of customer terminal equipment. Base mode provides additional features that enable future AS protocol enhancement.

All ASCEs compliant with this Recommendation shall implement both legacy and base modes. The AS protocol usage of GCC (see clause 7) is defined such that where all ASCEs in a conference support base mode (i.e., there are no pre-existing legacy mode-only terminals in the conference), then all ASCEs shall use base mode. It is ITU's intention that all future enhancements to the AS protocol shall be by enhancements to base mode.

A significant proportion of this Recommendation is common to both modes, with the primary differences between the two modes being in the following areas:

- The manner in which capabilities are exchanged and negotiated: see clause 8.2.
- The manner in which ASCEs are activated and deactivated: see clause 8.4.
- The definition and encoding of ASPDUs: see clause 9.

Where there are differences between the legacy and base modes of the AS protocol, they are highlighted in the text.

5.2 AS concepts

5.2.1 Desktop and window model

This Recommendation does not assume or require any particular local terminal equipment or terminal environment, nor does it assume or require a particular local terminal window manager or windowing model. For example, it does not assume:

- that the terminal equipment is a PC or workstation – it may be a dedicated terminal;
- that AS protocol windows correspond to windows managed by a terminal window manager – they may map onto arbitrary areas of the local terminal display;
- that AS protocol desktops correspond to local terminal displays – they may map into local browser or viewer windows, or onto dedicated terminal display areas.

While this Recommendation does not assume or require any particular local terminal equipment or terminal environment, the AS protocol does define a logical desktop and window model – consisting of a collection of windows on a desktop. This requires that each active ASCE within an AS session is capable of mapping local terminal environment concepts to and from the AS protocol desktop and window model. The AS protocol desktop and window model supports the following key concepts:

- **Desktop** The AS desktop is a rectangle defined in virtual desktop coordinates. An ASCE should provide a mapping from the AS desktop to an appropriate local terminal concept.
- **Virtual desktop** The virtual desktop is the union of the sizes of the desktops of hosting ASCEs (i.e., ASCEs that are hosting windows – see below). The virtual desktop origin (i.e., pixel 0,0) is defined as being at (top, left).
- **Window** An AS window is a rectangle defined in virtual desktop coordinates. AS windows may be wholly within, partially within or wholly outside the virtual desktop. An ASCE should provide a mapping from AS windows to an appropriate local terminal window concept.
- **Z-order** The AS window Z-order defines a window depth ordering between windows on the virtual desktop, such that for two windows, the window higher in the Z-order is in front of and/or may obscure the other.
- **Top-most** The top-most AS window in the Z-order is in front of and/or may obscure all other windows in the Z-order.
- **Bottom-most** The bottom-most AS window in the Z-order is behind and/or may be obscured by all other windows in the Z-order.

Further AS protocol concepts are explained below or in the appropriate protocol description clauses.

Figure 2 shows an example collection of ASCEs within an AS session – this figure is used throughout this clause to illustrate key AS protocol concepts.

- ASCE A is full function, viewing and hosting – that is, it both shares windows into the AS session and displays shadow windows shared from other ASCEs. It is executing on a general purpose PC, where the AS desktop and AS windows are mapped directly onto the terminal window manager's desktop.
- ASCE B is viewing only – that is, it displays shadow windows shared from other ASCEs but does not itself share windows into the AS session. It is executing on a specialized viewing terminal where the AS desktop and AS windows are mapped into an independently sizeable browser/viewer window, which is (currently) smaller than the AS desktop and is scrolled as required to view the entire AS desktop.

- ASCE C is hosting only – that is, it shares windows into the AS session, but does not display shadow windows. It is executing on a high-end unattended workstation, where the AS desktop and AS windows are mapped directly onto the terminal window manager's desktop.

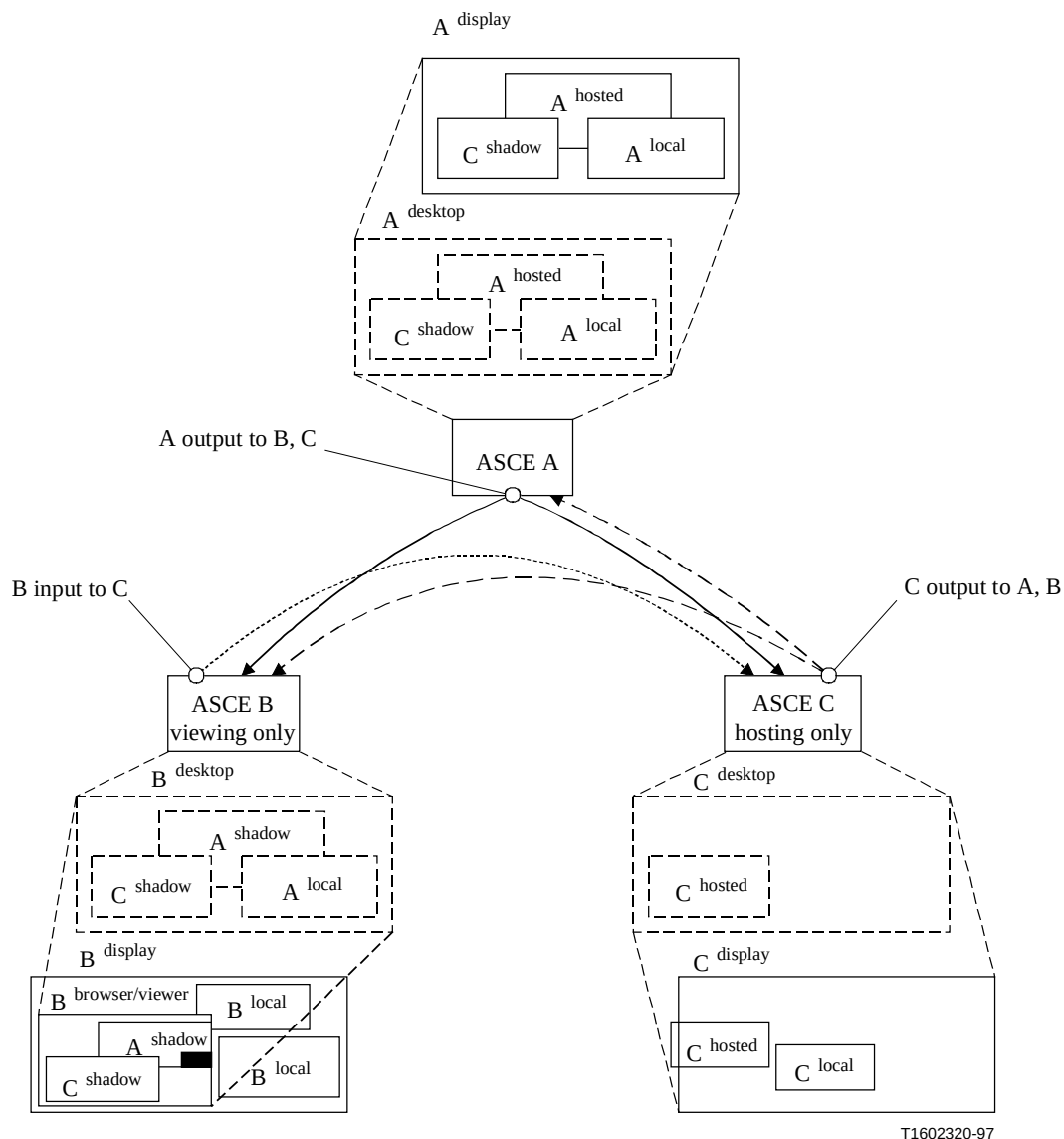


Figure 2 – Example collection of ASCEs within an AS session

Within an AS session, windows are of the following types:

- Hosted:** Hosted windows are owned by an application executing on the local terminal and are shared into the AS session. For each hosted window, there will be a corresponding shadow window on each peer ASCE (except for the case where an ASCE is hosting only).
- Shadow:** Shadow windows are drawn by the ASCE and correspond to a hosted window on a particular peer ASCE.
- Local:** Local windows are not shared – their output is only visible on the local terminal. An ASCE is only required to track local windows where they obscure a hosted window and where that obscuring prevents the ASCE obtaining valid drawing information for the hosted window.

Figure 2 shows (for each ASCE) both the local windows on the actual terminal display and the AS windows on the AS desktop that the ASCE manages via the AS protocol.

- ASCE A is managing three windows:
 - a shadow window corresponding to the hosted window on ASCE C;
 - a hosted window which is shared into the AS session;
 - a local window – because it obscures the hosted window and on the local terminal that prevents it obtaining valid drawing information for that hosted window.

Shadow window C is drawn on the local terminal display by the ASCE – and the other two windows are displayed by the local terminal.

- ASCE B is also managing three windows:
 - a shadow window corresponding to the hosted window on ASCE A;
 - a shadow window corresponding to the hosted window on ASCE C;
 - the local window from ASCE A – because it obscures the hosted window on ASCE A;
 - ASCE B does not track the local windows on its terminal display as it does not host windows.

Shadow windows A and C are drawn into the browser/viewer window on the local terminal display by the ASCE. The local window is not drawn, but is rather used to compute and draw an obscuring area, which partially obscures shadow window A (shown as a dark area at the bottom right of shadow window A).

- ASCE C is managing one window:
 - a hosted window which is shared into the AS session;
 - ASCE C does not track the shadow window corresponding to the hosted window on ASCE A as it does not view shadow windows;
 - ASCE C does not track its local window, as, while the local window does obscure the hosted window, the local terminal allows it to obtain valid drawing information for that hosted window.

The hosted window is displayed by the local terminal.

See clause 8.10 for further information on windows and window list management.

5.2.2 Output

When an ASCE is hosting a window, it is responsible for constructing an output stream for that window which will allow other ASCEs to faithfully draw corresponding shadow windows.

The AS output protocol stream is multicast via the AS-CHANNEL to all other active ASCEs within the conference. If an ASCE is hosting only (as is ASCE C in Figure 2), then it can ignore some or all of the output stream. Similarly, a receiving ASCE may decide to draw only a subset of the output stream based on its source (e.g., only draw windows from a particular ASCE).

An ASCE determines the allowable output repertoire based on the current negotiated capabilities. This requires that the ASCE is capable of adjusting its output strategy as ASCEs with lower or higher capabilities join and/or leave the conference. But within the current allowable repertoire, and the inherent constraints of the AS protocol, an ASCE has considerable freedom in constructing an output stream.

The output stream consists of:

- state information – such as window list information (see clause 8.10);
- color information – such as palette and cached ColorTable information (see clauses 8.15 and 8.16.8);

- orders and bitmap data (see clauses 8.16 and 8.17).

The AS protocol supports the following primary orders:

- Destination Blt.
- Pattern Blt.
- Screen Blt.
- Memory Blt.
- Memory three-way Blt.
- Text.
- Extended text.
- Frame.
- Rectangle.
- Line.
- Opaque rectangle.
- Desktop save.
- Desktop origin.

In typical application sharing scenarios, orders constitute a very high proportion of ASPDU traffic. Orders may therefore be differentially encoded to reduce the volume of orders in the output stream – this process is referred to as order encoding.

The AS protocol supports a single bitmap data format, which may be compressed using an AS-specific two-dimensional run-encoding bitmap compression algorithm.

Depending on local application drawing activity into hosted windows and on flow control (see clause 8.5), a hosting ASCE may use a range of strategies to reduce the volume of the output stream and/or retain responsiveness of remote drawing. Possible strategies include the following:

- Switching between sending orders and bitmap data:
Where a local application is very actively drawing into a hosted window and the ASCE is experiencing flow control back pressure, the ASCE may prefer to accumulate bounds information for the drawing activity, rather than sending the orders, and then subsequently send bitmap data for the accumulated bounds. This may reduce the frequency of updates at remote ASCEs, but will minimize the data in flight and ensures that remote ASCEs "keep up" with the drawing activity. The switching points between orders and bitmap data and vice versa will depend on application drawing behaviour, flow control and the particular ASCE implementation.
- Collapsing overlapping updates – this process is referred to as spoiling:
Where the ASCE is experiencing flow control back pressure and the local application draws a sequence of overlapping or occluding order or bitmap data updates, then the sequence may (in some cases) be collapsed into a smaller sequence and/or a single update. Again, this may reduce the frequency of updates at remote ASCEs, but will minimize the data in flight and ensure that remote ASCEs "keep up" with the drawing activity.

