INTERNATIONAL STANDARD

ISO/IEC 9636-2

First edition
1991-12-15

Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification —

Part 2: Control

Technologies de l'information — Infographie — Interfaces pour l'infographie — Spécifications fonctionnelles —

Partie 2: Contrôle

Reference number
ISO/IEC 9636-2:1991(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9636-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology.*

ISO/IEC 9636 consists of the following parts, under the general title *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification*:

— *Part 1: Overview, profiles, and conformance*
— *Part 2: Control*
— *Part 3: Output*
— *Part 4: Segments*
— *Part 5: Input and echoing*
— *Part 6: Raster*

Annexes A and B form an integral part of this part of ISO/IEC 9636. Annex C is for information only.

Introduction

This part of ISO/IEC 9636 describes the functions of the Computer Graphics Interface concerned with Virtual Device management, coordinate space control, and error control.

The functionality incorporated in this part of ISO/IEC 9636 is concerned with the management of the graphics image and the interrelationship of the graphical and non-graphical parts of the interface.

The functionality described in this part of ISO/IEC 9636 pertains to all classes of CGI Virtual Device (i.e. INPUT, OUTPUT, or OUTIN).

This page intentionally left blank

Information technology – Computer graphics – Interfacing techniques for dialogues with graphical devices (CGI) – Functional specification –

Part 2:
Control

1 Scope

This part of ISO/IEC 9636 establishes those functions of the Computer Graphics Interface concerned with Virtual Device management, coordinate space control, and error control.

The functionality incorporated in this part of ISO/IEC 9636 is concerned with the management of the graphics image and the interrelationship of the graphical and non-graphical parts of the interface.

This part of ISO/IEC 9636 is part 2 of ISO/IEC 9636, and should be read in conjunction with ISO/IEC 9636-1 and other parts. The relationship of this part of ISO/IEC 9636 to the other parts of ISO/IEC 9636 is described in ISO/IEC 9636-1 and in clause 4.

The functionality described in this part of ISO/IEC 9636 pertains to all classes of CGI Virtual Device (i.e. INPUT, OUTPUT, and OUTIN).

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9636. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9636 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 2382-13 : 1984 *Data processing – Vocabulary – Part 13: Computer graphics.*

ISO/IEC 9636-1 : 1991 *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 1 : Overview, profiles, and conformance.*

ISO/IEC 9636-3 : 1991 *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 3: Output.*

ISO/IEC 9636-4 : 1991 *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 4: Segments.*

ISO/IEC 9636-5 : 1991 *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 5: Input and echoing.*

ISO/IEC 9636-6 : 1991 *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 6: Raster.*

ISO/IEC 9637-1 : -[1] *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Data stream binding — Part 1: Character encoding.*

ISO/IEC 9637-2 : -[1] *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Data stream binding — Part 2: Binary encoding.*

ISO/IEC TR 9973 : 1988 *Information processing — Procedures for registration of graphical items.*

1) To be published.

3 Concepts

3.1 Introduction

This part of ISO/IEC 9636 defines those functions of the Computer Graphics Interface concerned with Virtual Device management, coordinate space control, and error control. The functionality incorporated in this part of ISO/IEC 9636 is concerned with the management of the graphics image and the interrelationship of the graphical and non-graphical parts of the interface. This functionality is divided into the following areas:

- *Virtual Device management functions*, which allows the CGI client to initiate and terminate sessions of dialogue and to manage the graphics image on the Virtual Device.
- *Coordinate space control functions*, for the establishment of coordinate information, placement of the picture on the drawing surface, and for the management of drawing surface clipping.
- *Error control functions*, which involves the detection of errors both at and subsequent to the transmission of parameters through the CGI.
- *Miscellaneous control functions*, for the establishment of data stream numeric precisions, the accessing of implementation specific functionality, and the accessing of CGI external functions.
- *Control inquiry functions*, which provide access to the description tables and state lists concerned with function and Profile support, device description, and CGI control information.

3.2 Virtual Device management

3.2.1 Device control

The CGI Virtual Device is initialized by the function INITIALIZE. The function TERMINATE sets the CGI Virtual Device in a state in which further CGI commands, other than INITIALIZE, will be ignored. It is not required that any other action take place. There are no state restrictions on the use of INITIALIZE and TERMINATE, i.e. INITIALIZE and TERMINATE may be used at any time.

The Virtual Device is required to perform in conformance with ISO/IEC 9636 from when the INITIALIZE function is invoked until the first TERMINATE.

3.2.2 Drawing surface

Graphical output through the CGI is to a conceptual drawing surface. Physical implementations of a drawing surface may vary, but the functional capability offered by the CGI to control the drawing surface shall be the same for any implementation.

Drawing surfaces are classified as being either hard-copy or soft-copy, on the basis of the medium that implements the display surface. A drawing surface that is hard-copy is implemented by means of a medium that has to be replaced for each new image. A soft-copy drawing surface is implemented by means of a medium that may be cleared for each new image.