A hosting ASCE is required to ensure that the final constructed output stream allows receiving ASCEs to correctly draw the corresponding shadow windows. Similarly, where the output stream relies on control information (such as window list or palette information) that has changed on the hosting ASCE, it is the hosting ASCE's responsibility to ensure that the control information is sent prior to the output. All order-dependent control and output ASPDUs are sent at low priority, so that the sending ASCE can reliably order such output.

5.2.3 Control and input

Within an AS session, control is managed via a combination of conductorship, the core control protocol and (where negotiated) the mediated control protocol – see clauses 8.19, 8.12 and 8.13, respectively. Conductorship has preference, in that where the conference is in conducted mode, the AS-specific control protocols are not used (other than to enforce conductorship). Where conductorship is not in force, then control is managed via a combination of the core control protocol and (where negotiated) the mediated control protocol.

A key common concept to these control schemes is that, at any point in time within the AS session, one ASCE is in control and has the right to provide input to hosted and/or shadow windows¹.

Where an ASCE is in control and can provide input to hosted or shadow windows (depending on whether it or other ASCEs are detached or not), then, when a local input event occurs, the ASCE determines, based on its local window structure and the shared window structure on its logical desktop, whether that event is for a local, hosted or shadow window. Where the event is for a local or hosted window, then the ASCE can defer processing of the event to the local terminal. Where the event is for shadow window, then the ASCE determines which peer ASCE owns the corresponding hosted window and constructs a suitable input stream for that ASCE.

The AS input stream consists of interleaved keyboard and pointing device events. Keyboard events may be represented either as codepoints or as virtual keycodes. Pointing device events are represented as a logical three-button pointing device. See clause 8.18 for further information.

Where an ASCE is experiencing flow control back pressure (see clause 8.5) and there is a high level of end-user pointing device activity, the ASCE may collapse a sequence of pointing device move events into a single move – this process is referred to as input spoiling. Input spoiling may also be applied at the hosting ASCE, where a sequence of pointing device events is queued within the ASCE awaiting injection into the local terminal environment. Both send and receive input spoiling may reduce the frequency of pointing device move updates presented to hosted windows and therefore degrade the smoothness of pointing device movement, but will minimize the data in flight and ensure that remote ASCEs "keep up" with the pointing device activity.

5.2.4 Colour

The AS protocol carries colour information for the processing of orders and bitmap data.

- For orders (with the exception of the memory Blt and memory three-way Blt orders – see below) colour information is embedded in the order itself – although it may be absent on individual orders as a result of order encoding.
- Bitmap data is always palettized and is interpreted with reference to a previously sent palette which contains the colour information.
- For the memory Blt and memory three-way Blt orders, the order contains some of the required colour information, but also refers to cached bitmap data (as the source for the operation), where that cached bitmap data (see clause 8.16.7) is always palettized and is interpreted with reference to a previously cached ColorTable (see clause 8.16.8) which contains the colour information.

In the legacy mode of the AS protocol, colour information is expressed solely as RGBs. In base mode, colour information is expressed as RGBs with optional colour accuracy information.

¹ This Recommendation does not specify control behaviour with respect to local ASCE windows.

ASCEs may supply colour space information:

- within palettes: where the colour space only applies to that palette;
- within ColorTables in cache ColorTable orders: where the colour space only applies to that cached ColorTable;
- in colour space orders: where the colour space applies to subsequent orders.

The AS protocol does not mandate a particular colour mapping between the ASCE and the local terminal, although ASCEs should attempt to maintain colour fidelity (within the constraints of the negotiated capabilities) when constructing and interpreting colour-related information in the output stream.

5.2.5 Coordinates and clipping

All coordinates in the AS protocol are in virtual desktop coordinates. The virtual desktop is the union of the sizes of the desktops of hosting ASCEs (i.e., ASCEs that are hosting windows) and is regarded as the common drawing area in use by hosting ASCEs. The virtual desktop origin (i.e., pixel 0,0) is defined as being at (top, left).

An ASCE shall clip all AS protocol coordinates and rectangles to the virtual desktop. The AS protocol does not mandate a particular clipping strategy with respect to the local terminal display area.

All rectangles in the AS protocol are inclusive. That is, the right-most and bottom-most pixel coordinates represent pixels that are within the rectangle. For example, a rectangle with coordinates (10,10,19,19) is 10 by 10 pixels in size.

6 Use of MCS

All T.128 communication shall be through MCS as specified in [ITU-T T.122]. This clause details specific use of MCS services, channel allocation and data priorities. This Recommendation complies with the mechanisms described in [ITU-T T.121] regarding proper operations for standard base sessions, non-standard base sessions and the registration session. All other session types are optional.

An ASCE uses the MCS service primitives described in Table 6-1 to attach and detach from a domain, join and leave the AS channel, and send and receive ASPDUs.

Table 6-1 – MCS primitives needed by an ASCE

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.

Table 6-1 – MCS primitives needed by an ASCE

MCS primitive	Description
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-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 request primitives are directed from the ASCE to the MCS provider, while indication primitives are directed from the MCS provider towards the ASCE. Additional detail on the MCS primitives described above can be found in [ITU-T T.122].

6.1 MCS channel usage

Table 6-2 describes MCS channel usage for ASCE sessions of the types defined in [ITU-T T.121]. In the case of a standard base session (see [ITU-T T.121]) which uses the base mode of the AS protocol, the channel IDs shown in Table 6-2 shall be used (symbolic IDs shown). For all other session types, the application registry resource IDs shown in the table shall be used for allocating dynamic channels. The given resource IDs shall be encoded as three-octet T.50 text strings using the characters shown in quotes in Table 6-2.

Table 6-2 – Description of AS channels

Mnemonic	Channel IDs for static channel	Application registry resource IDs for dynamic channels	Description
AS-{MCS-USER-ID}-CHANNEL	–	–	Certain ASPDUs are sent directly to individual ASCEs. To do this, the individual MCS-USER-ID channels of the peer ASCEs in the MCS domain are used.
AS-CHANNEL	AS-CHANNEL-0	"421"	This channel bears all ASPDUs to be broadcast to all peer ASCEs in a domain.

6.2 Use of MCS data services

Table 6-3 lists the use of the MCS data service MCS-SEND-DATA for each ASPDU (with mode-specific ASPDUs indicated where necessary). ASCEs do not use the MCS data service MCS-UNIFORM-SEND-DATA. This table includes the channel over which the data are sent and the data priority at which the data are sent.

The AS protocol uses three MCS priorities as a means of segregating ASPDUs with respect to flow control. Flow control is applied on a per-channel and per-priority basis (see clause 8.5), such that grouping ASPDUs by priority allows the application of a flow control regime that is applicable to the group, as follows:

- High priority is flow controlled and is used for activation, flow control and input ASPDUs.
- Medium priority is not flow controlled and is used for control ASPDUs.

- Low priority is flow controlled and is used for window information and output ASPDUs.

All ASPDUs specified in this Recommendation are placed in the data parameter of the MCS-SEND-DATA primitive. The ASPDUs 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 6-3 – Use of MCS data primitives for ASPDUs

ASPDU	Channel	Modes	Priority
ApplicationPDU	AS-CHANNEL or AS-{MCS-USER-ID}-CHANNEL (Note 1)	Both	Medium
ConfirmActivePDU	AS-CHANNEL	Legacy mode	All (Note 2)
ControlPDU	AS-CHANNEL	Both	Medium
DeactivateAllPDU	AS-CHANNEL	Legacy mode	High
DeactivateOtherPDU	AS-{MCS-USER-ID}-CHANNEL	Legacy mode	High
DeactivateSelfPDU	AS-CHANNEL	Legacy mode	High
DemandActivePDU	AS-CHANNEL	Legacy mode	High
FlowResponsePDU	AS-{MCS-USER-ID}-CHANNEL	Both	High
FlowStopPDU	AS-CHANNEL	Both	High
FlowTestPDU	AS-CHANNEL	Both	High, medium or low (Note 3)
FontPDU	AS-CHANNEL	Both	Medium
InputPDU	AS-CHANNEL	Both	High
MediatedControlPDU	AS-CHANNEL or AS-{MCS-USER-ID}-CHANNEL (Note 4)	Both	Medium
PointerPDU	AS-CHANNEL	Both	Medium
RemoteSharePDU	AS-{MCS-USER-ID}-CHANNEL	Both	Medium
RequestActivePDU	AS-{MCS-USER-ID}-CHANNEL	Legacy mode	High
SynchronizePDU	AS-CHANNEL	Both	All (Note 5)
UpdateCapabilityPDU	AS-{MCS-USER-ID}-CHANNEL	Legacy mode	Medium
UpdatePDU	AS-CHANNEL	Both	Low

Table 6-3 – Use of MCS data primitives for ASPDUs

ASPDU	Channel	Modes	Priority
WindowActivationPDU	AS-CHANNEL or AS-{MCS-USER-ID}-CHANNEL (Note 6)	Both	Low
WindowListPDU	AS-CHANNEL	Both	Low
NOTE 1 – The ApplicationPDU (NotifyHostedApplications) is sent on the AS-CHANNEL and the ApplicationPDU (UnhostApplication) is sent on a particular AS-{MCS-USER-ID}-CHANNEL. See clause 8.9 for further information. NOTE 2 – The ConfirmActivePDU is sent on all priorities. See clause 8.4.1 for further information. NOTE 3 – The FlowTestPDU is sent on the particular priority for which flow control is being applied. See clause 8.5 for further information. NOTE 4 – The MediatedControlPDU is sent on either the AS-CHANNEL or a particular AS-{MCS-USER-ID}-CHANNEL. See clause 8.13 for further information. NOTE 5 – The SynchronizePDU is sent on one of the high, medium or low priorities, as required. See clause 8.6 for further information. NOTE 6 – WindowActivationPDU indications are sent on the AS-CHANNEL and WindowActivationPDU requests are sent on a particular AS-{MCS-USER-ID}-CHANNEL. See clause 8.11 for further information.			

7 Use of GCC

The legacy mode of the AS protocol shall use the procedures of a non-standard base session in the manner specified in [ITU-T T.121] and shall use the application protocol key defined in Annex B. The base mode of the AS protocol may use the procedures defined for a registration session, a standard base session, a public session, or a private session in the manner specified in [ITU-T T.121] and shall use as its application protocol key the object identifier defined in Annex C.

All ASCEs compliant with this Recommendation shall first enrol actively or inactively in the registration session (to signal support for the base mode of the AS protocol) and then enrol actively or inactively in the non-standard base session (to signal support for the legacy mode of the AS protocol), in both cases using the procedures defined in [ITU-T T.121], and shall stay enrolled in both sessions for as long as support for the AS protocol is to be indicated.

Pre-existing terminal equipment that only provides support for legacy mode will only enrol actively or inactively in the non-standard base session using the procedures defined in [ITU-T T.121] and will stay enrolled in that session for as long as such support is to be indicated.

An ASCE shall not enrol in the standard base session of the AS protocol unless for each node in a conference, there are an equal number of ASCEs enrolled either actively or inactively in the non-standard base session as there are enrolled in the registration session.

If the non-standard base session is in progress (i.e., all ASCEs are active in legacy mode), upon receiving a GCC-Conference-Roster-Report indication where the number of ASCEs enrolled in the registration session is equal to the number of ASCEs enrolled in the non-standard base session in the conference (i.e., all ASCEs in the conference are compliant), all ASCEs shall become inactive in legacy mode, shall enrol in the standard base session and shall become active in base mode. See clause 8.4 for further information on ASCE activation.

If the standard base session is in progress (i.e., all ASCEs are active in base mode), upon receiving a GCC-Conference-Roster-Report indication where the number of ASCEs enrolled in the registration session is less than the number of ASCEs enrolled in the non-standard base session in the conference (i.e., one or more ASCEs in the conference are non-compliant pre-existing terminal

equipment), all ASCEs enrolled in the standard base session shall become inactive in base mode, shall unenroll from the standard base session and shall become active in legacy mode. See clause 8.4 for further information on ASCE activation.

The above rules for switching between legacy and base modes mean that legacy mode will only be in effect where the conference contains pre-existing terminal equipment supporting the legacy mode of the AS protocol only. Where all ASCEs in the conference comply with this Recommendation, then base mode will be in effect. Given this, the switch from base to legacy mode occurs when the first legacy-only mode node joins a conference consisting entirely of ASCEs that are compliant with this Recommendation. In contrast, the switch from legacy mode to base mode occurs when the last legacy only mode node leaves a conference which then consists entirely of ASCEs that are compliant with this Recommendation.

ASCEs supporting the base mode of the AS protocol may enrol in a public session or private session at their discretion using the procedures defined in [ITU-T T.121].

8 Protocol specification

8.1 AS sessions

An AS session consists of one or more ASCEs enrolled within a conference as described in clause 7. ASCEs may join or leave the AS session at any time.

8.2 Capabilities

AS capabilities are collected into capability sets, where each set consists of related individual capabilities. Capability sets are described in Tables 8-3 through 8-20. Hereafter, where this Recommendation makes reference to an individual capability, it uses the notation **capability_set.capability**. For example **Bitmap.desktopWidth** refers to the **desktopWidth** capability in the **Bitmap** capability set.

The full list of capability sets is referred to as an ASCE's combined capabilities. The combined capabilities list contains one copy of each of the corresponding capability sets in any order.

- General capability set: See clause 8.2.3.
- Bitmap capability set: See clause 8.2.4.
- Order capability set: See clause 8.2.5.
- Bitmap cache capability set: See clause 8.2.7.
- ColorTable cache capability set: See clause 8.2.8.
- Window activation capability set: See clause 8.2.9.
- Control capability set: See clause 8.2.10.
- Pointer capability set: See clause 8.2.11.
- Share capability set²: See clause 8.2.12.

In the legacy mode of the AS protocol, an ASCE may define non-standard capability extensions by adding private capability sets and/or by adding private capabilities at the end of the defined capability sets. The interpretation of such capabilities is not covered by this Recommendation. When using the legacy mode of the AS protocol, an ASCE should ignore unrecognized non-standard capability sets and unrecognized capability set extensions. Similarly, an ASCE shall

² The share capability set is only defined for the legacy mode of the AS protocol. It does not form part of an ASCE's combined capabilities in the base mode of the AS protocol.

treat any private capabilities that are not supplied by other ASCEs as zero (for integer values) or FALSE (for logical and bit flag values).

In the base mode of the AS protocol, an ASCE may define non-standard capability extensions by adding one or more non-standard capability sets (see clause 8.2.13) to its combined capabilities. The interpretation of such capabilities is not covered by this Recommendation.

8.2.1 Distribution of capabilities

For the legacy mode of the AS protocol, an ASCE advertises its combined capabilities during ASCE activation on the following ASPDUs.

- DemandActivePDU: See clause 8.4.1.
- RequestActivePDU: See clause 8.4.1.
- ConfirmActivePDU: See clause 8.4.1.

For the legacy mode of the AS protocol, an ASCE advertises a change in a particular capability set using the UpdateCapabilityPDU (see clause 8.2.14).

For the base mode of the AS protocol, capabilities exchange shall be performed according to [ITU-T T.121], via the application enrolment mechanism. An ASCE shall use GCC-Application-Enroll requests to enrol and advertise its current combined capabilities. Where an ASCE's capabilities change while enrolled, it shall re-enrol by issuing a GCC-Application-Enroll request with the Enroll/Unenroll flag set to Enroll, including the revised combined capabilities. ASCEs receive collapsed and non-collapsed application capabilities, generated from the combined capabilities supplied by all enrolled ASCEs, via GCC-Application-Roster-Report indications. An ASCE may be required to process GCC-Application-Roster-Report indications multiple times during an AS session and shall adhere to the bounds imposed by the capabilities reported in this fashion.

8.2.2 Capabilities negotiation

In the legacy mode of the AS protocol, the ASCE activation ASPDU exchanges (see clause 8.4.1) ensure that an ASCE has a copy of each other active ASCE's capabilities. And, in addition, an ASCE is responsible for performing all capabilities negotiation. That is, GCC is not involved in capabilities distribution or negotiation.

In the base mode of the AS protocol, capabilities are distributed by GCC. There are a small number of capabilities that must be advertised via the non-collapsing capabilities list in the roster – they are specified as such in the description of the capability in the appropriate base mode capability set. However, in general, a particular capability may be advertised via one of either the collapsing or non-collapsing capabilities lists in the roster. Where an ASCE advertises a particular capability via one list, it shall not advertise the same capability via the other list. However, an ASCE is not required to advertise a capability – it may choose not to advertise a particular capability in either list and rely on the defined GCC roster counting behaviour to force use of a default value (see below).

This hybrid approach to base mode capabilities distribution is provided so that ASCEs may advertise capabilities via the collapsing capabilities, which results in a reduction in roster-related GCC traffic, but may still advertise capabilities via the non-collapsing capabilities where that is more appropriate. ASCEs are recommended to use collapsing capabilities wherever possible.

In the base mode of the AS protocol, where an ASCE advertises a particular capability via the non-collapsing capabilities list in the roster, the ASCE shall encode the capability value using the encoding rules defined in clause 9.4.

ASCE capabilities negotiation for a particular capability depends on the defined capability negotiation rule for that capability, the protocol mode (i.e., whether the ASCE is using legacy or

base mode) and whether (in base mode) the capability is advertised via collapsing or non-collapsing capabilities.

8.2.2.1 One and info capability rules

Where the defined capability negotiation rule is one or info, then an ASCE shall process the advertised capabilities as follows.

In base mode, where a capability is negotiated using either the one or info rules, the capability shall be advertised using non-collapsing capabilities.

Rule	Required ASCE processing
One	The negotiated capability value is the value advertised by a particular peer ASCE.
Info	Capabilities may also be classified as informational – they are provided solely for informational or diagnostics purposes and are not negotiated.

8.2.2.2 Min and max capability rules

Where the defined capability negotiation rule is min or max, then an ASCE shall process the advertised capabilities based on the number of candidate values, as follows:

- For the legacy mode of the AS protocol, candidate values correspond to the advertised values supplied by each active ASCE during ASCE activation, but excluding the value advertised by the negotiating ASCE. If there are N active ASCEs, then there are N–1 candidate values.
- For the base mode of the AS protocol, candidate values consist of the collapsed capability value provided in the roster and any non-collapsing values, but excluding a non-collapsing value advertised by the negotiating ASCE. The roster indicates how many ASCEs advertised capability values contributed to the collapsed value, which allows an ASCE to determine whether all ASCEs advertised a value. So, if there are N ASCEs in the roster, C ASCEs advertised collapsing values and NC ASCEs advertised non-collapsing values for a particular capability, then:
 - if $N = C$, all ASCEs advertised collapsing values for the particular capability: there is one candidate value;
 - if $N = NC$, all ASCEs advertised non-collapsing values for the particular capability: there are NC–1 candidate values – where the non-collapsing value advertised by the negotiating ASCE is excluded;
 - if $N = C + NC$, all ASCEs advertised either a collapsing or non-collapsing value for the particular capability: there are:
 - C + NC–1 candidate values where the negotiating ASCE advertised a non-collapsing value which is excluded;
 - C + NC candidate values where the negotiating ASCE advertised a collapsing value;
 - if $N > C + NC$, not all ASCEs advertised a value: the negotiated capability value is the default value (see below).

In base mode, if an ASCE determines that all ASCEs have advertised collapsing values (i.e., case $N = C$ above), then the negotiated capability value is the collapsed value in the roster.

In base mode, if an ASCE determines that not all ASCEs advertised a value (i.e., case $N > C + NC$ above), then the negotiated capability value is the default value. For logical capabilities, the default value is FALSE (or not established). For integer capabilities, the default value is specified in the description of the capability in the appropriate base mode capability set.

For legacy mode, or for base mode where all ASCEs advertised values and the number of candidate values is greater than one (i.e., some or all ASCEs advertised non-collapsing values – cases $N = NC$ and $N = C + NC$ above), then the ASCE applies the defined min or max capability negotiation rule, as follows.

Rule	Required ASCE processing
Min	The negotiated capability value is the minimum of the candidate values: - For integer values, the minimum is the lowest integer value. - For logical values, the minimum is FALSE. - For bit flags (in legacy mode only), each bit flag is negotiated independently and the minimum is not set.
Max	The negotiated capability value is the maximum of the candidate values: - For integer values, the maximum is the highest integer value. - For logical values, the maximum is TRUE. - For bit flags (in legacy mode only), each bit flag is negotiated independently and the maximum is set.

8.2.2.3 Group capability rules

ASCE capabilities negotiation is generally performed on single capability values. However, there are cases where a number of capabilities are negotiated as a group – for example, negotiation of the bitmap capability set bits-per-pixel related capabilities (see clause 8.2.4). Where this is the case, an individual capability in the group may be negotiated either by using one of the previously described capability negotiation rules – such as min or max – with the negotiated value being fed into the group negotiation, or by feeding its unnegotiated value directly into the group negotiation.

Where capabilities are negotiated as a group, they are identified as such in the particular capability set together with a reference to the specific rule used to negotiate the group.

Table 8-1 summarizes the notation used in the remainder of this clause when describing capabilities.

Table 8-1 – Capability notation – Classes

Class	Description
L	Logical: A logical value. In legacy mode, the value is either TRUE or FALSE. In base mode, if the capability is supplied, the value is TRUE. In base mode, the default value is FALSE.
F	Bit flags: A collection of bit values, each of which is TRUE or FALSE. This class is only allowed in legacy mode capability sets.
N	Integer: A signed or unsigned integer value. In base mode, the default value is as specified in the description of the capability set.
S	String: A null-terminated T.50 text string. This class is only allowed in legacy mode capability sets.

Table 8-2 illustrates example capability negotiations. Each row shows, for each ASCE (ASCEs 1 and 2), the capability value set (for each class) by the ASCE when advertising its capabilities, the resultant values for the one rule with respect to the other ASCEs and the negotiated value for the min and max rules for the cases where all ASCEs advertise using collapsing capabilities and where all advertise using non-collapsing capabilities (which gives four combinations: min (C) is the min

rule and all advertise using collapsing capabilities, min (NC) is the min rule and all advertise using non-collapsing capabilities, and so on ...)³. For example:

- ASCE 1's advertised logical value is TRUE, its bit flag value is 0x0001 and its integer value is 100.
- ASCE 2's advertised logical value is FALSE, its bit flag value is 0x0003 and its integer value is 300.
- When ASCE 1 applies the one rule with respect to ASCE 2, it generates ASCE 2's advertised values – i.e., FALSE, 0x0003 and 300 respectively.
- When ASCE 1 applies the max rule for the case where both ASCEs advertised using collapsing capabilities, it generates the maximum of all advertised values – i.e., TRUE and 300 respectively.
- When ASCE 1 applies the max rule for the case where both ASCEs advertised using non-collapsing capabilities, it generates the maximum of all advertised values excluding its own – i.e., FALSE, 0x0003 and 300 respectively.
- When ASCE 2 applies the one rule with respect to ASCE 1, it generates ASCE 1's advertised values – i.e., TRUE, 0x0001 and 100 respectively.

This is not an exhaustive example. The itemized cases cover four out of the ten example cases. In addition, there are further combinations where ASCEs provide mixed collapsing and non-capabilities.

Table 8-2 – Example capability negotiation

Class	ASCE 1						ASCE 2					
	Value	One ⇒2	Min (C)	Min (NC)	Max (C)	Max (NC)	Value	1⇐ One	Min (C)	Min (NC)	Max (C)	Max (NC)
B	T	F	F	F	T	F	F	T	F	T	T	T
F	x0001	x0003	n/a	x0003	n/a	x0003	x0003	x0001	n/a	x0001	n/a	x0001
N	100	300	100	300	300	300	300	100	100	100	300	300

8.2.3 General capability set

The general capability set provides capabilities for the general characteristics of the issuing ASCE. See Tables 8-3 and 8-4.

Table 8-3 – General capability set (legacy mode)

Capability	Description	Class	Rule
OSMajorType	This capability indicates the operating system major type. Allowable values are as follows: • Unspecified. • Windows. • OS/2. • Macintosh. • UNIX/X. This capability is for information and diagnostic purposes only.	N	Info

³ The bit flag class is only allowed in legacy mode, where the distribution of capabilities via the ASCE activation ASPDUs is equivalent to advertising via non-collapsing capabilities. Therefore, the bit flag class resultant values for min and max collapsing capability cases are not shown.

Table 8-3 – General capability set (legacy mode)

Capability	Description	Class	Rule												
OSMinorType	This capability indicates the operating system minor type. The values depend on the OSMajorType capability (see above). Allowable values are as follows: <table><tr><td>OSMajorType</td><td>OSMinorType</td></tr><tr><td>Windows</td><td>Unspecified Windows 3.1x Windows 95 Windows NT</td></tr><tr><td>OS/2</td><td>Unspecified OS/2 Warp (Intel x86) PowerPC</td></tr><tr><td>Macintosh</td><td>Unspecified Macintosh PowerPC</td></tr><tr><td>UNIX/X</td><td>Unspecified Native server Pseudo server</td></tr><tr><td>Unspecified</td><td>Unspecified</td></tr></table> This capability is for information and diagnostic purposes only.	OSMajorType	OSMinorType	Windows	Unspecified Windows 3.1x Windows 95 Windows NT	OS/2	Unspecified OS/2 Warp (Intel x86) PowerPC	Macintosh	Unspecified Macintosh PowerPC	UNIX/X	Unspecified Native server Pseudo server	Unspecified	Unspecified	N	Info
OSMajorType	OSMinorType														
Windows	Unspecified Windows 3.1x Windows 95 Windows NT														
OS/2	Unspecified OS/2 Warp (Intel x86) PowerPC														
Macintosh	Unspecified Macintosh PowerPC														
UNIX/X	Unspecified Native server Pseudo server														
Unspecified	Unspecified														
protocolVersion	This capability specifies the protocol version level. The allowable value is 0x0200 (indicating major and minor versions of 2 and 0 respectively).	N	Info												
generalCompressionTypes	This capability is a set of bit flags itemizing which (if any) non-standard general compression schemes are supported by this ASCE. Interpretation of this field depends on the negotiated generalCompressionLevel capability (see below) as follows: • If the negotiated generalCompressionLevel capability is zero, then only the least significant bit flag of this field is valid (i.e., bit 0). • If the negotiated generalCompressionLevel capability is greater than zero, then all bit flags within this field are valid. See clause 8.3.2 for further information on non-standard general compression.	F	Min												
updateCapabilityFlag	This capability indicates whether this ASCE can receive the UpdateCapabilityPDU. A value of TRUE indicates that it can receive the UpdateCapabilityPDU and a value of FALSE indicates that it cannot. See clause 8.2.14 for further information.	L	One												
remoteUnshareFlag	This capability indicates whether this ASCE can receive an ApplicationPDU with the UnhostApplication action. A value of TRUE indicates that it can receive an ApplicationPDU with the UnhostApplication action and a value of FALSE indicates that it cannot. See clause 8.9 for further information.	L	One												

Table 8-3 – General capability set (legacy mode)

Capability	Description	Class	Rule
generalCompressionLevel	This capability indicates which level of general compression scheme handling is supported by this ASCE. See clause 8.3.2 for further information on non-standard general compression.	N	Min

Table 8-4 – General capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
remoteUnshareFlag	This capability indicates whether this ASCE can receive an ApplicationPDU with the UnhostApplication action. See clause 8.9 for further information. This capability must be advertised using non-collapsing capabilities.	1	L	One
v42bisCompressionFlag	This capability indicates whether this ASCE can receive ASPDUs general compressed using V.42 <i>bis</i> compression. See clause 8.3.2 for further information.	2	L	Min
v42bisNumberCodeWords (default: 512)	This capability specifies the total number of codewords to be used by the V.42 <i>bis</i> compression algorithm. This is an upper bound on V.42 <i>bis</i> parameter P1. [ITU-T V.42 <i>bis</i>] does not impose an upper limit on its value. See clause 8.3.2 for further information.	3	N	Min (Note)
v42bisMaxStringLength (default: 6)	This capability specifies the maximum string length input to the V.42 <i>bis</i> encoder. This is an upper bound on V.42 <i>bis</i> parameter P2. See clause 8.3.2 for further information.	4	N	Min (Note)
NOTE – The v42bisNumberCodeWords and v42bisMaxStringLength capabilities are dependent on the v42bisCompressionFlag capability. If the respective numberOfEntities parameter returned by the GCC-Application-Roster-Report indication for the v42bisNumberCodeWords or v42bisMaxStringLength capabilities is equal to the numberOfEntities parameter for the v42bisCompressionFlag capability, then the respective negotiated value is established; otherwise the default value of the capability is established.				

8.2.4 Bitmap capability set

The bitmap capability set provides capabilities for the bitmap-oriented characteristics of the issuing ASCE. See Tables 8-5 and 8-6.

Table 8-5 – Bitmap capability set (legacy mode)

Capability	Description	Class	Rule
preferredBitsPerPixel	This capability indicates this ASCE's preferred format for bitmap data. Allowable values are 1, 4 and 8.	N	Group (Note 1)

Table 8-5 – Bitmap capability set (legacy mode)

Capability	Description	Class	Rule
receive1BitPerPixelFlag	This capability indicates whether this ASCE can receive 1 bit-per-pixel bitmap data. An ASCE is required to be able to receive 1 bit-per-pixel bitmap and shall set this parameter to TRUE.	L	Group (Note 1)
receive4BitsPerPixelFlag	This capability indicates whether this ASCE can receive 4 bits-per-pixel bitmap data. A value of TRUE indicates that it can receive 4 bits-per-pixel bitmap data and a value of FALSE indicates that it cannot. Where an ASCE specifies that it can receive 4 bits-per-pixel, then it must also receive 1 bit-per-pixel.	L	Group (Note 1)
receive8BitsPerPixelFlag	This capability indicates whether this ASCE can receive 8 bits-per-pixel bitmap data. A value of TRUE indicates that it can receive 8 bits-per-pixel bitmap data and a value of FALSE indicates that it cannot. Where an ASCE specifies that it can receive 8 bits-per-pixel, then it must also receive 1 and 4 bits-per-pixel.	L	Group (Note 1)
desktopWidth	This capability specifies this ASCE's current desktop width in pixels.	N	Group (Note 2)
desktopHeight	This capability specifies this ASCE's current desktop height in pixels.	N	Group (Note 2)
desktopResizeFlag	This capability indicates whether this ASCE can receive UpdateCapabilityPDUs containing a bitmap capability set as a result of a peer ASCE desktop resize. A value of TRUE indicates that it can receive an UpdateCapabilityPDU containing a bitmap capability set and a value of FALSE indicates that it cannot. See clause 8.2.14 for further information.	L	One
bitmapCompressionFlag	This capability indicates whether this ASCE can receive compressed bitmap data in UpdatePDU (bitmap) and in cache bitmap orders. See clause 8.17 for further information on bitmap compression.	L	Min
NOTE 1 – The bits-per-pixel related capabilities are negotiated as a group to determine the sendingBitsPerPixel using the algorithm described in clause 8.2.4.1. NOTE 2 – The desktop width and desktop height capabilities are negotiated as a group to determine (among others) the size of the virtual desktop. See clause 8.2.4.2.			

Table 8-6 – Bitmap capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
preferredBitsPerPixel (default: 8)	This capability indicates this ASCE's preferred format for bitmap data. Allowable values are 1, 4 and 8. This capability must be advertised using non-collapsing capabilities.	10	N	Group (Note 1)

Table 8-6 – Bitmap capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
receive4BitsPerPixelFlag	This capability indicates whether this ASCE can receive 4 bits-per-pixel bitmap data. Where an ASCE specifies that it can receive 4 bits-per-pixel, then it must also be able to receive 1 bit-per-pixel.	11	L	Group (Note 1)
receive8BitsPerPixelFlag	This capability indicates whether this ASCE can receive 8 bits-per-pixel bitmap data. Where an ASCE specifies that it can receive 8 bits-per-pixel, then it must also be able to receive 1 and 4 bits-per-pixel.	12	L	Group (Note 1)
desktopWidth (default: 640)	This capability specifies this ASCE's current desktop width in pixels. This capability must be advertised using non-collapsing capabilities.	13	N	Group (Note 2)
desktopHeight (default: 480)	This capability specifies this ASCE's current desktop height in pixels. This capability must be advertised using non-collapsing capabilities.	14	N	Group (Note 2)
bitmapCompressionFlag	This capability indicates whether this ASCE can receive compressed bitmap data in UpdatePDU (bitmap) and in cache bitmap orders. See clause 8.17 for further information on bitmap compression.	15	L	Min
NOTE 1 – The bits-per-pixel related capabilities are negotiated as a group to determine the sendingBitsPerPixel using the algorithm described in clause 8.2.4.1.				
NOTE 2 – The desktop width and desktop height capabilities are negotiated as a group to determine (among others) the size of the virtual desktop. See clause 8.2.4.2.				

8.2.4.1 Sending bits-per-pixel capabilities negotiation

The bitmap capability set preferredBitsPerPixel, receive1BitPerPixelFlag (in legacy mode), receive4BitsPerPixelFlag and receive8BitsPerPixelFlag capabilities are negotiated as a group to determine the sendingBitsPerPixel used by each ASCE when sending bitmap data, palettes and ColorTables.

Certain terminal types can acquire and render bitmap data in a range of colour depths, but may have a preferred colour depth (which normally corresponds to the actual terminal display colour depth) at which acquisition and rendering is either more efficient or can be achieved with better colour fidelity. Therefore, the bitmap capability set provides a range of capabilities, which enables an ASCE to express:

- its preferred colour depth for the receipt of bitmap data, palettes and ColorTables;
- other colour depths at which it can receive bitmap data, palettes and ColorTables.

The capabilities negotiation algorithm for this group determines a sendingBitsPerPixel for a particular ASCE using the advertised preferredBitsPerPixel, receive4BitsPerPixelFlag and receive8BitsPerPixelFlag capability values, as follows:

- set combinedBitsPerPixel to the minimum of this ASCE's preferredBitsPerPixel value and the maximum of all other ASCEs' preferredBitsPerPixel values;
- if the combinedBitsPerPixel is 1 then set sendingBitsPerPixel to 1;

- else if combinedBitsPerPixel is less than or equal to 4 and the negotiated receive4BitsPerPixelFlag value is TRUE then sendingBitsPerPixel is 4;
- else if the negotiated receive8BitsPerPixelFlag value is TRUE then sendingBitsPerPixel is 8;
- else if the negotiated receive4BitsPerPixelFlag value is TRUE then sendingBitsPerPixel is 4;
- else sendingBitsPerPixel is 1.

In the legacy mode of the AS protocol, the bitmap capability set contains a receive1BitPerPixelFlag capability, but this does not play a part in the negotiation, as the requirement to support lower colour depths (i.e., an ASCE is required to support at least 1 bit-per-pixel if it supports 4 bits-per-pixel and to support 1 and 4 bits-per-pixel if it supports 8 bits-per-pixel) means that all ASCEs must support 1 bit-per-pixel. In the base mode of the AS protocol, the bitmap capability set does not contain a receive1BitPerPixelFlag capability.

The requirement for an ASCE to support lower colour depths means that an ASCE conforming to this Recommendation shall not cause capabilities negotiation to fail to find a suitable value.

An ASCE may use an alternate private algorithm in certain circumstances. For example, certain 8 bits-per-pixel terminals do not reliably generate 4 bits-per-pixel bitmap data. Where this is the case and the negotiated receive8BitsPerPixel value is TRUE, the ASCE may set sendingBitsPerPixel to 8, even where the combinedBitsPerPixel value is 4. However, where an ASCE does use a private algorithm, it shall still generate a sendingBitsPerPixel value which is consistent with the negotiated values of the receive4BitsPerPixelFlag and receive8BitsPerPixelFlag capabilities.

8.2.4.2 Desktop size capabilities negotiation

The bitmap capability set desktop width and desktop height capabilities are independently negotiated to determine the size of the virtual desktop. See clause 5.2.5 for further information on the virtual desktop.

The capabilities negotiation algorithm essentially uses the max rule (see clause 8.2.2), but the candidate values are the capability values advertised by all active hosting ASCEs, which may or may not include the determining ASCE.

For example, in a conference with four active ASCEs – ASCEs A, B, C and D where:

- ASCE A advertises Bitmap.desktopWidth and Bitmap.desktopHeight as 800 and 600 respectively;
- ASCE B advertises Bitmap.desktopWidth and Bitmap.desktopHeight as 1024 and 768 respectively;
- ASCE C advertises Bitmap.desktopWidth and Bitmap.desktopHeight as 1600 and 1200 respectively;
- ASCE D advertises Bitmap.desktopWidth and Bitmap.desktopHeight as 640 and 480 respectively;
- then if ASCEs A and C are hosting, the negotiated virtual desktop capability values are 1600 by 1200;
- whereas if ASCEs B and D are hosting, the negotiated virtual desktop capability values are 1024 by 768.

8.2.5 Order capability set

The order capability set provides capabilities for the order characteristics of the issuing ASCE.

See Appendix I for informative values for the desktop save capabilities (i.e., desktopSaveXGranularity, desktopSaveYGranularity and desktopSaveSize). See Tables 8-7 and 8-8.

Table 8-7 – Order capability set (legacy mode)

Capability	Description	Class	Rule																											
terminalDescriptor	This capability is a null-terminated character T.50 text string which may be used to identify local terminal characteristics for information and diagnostics purposes.	S	Info																											
desktopSaveXGranularity	This capability specifies the minimum X granularity in pixels for this ASCE when receiving desktop save orders. See clause 8.16.17 for further information.	N	Max (Note 1)																											
desktopSaveYGranularity	This capability specifies the minimum Y granularity in pixels for this ASCE when receiving desktop save orders. See clause 8.16.17 for further information.	N	Max (Note 1)																											
maximumOrderLevel	This capability specifies the maximum order level supported within the orderSupport capability (see below). See clause 8.2.6 for further information on order levels.	N	Info																											
numberFonts	This capability is the maximum number of matchable fonts for this ASCE, details of which are subsequently supplied in the FontPDU. See clause 8.8 for further information on the FontPDU.	N	Info (Note 2)																											
orderFlags	This capability is a set of bit flags indicating the order support provided by this ASCE. Defined bit flag values are as follows. - Negotiate order support: It is mandatory to set this flag. - Cannot receive orders: If this flag is set, it indicates that this ASCE cannot receive orders. An ASCE shall always set the negotiate order support flag. See clause 8.16.3 for further information on order encoding.	F	Min (Note 3)																											
orderSupport	This capability is an array of 32 order levels indexed by order type. The allowable array indices are as follows. All other array values shall be set to zero. See below for further information on order levels. <table><thead><tr><th>Order</th><th>Index</th><th></th></tr></thead><tbody><tr><td>Destination Blt support</td><td>0</td><td>See clause 8.16.4.</td></tr><tr><td>Pattern Blt support</td><td>1</td><td>See clause 8.16.5.</td></tr><tr><td>Screen Blt support</td><td>2</td><td>See clause 8.16.6.</td></tr><tr><td>Memory Blt support</td><td>3</td><td>See clause 8.16.9.</td></tr><tr><td>Memory three-way Blt support</td><td>4</td><td>See clause 8.16.10.</td></tr><tr><td>Text support</td><td>5</td><td>See clause 8.16.11.</td></tr><tr><td>Extended text support</td><td>6</td><td>See clause 8.16.12.</td></tr><tr><td>Rectangle support</td><td>7</td><td>See clause 8.16.14.</td></tr></tbody></table>	Order	Index		Destination Blt support	0	See clause 8.16.4.	Pattern Blt support	1	See clause 8.16.5.	Screen Blt support	2	See clause 8.16.6.	Memory Blt support	3	See clause 8.16.9.	Memory three-way Blt support	4	See clause 8.16.10.	Text support	5	See clause 8.16.11.	Extended text support	6	See clause 8.16.12.	Rectangle support	7	See clause 8.16.14.	N	Min
Order	Index																													
Destination Blt support	0	See clause 8.16.4.																												
Pattern Blt support	1	See clause 8.16.5.																												
Screen Blt support	2	See clause 8.16.6.																												
Memory Blt support	3	See clause 8.16.9.																												
Memory three-way Blt support	4	See clause 8.16.10.																												
Text support	5	See clause 8.16.11.																												
Extended text support	6	See clause 8.16.12.																												
Rectangle support	7	See clause 8.16.14.																												

Table 8-7 – Order capability set (legacy mode)

Capability	Description	Class	Rule
	Line support 8 See clause 8.16.16. Frame support 9 See clause 8.16.13. Opaque rectangle 10 See clause 8.16.15. support Desktop save support 11 See clause 8.16.17.		
textFlags	This capability is a set of bit flags indicating font matching and text options supported by this ASCE. Defined bit flag values are as follows: • Check font aspect If this flag is set, it indicates that this ASCE supports the checking of font horizontal and vertical aspects during font matching. • Check font signatures If this flag is set, it indicates that this ASCE supports the checking of font signatures during font matching. • DeltaX simulation If this flag is set, it indicates that this ASCE allows Delta X approximations during font matching. • Baseline start If this flag is set, it indicates that this ASCE can receive text and extended text orders where the text start position is specified with respect to the character baseline. See clause 8.8 for further information on font matching. See clause 8.16.11 and clause 8.16.12 for further information on text and extended text orders.	F	Min
desktopSaveSize	This capability specifies the total size in pixels of this ASCE's desktop cache per hosting ASCE. See clause 8.16.17 for further information.	N	Min (Note 1)

NOTE 1 – As a result of capabilities negotiation, an ASCE must be able to receive desktop save orders constructed using X and/or Y granularities larger than, or not exact multiples of, its advertised desktopSaveXGranularity and desktopSaveYGranularity capabilities. Similarly, an ASCE must be able to send desktop save orders where the negotiated desktopSaveSize is not an exact multiple of its advertised desktopSaveXGranularity and desktopSaveYGranularity capabilities.

NOTE 2 – A receiving ASCE may use the numberFonts capability to determine whether it has sufficient storage to receive this ASCE's fonts on subsequent FontPDUs. The numberFonts capability is an upper limit and subsequent FontPDUs may contain a number of font attributes less than or equal to the numberFonts capability. See clause 8.8 for further information on the FontPDU.

NOTE 3 – It is mandatory that the negotiate order support bit flag is set. The cannot receive orders bit flag allows an ASCE to advertise whether it can or cannot receive orders. Where the cannot receive orders bit flag is not set, it indicates that the ASCE can receive orders. Where the cannot receive orders bit flag is set, it indicates that the ASCE cannot receive orders, which forces peer ASCEs to disable the sending of order updates entirely and to just send bitmap updates.

Table 8-8 – Order capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
desktopSaveXGranularity (default: 1)	This capability specifies the minimum X granularity in pixels for this ASCE when receiving desktop save orders. See clause 8.16.17 for further information.	20	N	Max (Note)
desktopSaveYGranularity (default: 1)	This capability specifies the minimum Y granularity in pixels for this ASCE when receiving desktop save orders. See clause 8.16.17 for further information.	21	N	Max (Note)
desktopSaveSize (default: 160,000)	This capability specifies the total size in pixels of this ASCE's desktop cache. See clause 8.16.17 for further information.	22	N	Min (Note)
checkFontAspectFlag	This capability indicates whether this ASCE supports the checking of font horizontal and vertical aspects during font matching. See clause 8.8 for further information on font matching.	23	L	Min
checkFontSignaturesFlag	This capability indicates whether this ASCE supports the checking of font signatures during font matching. See clause 8.8 for further information on font matching.	24	L	Min
allowDeltaXFlag	This capability indicates whether this ASCE allows delta X approximations during font matching. See clause 8.8 for further information on font matching.	25	L	Min
baselineStartFlag	This capability indicates whether this ASCE can receive text and extended text orders where the text start position is specified with respect to the character baseline. See clause 8.16.11 and clause 8.16.12 for further information on text and extended text orders.	26	L	Min
receiveOrdersFlag	This capability indicates whether this ASCE can receive UpdatePDU (orders) ASPDUs – i.e., whether it can receive orders or not. See clause 8.16 for further information on orders.	27	L	Min
DesktopSaveLevel (default: 1)	This capability indicates the desktop save order level supported by this ASCE. See clause 8.16.17 for further information on the desktop save order.	30	N	Min
DestinationBlitLevel (default: 1)	This capability specifies the destination Blt order level supported by this ASCE. See clause 8.2.6 for further information on order levels and 8.16.4 for further information on the destination Blt order.	31	N	Min

Table 8-8 – Order capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
ExtendedTextLevel (default: 1)	This capability specifies the extended text order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.12 for further information on the extended text order.	32	N	Min
FrameLevel (default: 1)	This capability specifies the frame order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.13 for further information on the frame order.	33	N	Min
LineLevel (default: 1)	This capability specifies the line order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.16 for further information on the line order.	34	N	Min
MemoryBltLevel (default: 1)	This capability specifies the memory Blt order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.9 for further information on the memory Blt order.	35	N	Min
MemoryThreeWayBltLevel (default: 1)	This capability specifies the memory three-way Blt order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.10 for further information on the memory three-way Blt order.	36	N	Min
OpaqueRectangleLevel (default: 1)	This capability specifies the opaque rectangle order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.15 for further information on the opaque rectangle order.	37	N	Min
PatternBltLevel (default: 1)	This capability specifies the pattern Blt order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.5 for further information on the pattern Blt order.	38	N	Min
RectangleLevel (default: 1)	This capability specifies the rectangle order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.14 for further information on the rectangle order.	39	N	Min
ScreenBltLevel (default: 1)	This capability specifies the screen Blt order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.6 for further information on the screen Blt order.	40	N	Min

Table 8-8 – Order capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
TextLevel (default: 1)	This capability specifies the text order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.11 for further information on the text order.	41	N	Min
DesktopOriginLevel (default: 1)	This capability specifies the desktop origin order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.18 for further information on the desktop origin order.	42	N	Min
CacheBitmapLevel (default: 1)	This capability specifies the cache bitmap order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.7 for further information on the cache bitmap order.	43	N	Min
CacheColorTableLevel (default: 1)	This capability specifies the cache ColorTable order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.8 for further information on the cache ColorTable order.	44	N	Min
ColorSpaceLevel (default: 1)	This capability specifies the colour space order level supported by this ASCE. See clause 8.2.6 for further information on order levels and clause 8.16.19 for further information on the colour space order.	45	N	Min
NOTE – As a result of capabilities negotiation, an ASCE must be able to receive desktop save orders constructed using X and/or Y granularities larger than, or not exact multiples of, its advertised desktopSaveXGranularity and desktopSaveYGranularity capabilities. Similarly, an ASCE must be able to send desktop save orders where the negotiated desktopSaveSize is not an exact multiple of its advertised desktopSaveXGranularity and desktopSaveYGranularity capabilities.				

8.2.6 Order levels

The AS protocol allows for future enhancement of order support using order levels. Order level values are in the range 0..255, with the value 0 indicating that the indicated order is not supported.

The order level values in the Order.orderSupport capability (for the legacy mode of the AS protocol) and in the per-order level capability parameters (for the base mode of the AS protocol) indicate the maximum order level that the issuing ASCE can receive on an order-by-order basis. If an ASCE indicates that it can receive order level N for a particular order, then it shall be able to receive order levels in the range 1..N. It is envisaged that as additional application sharing requirements emerge (e.g., reflecting changing order usage in target terminals), then existing AS orders may be enhanced. Where that is the case, an ASCE may advertise support for a range of enhanced orders and, where it is in a conference with a group of active ASCEs that also support the enhanced orders, may use them when constructing UpdatePDU (orders) ASPDUs. In summary, this means that:

- where an ASCE supports an order at order level N, it shall be prepared to receive orders at order levels 1..N;

- where an ASCE supports an order at order level N and the negotiated order level is greater than or equal to N, then the ASCE may send the order at order levels 1..N;
- where an ASCE supports an order at order level N and the negotiated order level is less than N, then the ASCE may send the order at order levels 1 through the negotiated value – but may not send orders at order levels greater than the negotiated value.

For the legacy mode of the AS protocol, order level support is not defined for the desktop origin, cache bitmap and cache ColorTable orders (see clauses 8.16.18, 8.16.7 and 8.16.8, respectively). In the absence of order level support for these orders:

- if an ASCE supports the screen Blt order at order level 1 or above, it shall also support the desktop origin order;
- if an ASCE supports either of the memory Blt and memory three-way Blt orders at order level 1 or above, it shall also support the cache bitmap and cache ColorTable orders and shall ensure that it advertises valid values for the bitmap cache and ColorTable cache capability sets (see clauses 8.2.7 and 8.2.8 for further information on the bitmap cache and ColorTable cache capability sets).

8.2.7 Bitmap cache capability set

The bitmap cache capability set provides capabilities for the bitmap cache characteristics of the issuing ASCE. These capabilities are used to negotiate values used to construct cache bitmap orders in UpdatePDUs. See clause 8.16.7 for further information on cache bitmap orders.

If an ASCE supports either of the memory Blt and memory three-way Blt orders at order level 1 or above, it shall support the cache bitmap order (in base mode at order level 1 or above) and ensure that it advertises bitmap cache capability set values with:

- non-zero values for the cache1Entries, cache2Entries and cache3Entries capabilities;
- allowable values for the cache1MaximumCellSize, cache2MaximumCellSize and cache3MaximumCellSize capabilities (as specified in Tables 8-9 and 8-10);
- $\text{cache3MaximumCellSize} \geq \text{cache2MaximumCellSize} \geq \text{cache1MaximumCellSize}$.

Where an ASCE supports bitmap caching, these capabilities indicate the bitmap cache sizes per each other hosting ASCE. That is, by advertising these capabilities, the ASCE is committing to provide a set of bitmap caches of the advertised sizes for each other active ASCE in the conference that is hosting windows.

See Appendix I for informative values for bitmap cache capabilities.

Table 8-9 – Bitmap cache capability set (legacy mode)

Capability	Description	Class	Rule
cache1Entries	This capability is the number of cache entries in the first cache area.	N	Min
cache1MaximumCellSize	This capability is the maximum cell size in octets for the first cache area. The value for this capability is in the range 256..16384.	N	Min
cache2Entries	This capability is the number of cache entries in the second cache area.	N	Min
cache2MaximumCellSize	This capability is the maximum cell size in octets for the second cache area. The value for this capability is in the range 256..16384.	N	Min
cache3Entries	This capability is the number of cache entries in the third cache area.	N	Min

Table 8-9 – Bitmap cache capability set (legacy mode)

Capability	Description	Class	Rule
cache3MaximumCellSize	This capability is the maximum cell size in octets for the third cache area. The value for this capability is in the range 256..16384.	N	Min

Table 8-10 – Bitmap cache capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
cache1Entries (default: 600)	This capability is the number of cache entries in the first cache area.	80	N	Min
cache1MaximumCellSize (default: 496)	This capability is the maximum cell size in octets for the first cache area. The value for this capability is in the range 256..16384.	81	N	Min
cache2Entries (default: 300)	This capability is the number of cache entries in the second cache area.	82	N	Min
cache2MaximumCellSize (default: 2032)	This capability is the maximum cell size in octets for the second cache area. The value for this capability is in the range 256..16384.	83	N	Min
cache3Entries (default: 150)	This capability is the number of cache entries in the third cache area.	84	N	Min
cache3MaximumCellSize (default: 4080)	This capability is the maximum cell size in octets for the third cache area. The value for this capability is in the range 256..16384.	85	N	Min

8.2.8 ColorTable cache capability set

The ColorTable cache capability set provides capabilities for the ColorTable cache characteristics of the issuing ASCE. These capabilities are used to negotiate values used to construct cache ColorTable orders in UpdatePDUs. See clause 8.16.8 for further information on cache ColorTable orders and also Tables 8-11 and 8-12.

If an ASCE supports either of the memory Blt and memory three-way Blt orders at order level 1 or above, it shall support the cache ColorTable order (in base mode at order level 1 or above) and ensure that it advertises a ColorTable cache capability set containing a non-zero value for the colorTableCacheSize capability.

Where an ASCE supports ColorTable caching, these capabilities indicate the ColorTable cache size per each other hosting ASCE. That is, by advertising these capabilities, the ASCE is committing to provide a ColorTable cache of the advertised size for each other active ASCE in the conference that is hosting windows.

See Appendix I for informative values for ColorTable cache capability set values.

Table 8-11 – ColorTable cache capability set (legacy mode)

Capability	Description	Class	Rule
colorTableCacheSize	This capability specifies the number of ColorTable entries in this ASCE's receiving ColorTable cache. Where an ASCE supports ColorTable caching, the allowable values are in the range 1..255 (zero is not allowed). See clause 8.16.8 for further information on ColorTable caching.	N	Min

Table 8-12 – ColorTable cache capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
colorTableCacheSize (default: 6)	This capability specifies the number of ColorTable entries in this ASCE's receiving ColorTable cache. Where an ASCE supports ColorTable caching, the allowable values are in the range 1..255 (zero is not allowed). See clause 8.16.8 for further information on ColorTable caching.	90	N	Min

8.2.9 Window activation capability set

The window activation capability set provides capabilities for the window activation characteristics of the issuing ASCE, and in particular about its support for specific WindowActivationPDU activation messages. See clause 8.11 for further information on WindowActivationPDUs and also Tables 8-13 and 8-14.

Table 8-13 – Window activation capability set (legacy mode)

Capability	Description	Class	Rule
helpKeyFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the ActivationHelpKey action. A value of TRUE indicates that it can receive WindowActivationPDUs containing the ActivationHelpKey action and a value of FALSE indicates that it cannot. See clause 8.11 for further information.	L	One
helpIndexKeyFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the ActivationHelpIndexKey action. A value of TRUE indicates that it can receive WindowActivationPDUs containing the ActivationHelpIndexKey action and a value of FALSE indicates that it cannot. See clause 8.11 for further information.	L	One
helpExtendedKeyFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the ActivationHelpExtendedKey action. A value of TRUE indicates that it can receive WindowActivationPDUs containing the ActivationHelpExtendedKey action and a value of FALSE indicates that it cannot. See clause 8.11 for further information.	L	One

Table 8-13 – Window activation capability set (legacy mode)

Capability	Description	Class	Rule
windowManagerMenuFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the WindowManagerMenu action. A value of TRUE indicates that it can receive WindowActivationPDUs containing the WindowManagerMenu action and a value of FALSE indicates that it cannot. See clause 8.11 for further information.	L	One

Table 8-14 – Window activation capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
helpKeyFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the ActivationHelpKey action. See clause 8.11 for further information. This capability must be advertised using non-collapsing capabilities.	100	L	One
helpIndexKeyFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the ActivationHelpIndexKey action. See clause 8.11 for further information. This capability must be advertised using non-collapsing capabilities.	101	L	One
helpExtendedKeyFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the ActivationHelpExtendedKey action. See clause 8.11 for further information. This capability must be advertised using non-collapsing capabilities.	102	L	One
windowManagerMenuFlag	This capability indicates whether this ASCE can receive WindowActivationPDUs containing the WindowManagerMenu action. See clause 8.11 for further information. This capability must be advertised using non-collapsing capabilities.	103	L	One

8.2.10 Control capability set

The control capability set provides capabilities for the control characteristics of the issuing ASCE. These capabilities are used to negotiate values used in the management of control and detached status between ASCEs. See clause 8.12 for further information on control characteristics and also Tables 8-15 and 8-16.

Table 8-15 – Control capability set (legacy mode)

Capability	Description	Class	Rule
controlFlags	This capability is a set of bit flags indicating the control options supported by this ASCE. Defined bit flag values are as follows: Allow mediated control If an ASCE does not set the allow mediated control bit flag capability, the ASCE does not support the AS mediated control protocol. Where this is the case, the remoteDetachFlag, controlInterest and detachInterest capabilities need not be negotiated and the MediatedControlPDU is not valid. See clause 8.13 for further information.	F	Min
remoteDetachFlag	This capability indicates whether this ASCE allows other ASCEs to force it into detached control mode. A value of TRUE indicates that this ASCE allows other ASCEs to force it into detached control mode and a value of FALSE indicates that it does not. See clause 8.13 for further information.	L	One
controlInterest	This capability indicates this ASCE's behaviour for changes of control. Allowable values are as follows: - Always This ASCE always permits changes of control. - Confirm This ASCE requires to confirm changes of control. - Never This ASCE never allows changes of control. See clause 8.13 for further information.	N	Max (Note)
detachInterest	This capability indicates this ASCE's behaviour for changes in detach status. Allowable values are as follows: - Always This ASCE always permits changes in detach status. - Confirm This ASCE requires to confirm changes in detach status. - Never This ASCE never allows changes in detach status. See clause 8.13 for further information.	N	Max (Note)
NOTE – The controlInterest and detachInterest capabilities are negotiated using values such that never is greater than confirm which is greater than always.			

Table 8-16 – Control capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
mediatedControlFlag	This capability indicates whether the issuing ASCE supports the mediated control protocol. Where this capability is not supported, the remoteDetachFlag, controlInterest and detachInterest capabilities need not be negotiated and the MediatedControlPDU is not valid. See clause 8.13 for further information.	110	L	Min

Table 8-16 – Control capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
remoteDetachFlag	This capability indicates whether this ASCE allows other ASCEs to force it into detached control mode. See clause 8.13 for further information. This capability must be advertised using non-collapsing capabilities.	111	L	One
controlInterest (default: always)	This capability indicates this ASCE's behaviour for changes of control. Allowable values are as follows. • Always This ASCE always permits changes of control. • Confirm This ASCE requires to confirm changes of control. • Never This ASCE never allows changes of control. See clause 8.13 for further information. This capability must be advertised using non-collapsing capabilities.	112	N	Max (Note)
detachInterest (default: always)	This capability indicates this ASCE's behaviour for changes in detach status. Allowable values are as follows: • Always This ASCE always permits changes in detach status. • Confirm This ASCE requires to confirm changes in detach status. • Never This ASCE never allows changes in detach status. See clause 8.13 for further information. This capability must be advertised using non-collapsing capabilities.	113	N	Max (Note)
NOTE – The controlInterest and detachInterest capabilities are negotiated using values such that never is greater than confirm which is greater than always.				

8.2.11 Pointer capability set

The pointer capability set provides capabilities for the pointer characteristics of the issuing ASCE. These capabilities are used to negotiate values used to construct PointerPDUs. See clause 8.14 for further information on pointers and also Tables 8-17 and 8-18.

Where an ASCE does not support colour pointers, or where it supports colour pointers but does not wish to support colour pointer caching, then it should advertise its Pointer.colorPointerCacheSize as one. This shall be interpreted as indicating that the advertising ASCE can only remember the last monochrome and/or colour pointer (depending on the negotiated value of the Pointer.colorPointerFlag) – that is, it does not support pointer caching.

Where an ASCE supports colour pointers and colour pointer caching, these capabilities indicate the colour pointer cache size per each other hosting ASCE. That is, by advertising these capabilities, the ASCE is committing to provide a colour pointer cache of the advertised size for each other active ASCE in the conference that is hosting windows.

See Appendix I for informative values for pointer capabilities.

Table 8-17 – Pointer capability set (legacy mode)

Capability	Description	Class	Rule
colorPointerFlag	This capability indicates whether this ASCE supports colour pointers. A value of TRUE indicates that this ASCE supports colour pointers and a value of FALSE indicates that it does not. See clause 8.14 for further information.	L	Min
colorPointerCacheSize	This capability specifies the number of entries in this ASCE's receiving colour pointer cache. The allowable values are in the range 1..500. Where an ASCE does not support colour pointers, it shall set this capability to 1 (a value of zero is not allowed). See clause 8.14 for further information on pointer caching.	N	Min

Table 8-18 – Pointer capability set (base mode)

Capability (default value)	Description	ID	Class	Rule
colorPointerFlag	This capability indicates whether this ASCE supports colour pointers. See clause 8.14 for further information.	120	L	Min
colorPointerCacheSize (default: 25)	This capability specifies the number of entries in this ASCE's receiving colour pointer cache. The allowable values are in the range 1..500. Where an ASCE does not support colour pointers, it shall set this capability to 1 (a value of zero is not allowed). See clause 8.14 for further information on pointer caching.	121	N	Min

8.2.12 Share capability set

The share capability set provides information about the node on which the ASCE executes. This capability set is only supported in the legacy mode of the AS protocol.

The GCC node ID may be used by receiving ASCEs to correlate issuing ASCEs with their application records in the GCC application roster. See Table 8-19.

Table 8-19 – Share capability set (legacy mode)

Capability	Description	Class	Rule
nodeID	This capability is the GCC node ID of the advertising ASCE's node.	N	Info

8.2.13 Non-standard capability set

The non-standard capability set is used to negotiate non-standard functions. Any number of non-standard capabilities may appear in the non-standard capability set as long as they each have a unique Non-Standard-Identifier. Interpretation of these capabilities is not covered by this Recommendation.

The non-standard capability set is only supported in the base mode of the AS protocol. There is no mechanism in the legacy mode of the AS protocol for the negotiation of non-standard capabilities. See Table 8-20.

Table 8-20 – Non-standard capability set (base mode)

Capability	Description	ID	Class	Rule
nonStandardCapability	This capability is used to negotiate non-standard functions. Any number of these may appear in the ASCE base mode capabilities 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	–	–

8.2.14 Capability update

For the base mode of the AS protocol, where an ASCE's capabilities change while enrolled, it shall re-enrol by issuing a GCC-Application-Enroll request to enrol in the standard base session with the enroll/unenroll flag set to enroll, including the revised combined capabilities. See clause 7 for further information on session enrolment and clause 8.2.1 for further information on the distribution of capabilities.

For the legacy mode of the AS protocol, an ASCE may advertise changes in a capability set by sending an UpdateCapabilityPDU to other ASCEs within the conference in the manner indicated in Table 6-3. The content of the UpdateCapabilityPDU is shown in Table 8-21.

An ASCE shall only send the UpdateCapabilityPDU where both the negotiated General.updateCapabilityFlag and Bitmap.desktopResizeFlag capabilities are TRUE.

The only allowable capability set that can be advertised in an UpdateCapabilityPDU is the legacy mode bitmap capability set (see clause 8.2.4).

Table 8-21 – UpdateCapabilityPDU

Parameter	Description
ShareData header	The ShareData header is described in clause 8.3.
updateCapabilitySet	This parameter shall be a bitmap capability set. See clause 8.2.4 for a description of the bitmap capability set.

When an ASCE determines that one or more capabilities have been updated, it shall perform any required capabilities negotiation (see clause 8.2.2) and shall perform hosting synchronization (if it is hosting) and shadow synchronization (see clause 8.6).

8.3 ASPDU formats

In the legacy mode of the AS protocol, all ASPDUs, other than the following flow control ASPDUs, contain a ShareControl header. Table 8-22 describes the ShareControl header for the legacy mode of the AS protocol. In the base mode of the AS protocol, ASPDUs do not contain ShareControl headers.

- FlowTestPDU: See clause 8.5.
- FlowResponsePDU: See clause 8.5.
- FlowStopPDU: See clause 8.5.

Table 8-22 – ShareControl header (legacy mode)

Parameter	Description
totalLength	This is the total length in octets of the ASPDU within which this header is included. This parameter is required as MCS implementations may segment ASPDUs in transmission and are not required to reassemble on delivery. This parameter allows receiving ASCEs to efficiently perform reassembly where MCS segmentation is present.
protocolVersion	This parameter identifies the protocol version supported by the issuing ASCE. The allowable value is 1.
PDUSource	This parameter is the MCS user ID of the ASCE sending the ASPDU containing this ShareControl header.

All ASPDUs, other than the following ASCE activation and flow control ASPDUs, contain a ShareData header and are referred to as data ASPDUs.

- DemandActivePDU: See clause 8.4.1.
- ConfirmActivePDU: See clause 8.4.1.
- RequestActivePDU: See clause 8.4.1.
- DeactivateOtherPDU: See clause 8.4.1.
- DeactivateSelfPDU: See clause 8.4.1.
- DeactivateAllPDU: See clause 8.4.1.
- FlowTestPDU: See clause 8.5.
- FlowResponsePDU: See clause 8.5.
- FlowStopPDU: See clause 8.5.

Tables 8-23 and 8-24 describe the ShareData header for the legacy and base modes of the AS protocol.

Table 8-23 – ShareData header (legacy mode)

Parameter	Description
ShareControl header	The ShareControl header is described in clause 8.3.
shareID	This parameter uniquely identifies the ASCE session within which this ASPDU is issued. See clause 8.4.2 for a description of share identifiers.
streamID	This parameter identifies the stream for this ASPDU. See clause 8.3.1 for a description of streams.
uncompressedLength	This parameter is the length of the uncompressed ASPDU data in octets, starting from and including the generalCompressedType parameter. This parameter may be used as a check on decompression. See clause 8.3.2 for further information on non-standard general compression.

Table 8-23 – ShareData header (legacy mode)

Parameter	Description
generalCompressedType	This parameter indicates whether the issuing ASCE has general compressed the ASPDU containing this ShareData header – and if so, indicates which general compression scheme type has been applied. Interpretation of this field depends on the negotiated General.generalCompressionLevel capability as follows: • If the negotiated General.generalCompressionLevel capability is zero, then the ASCE shall only reference the most significant bit of this field. • If the negotiated General.generalCompressionLevel capability is greater than zero, then the ASCE shall reference all bits within this field. A value of zero indicates that no general compression has been applied. A non-zero value specifies the particular compression type that has been applied. See clause 8.3.2 for further information on general compression.
generalCompressedLength	Where the generalCompressedType is non-zero, this parameter is the length of the compressed ASPDU data in octets, starting from after the uncompressedLength parameter. Where the generalCompressedType is zero, indicating that no general compression has been applied, this parameter shall be zero. See clause 8.3.2 for further information on general compression.

Table 8-24 – ShareData header (base mode)

Parameter	Description
shareID	This parameter is the roster instance of the ASCE session within which this ASPDU is issued. See clause 8.4.2 for a description of share identifiers.
generalCompressionSpecifier	This optional parameter identifies the general compression scheme and any associated parameters applicable to this ASPDU. See clause 8.3.2 for further information.

8.3.1 Streams

In the legacy mode of the AS protocol, an ASCE is required to send data ASPDUs over a set of logical streams. Streams are used to represent ASCE data priorities and map onto MCS priorities as described in Table 8-25. Stream identifiers are present in all legacy mode data ASPDUs. Streams are not supported in the base mode of the AS protocol.

Table 8-25 – Stream identifiers (legacy mode)

Stream ID	MCS priority
4	High
2	Medium
1	Low

8.3.2 General compression

This Recommendation specifies compression schemes that may be applied to orders (which is termed order encoding – see clause 8.16.3) and bitmap data (which is termed bitmap compression – see clause 8.17) in UpdatePDUs. Both these compression schemes deliver good compression ratios on typical AS output data for relatively low processor loads.

However, where an ASCE experiences flow control back pressure (see clause 8.5) – for example, when sending over slower links such as typical PSTN lines, or when sending over congested higher speed links – experience shows that significant performance enhancements may be achieved by the application of a dictionary-based compression scheme to selected ASPDUs – including already compressed data in UpdatePDU (orders) and UpdatePDU (bitmap) ASPDUs. This approach is termed general compression.

8.3.2.1 Legacy mode general compression

This Recommendation does not specify a general compression mechanism for legacy mode, but rather provides a compression scheme-independent mechanism whereby ASCEs can negotiate and use mutually agreed general compression schemes, as follows.

An ASCE shall negotiate the general compression scheme using the General.generalCompressionLevel and General.generalCompressionTypes capabilities. See clause 8.2.3 for further information on the general capability set. The GeneralCompressionLevel capability indicates which level of general compression is supported by an ASCE while the General.generalCompressionTypes capability is a set of bit flags, allowing an ASCE to indicate that it supports none, one or a selection of general compression schemes.

Note that the interpretation of the General.generalCompressedTypes capability is dependent on the negotiated General.generalCompressionLevel capability. Where the negotiated General.generalCompressionLevel capability is zero, then only the least significant bit flag of the General.generalCompressedTypes capability shall be considered. Where the negotiated generalCompressionLevel capability is greater than zero, then all bit flags within the General.generalCompressedTypes capability shall be considered.

Where an ASCE determines that other peer ASCEs can receive a particular general compression scheme, it may apply general compression to data ASPDUs on a selective basis – that is, an ASCE may apply general compression to none, some or all ASPDUs. The criteria by which an ASCE determines whether general compression is appropriate for a particular data ASPDU may include flow control status (see clause 8.5) and the size of the particular ASPDU. This Recommendation does not recommend a particular approach to the application of general compression – it is entirely a local matter for each ASCE.

Where an ASCE does not apply general compression to a data ASPDU, either because no suitable general compression scheme is available after capabilities negotiation, or where a suitable general compression scheme has been negotiated but the sending ASCE determines that general compression is inappropriate, the ASCE shall set the generalCompressedType to zero to indicate that the ASPDU is not general compressed, the uncompressedLength parameter to the ASPDU octet length and the generalCompressedLength parameter to zero. Where an ASCE applies general compression to a data ASPDU, it shall set the generalCompressedType to a value corresponding to one of the negotiated general compression schemes to indicate that the ASPDU is general compressed, the uncompressedLength parameter to the ASPDU octet length before compression and the generalCompressedLength parameter to the ASPDU octet length after compression. Setting both the uncompressedLength and generalCompressedLength parameters allows a receiving ASCE to check that the ASPDU length after decompression is consistent with its original uncompressed length.

Note that the interpretation of the `generalCompressedType` parameter in the `ShareData` header is dependent on the negotiated `General.generalCompressionLevel` capability. Where the negotiated `General.generalCompressionLevel` capability is zero, then only the most significant bit of the `General.generalCompressedType` parameter shall be considered. Where the negotiated `General.generalCompressionLevel` capability is greater than zero, then all bits of the `General.generalCompressedType` parameter shall be considered.

See Appendix I for informative values and information on legacy mode general compression.

8.3.2.2 Base mode general compression

This Recommendation specifies the optional use of [ITU-T V.42 *bis*] for general compression in base mode.

Where an ASCE determines that other peer ASCEs can receive ASPDUs general compressed using V.24 *bis*, it may apply V.42 *bis* general compression to data ASPDUs on a selective basis – that is, an ASCE may apply V.42 *bis* general compression to none, some or all ASPDUs. The criteria by which an ASCE determines whether V.42 *bis* general compression is appropriate for a particular data ASPDU may include flow control status (see clause 8.5) and the size of the particular ASPDU. This Recommendation does not recommend a particular approach to the application of V.42 *bis* general compression – it is entirely a local matter for each ASCE.

Where V.42 *bis* general compression is applied, the encoder is initialized during the sender's hosting synchronization and the decoder is initialized during the receivers' shadow synchronization (see clauses 8.6.2 and 8.6.3).

Where an ASCE does apply V.42 *bis* general compression to a particular data ASPDU, it shall set the `generalCompressionSpecifier` parameter to indicate that the ASPDU is V.42 *bis* general compressed. If the ASPDU is the first V.42 *bis* general compressed ASPDU after a V.42 *bis* encoder initialization, then the ASCE may also include the optional V.42 *bis* P1 and P2 parameters in the general `CompressionSpecifier` to allow peer ASCEs to optimize general compression-related resources.

8.4 ASCE activation

Within an AS session, an ASCE may be in one of the following three activation states:

- An inactive ASCE is not participating in the sharing of windows (i.e., it is not hosting windows, drawing shadow windows, or sending or receiving AS input). When an ASCE is inactive, it need not retain local resources that are required for hosting and/or sharing.
- A pending active ASCE is in the process of becoming active and is not participating in the sharing of windows (i.e., it is not hosting windows, drawing shadow windows, or sending or receiving AS input).
- An active ASCE is cooperating with other active ASCEs to share windows.

The following clauses describe ASCE activation and the handling of share identifiers for both legacy and base modes.

8.4.1 ASCE activation (legacy mode)

In the legacy mode of the AS protocol, ASCE activation and share identifier handling is largely independent of GCC and is performed using AS-specific protocol elements.

An ASCE joins the AS non-standard base session (see clause 7) using a GCC-Application-Enroll request with the active/inactive flag set to active and with limited capabilities. This does not make the ASCE active (with respect to the AS session states described above) but does ensure that the ASCE is visible to other ASCEs within the conference.

An ASCE may remain enrolled in the non-standard base session over a period, while moving from inactive to active state and back again a number of times. This distinction, between enrolment and activation, enables an ASCE to defer the commitment of application sharing-specific resources to those points within the conference where application sharing is actually taking place.

An ASCE that wishes to become active shall send either a DemandActivePDU or a RequestActivePDU to all ASCEs within the conference in the manner indicated in Table 6-3. The content of the DemandActivePDU is shown in Table 8-26 and the content of the RequestActivePDU is shown in Table 8-27.

It is recommended that an ASCE should initially send a RequestActivePDU to all ASCEs within the conference to determine if there are other active ASCEs in the AS session. If no active ASCEs respond to the RequestActivePDU, then, at some later point, when the ASCE wishes to start sharing, it should send a DemandActivePDU to all ASCEs within the conference to initiate activation.

Sending a DemandActivePDU requires that the sending ASCE shall have generated a unique share identifier, whereas RequestActivePDU does not. Sending DemandActivePDU may force a change in the current agreed share identifier. See clause 8.4.2 for further information on share identifiers.

On receipt of a DemandActivePDU, subject to successful capabilities negotiation (see clause 8.2), an ASCE may send ConfirmActivePDUs (see below) to all ASCEs within the conference. An inactive ASCE is not required to respond to a DemandActivePDU – it may choose to remain inactive. If an inactive ASCE responds to a DemandActivePDU, then it enters the active state.

If an ASCE is already active, then, on receipt of a RequestActivePDU and subject to successful capabilities negotiation (see clause 8.2), it shall send ConfirmActivePDUs (see below) to all ASCEs within the conference.

For both cases, the ConfirmActivePDU shall be sent to all ASCEs within the conference on all priority channels – high, medium and low – in the manner indicated in Table 6-3. The content of the ConfirmActivePDU is shown in Table 8-28. This ensures that any data on any channel is received after the ConfirmActivePDU. The first ConfirmActivePDU received on any channel causes the receiving pending inactive ASCE to enter the active state. Subsequent ConfirmActivePDUs shall be discarded.

Figure 3 illustrates the use of RequestActivePDU, DemandActivePDU and ConfirmActivePDU in the activation of three ASCEs, where initially no ASCEs are active and ASCE 1 uses RequestActivePDU to determine the activation state of other ASCEs within the conference. ASCE 1 then starts sharing and uses DemandActivePDU to initiate the activation of the other ASCEs within the conference.

See Figure 5 for a phase diagram summary of allowable transitions between ASCE states.

Figure 3

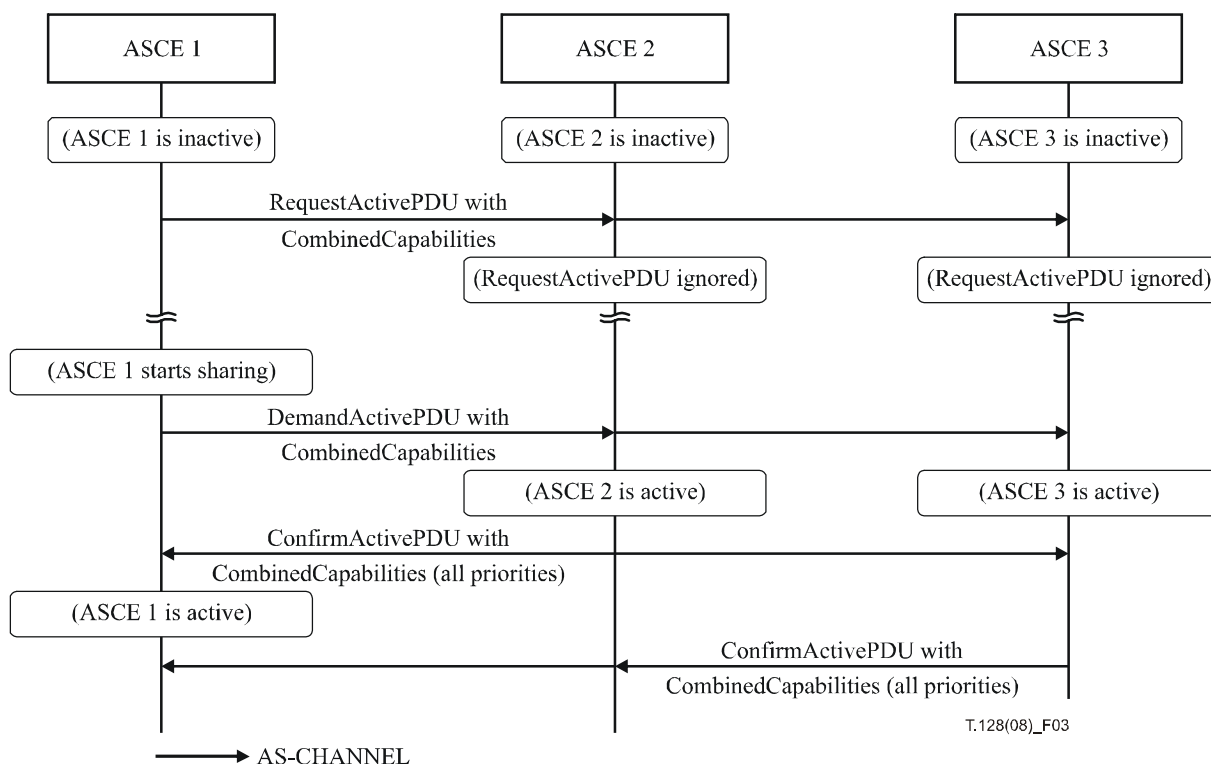


Figure 3 – Successfully activating three ASCEs (legacy mode)

Table 8-26 – DemandActivePDU

Parameter	Description
ShareControl header	The ShareControl header is described in clause 8.3.
shareID	This parameter is the proposed share identifier for use within the AS session. See clause 8.4.2 for a description of this parameter.
sourceDescriptor	This parameter is a null-terminated T.50 text string identifying this ASCE suitable for display to an end-user. This Recommendation does not place any interpretation upon the string contents.
combinedCapabilities	This parameter is a list of this ASCE's combined capabilities, which contains one copy of each of the legacy mode capability sets in any order. See clause 8.2 for further information on capabilities.

Table 8-27 – RequestActivePDU

Parameter	Description
ShareControl Header	The ShareControl header is described in clause 8.3.
sourceDescriptor	This parameter is a null-terminated T.50 text string identifying this ASCE suitable for display to an end-user. This Recommendation does not place any interpretation upon the string contents.
combinedCapabilities	This parameter is a list of this ASCE's combined capabilities, which contains one copy of each of the legacy mode capability sets in any order. See clause 8.2 for further information on capabilities.

Table 8-28 – ConfirmActivePDU

Parameter	Description
ShareControl header	The ShareControl header is described in clause 8.3.
shareID	This parameter is the current share identifier in use within the AS session. See clause 8.4.2 for a description of this parameter.
originatorID	This parameter is the MCS user ID of the ASCE that issued the DemandActivePDU or RequestActivePDU to which this is a response. This enables an ASCE receiving this ConfirmActivePDU to correlate it with a previous DemandActivePDU or RequestActivePDU.
sourceDescriptor	This parameter is a null-terminated T.50 text string identifying this ASCE suitable for display to an end-user. This Recommendation does not place any interpretation upon the string contents.
combinedCapabilities	This parameter is a list of this ASCE's combined capabilities, which contains one copy of each of the legacy mode capability sets in any order. See clause 8.2 for further information on capabilities.

To deactivate another active ASCE, an ASCE shall send a DeactivateOtherPDU to the ASCE that is to be deactivated in the manner indicated in Table 6-3. The content of the DeactivateOtherPDU is shown in Table 8-29. On receipt of a DeactivateOtherPDU, an ASCE shall become inactive. It is recommended that an ASCE should only issue DeactivateOtherPDU where it has received a DemandActivePDU, RequestActivePDU or ConfirmActivePDU with capabilities that would seriously affect the successful progress of application sharing within the conference. This situation should not occur where other ASCEs provide capabilities that conform to the legacy mode of the AS protocol.

An ASCE may deactivate all active ASCEs by sending a DeactivateAllPDU to all ASCEs within the conference in the manner indicated in Table 6-3. The content of the DeactivateAllPDU is shown in Table 8-31. On receipt of a DeactivateAllPDU, an ASCE shall become inactive. It is recommended that an ASCE should only issue DeactivateAllPDU in exceptional circumstances.

When an ASCE determines that there are no other active ASCEs within the AS session, it shall become inactive.

When an ASCE becomes inactive, it shall send a DeactivateSelfPDU to all ASCEs within the conference in the manner indicated in Table 6-3. The content of the DeactivateSelfPDU is shown in Table 8-30.

Figure 4 illustrates the use of DeactivateOtherPDU and DeactivateSelfPDU where ASCE 1 deactivates ASCE 3. On receipt of the DeactivateOtherPDU, ASCE 3 sends DeactivateSelfPDU to ASCEs 1 and 2, whereby both receive notification that ASCE 3 has become inactive.

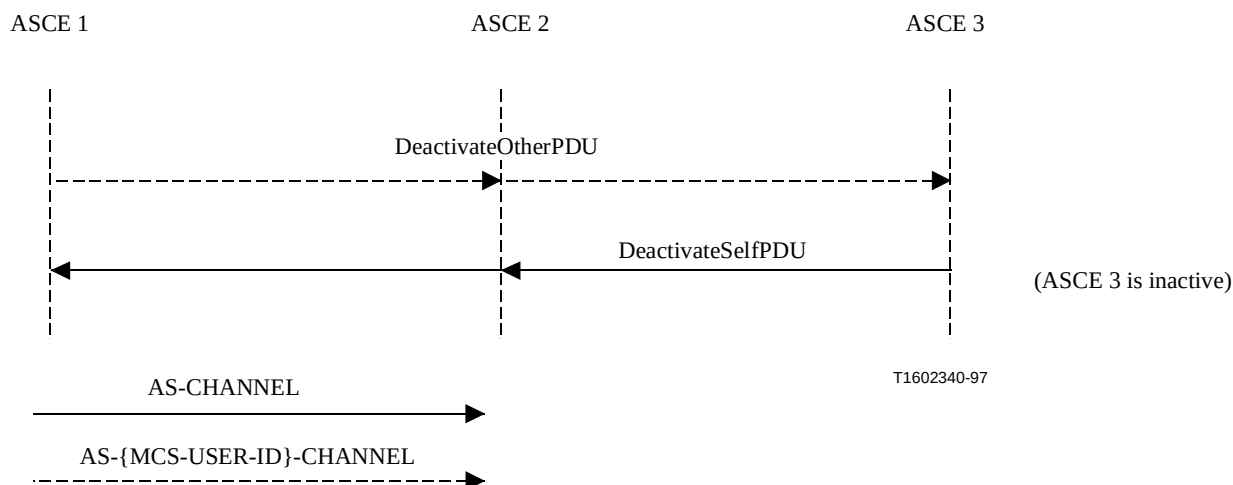


Figure 4 – Deactivating another ASCE

Table 8-29 – DeactivateOtherPDU

Parameter	Description
ShareControl header	The ShareControl header is described in clause 8.3.
shareID	This parameter is the current share identifier in use within the AS session. See clause 8.4.2 for a description of this parameter.
deactivateID	This parameter is the MCS user ID of the ASCE to be deactivated.
sourceDescriptor	This parameter is a null-terminated T.50 text string identifying this ASCE suitable for display to an end-user. This Recommendation does not place any interpretation upon the string contents.

Table 8-30 – DeactivateSelfPDU

Parameter	Description
ShareControl header	The ShareControl header is described in clause 8.3.
shareID	This parameter is the current share identifier in use within the AS session. See clause 8.4.2 for a description of this parameter.

Table 8-31 – DeactivateAllPDU

Parameter	Description
ShareControl header	The ShareControl header is described in clause 8.3.
shareID	This parameter is the current share identifier in use within the AS session. See clause 8.4.2 for a description of this parameter.
sourceDescriptor	This parameter is a null-terminated T.50 text string identifying this ASCE suitable for display to an end-user. This Recommendation does not place any interpretation upon the string contents.

8.4.2 Share identifiers (legacy mode)

A share identifier is a unique 32-bit handle present in (almost) all ASPDUs, which ASCEs use to detect and ignore "late" data generated with respect to previous activations and capabilities within the AS session. In the legacy mode of the AS protocol, share identifiers are constructed locally by