Examples of hard-copy drawing surfaces are found in plotter media and film for projection displays. Examples of soft-copy drawing surfaces are found in storage cathode ray tubes, cursively or raster refreshed cathode ray tubes, and in liquid crystal cells.

The PREPARE DRAWING SURFACE function is used to ensure that the Virtual Device is ready to accept graphics at the start of a page or frame.

The END PAGE function ensures that all output is visible and, for a hard-copy device, advances the medium if it has been marked upon. This permits the CGI client to protect the page from being overwritten by the next client to use the device, which is of particular value in a shared peripheral environment.

In some environments, the display surface is subject to spontaneous change in size or shape, for example, in window-managed environments. Whether or not such spontaneous change in display surface size or shape can occur in a given implementation is indicated by an entry in the Output Device Description Table. If such spontaneous change in size or shape information can occur in a given implementation, the device coordinate and size information in the Output Device Description Table may be modified to reflect the change. There is no standardized mechanism to inform the client of a spontaneous change of size or shape. The client can occasionally inquire the Output Device Description Table to discover such a change.

3.2.3 Deferral mode

The CGI permits an implementation to buffer the actions requested by the client in order to provide for efficient use of the resources of physical devices. During this buffering period, the state of the drawing surface may be undefined.

The CGI client has control over this buffering by means of the Deferral Mode entry in the Control State List. The Deferral Mode may have one of three values:

ASTI: requires only that the Virtual Device complete the display of an image "At Some TIme", that is, at its own convenience;

BNI: requires that the Virtual Device complete the display of an image "Before the Next Interaction", that is, before the next interaction with a Logical Input Device gets underway; If an interaction is already underway (i.e. some LID is initialized for events) then BNI is equivalent to ASAP;

ASAP: requires that the Virtual Device complete the display of an image "As Soon As Possible".

Note that none of these values requires an implementation to delay the display of an image. On the other hand, for hard-copy devices, the CGI does not require a page to be printed per function.

Explicit control of deferral is provided by the EXECUTE DEFERRED ACTIONS function which ensures that any pending actions are completed (such as rendering any buffered output so that the operator can see it). The CGI requires that any soliciting function immediately following EXECUTE DEFERRED ACTIONS will not return data until all pending actions are performed and the drawing surface is up to date.

NOTE – Some implementations, such as buffered one-way output devices, may be unable to support Deferral Mode ASAP.

3.2.4 Serial synchronous interface

The CGI is a serial synchronous interface. There are no asynchronous signals over the interface to report events (whether from input interactions or from environmental changes) or the occurrence of errors. The CGI is therefore able to guarantee synchronization of its soliciting functions, including DEQUEUE ERROR REPORTS, with preceding function executions. Invocation of DEQUEUE ERROR REPORTS will return all errors detected as a result of the execution of the preceding functions provided the error queue has not overflowed.

This synchronous interface does not preclude implementations that have many parallel processes within them. Deferral allows for this potential parallelism within the implementation and the function EXECUTE DEFERRED ACTIONS provides a client with some degree of control of this parallelism.

3.3 Coordinate space concepts

3.3.1 The Virtual Device coordinate system

Coordinate data across the CGI is specified in Virtual Device Coordinates (VDCs), except where a direct reference is made to the drawing or display surface. VDC space is an abstract space described in more detail below. The subset of VDC space specified by the finite VDC extent is mapped to a portion of the physical device drawing surface specified by the device viewport.

There are two ways for a CGI client to ensure isotropic mapping from VDC space to the display surface: by asking the CGI to enforce it, or by using a VDC extent whose aspect ratio matches the visual aspect ratio of the selected device viewport. Entries in the Output Device Description Table provide the information that enables the client to ensure isotropy without resorting to implicit CGI mechanisms.

Furthermore, the CGI allows viewport specifications to cause the entire image to be mirrored relative to the normal orientation, in either axes. The Device Viewport Mirroring entry in the Control Description Table provides information on the support of this mirroring capability.

3.3.2 Device coordinates

The drawing surface and display surface are addressed by means of a Cartesian coordinate system. The Display Surface Bottom-Left Corner and Display Surface Upper-Right corner entries in the Output Device Description Table specify this physical device coordinate system. Although the graphic object pipeline model recognizes an abstract DC space with real coordinates, the only form in which device coordinates are passed across the CGI is as integers. If the implementation uses raster techniques, then the units of DCs correspond to single pixel displacements.

3.3.3 Device viewport

The device viewport specifies the region of the device drawing surface onto which the VDC extent is to be mapped.

The position of the device viewport is specified in one of three coordinate systems determined by the Device Viewport Specification Mode entry in the Control State List:

- by fraction [0.0 .. 1.0] × [0.0 .. 1.0] of the available display surface, which allows reasonable placement and relative sizing of the viewport, even without inquiry;
- in millimetres times a metric scale factor, which allows absolute sizing of images without inquiry, but which requires inquiry or prior knowledge of the device for assurance that the image will fit on the display surface;
- in physical device coordinates, which requires either inquiry or prior knowledge of the device.

The device viewport is specified in terms of two points (on the display surface) at diagonally opposite corners of the rectangle. The order in which the points are specified is significant.

The VDC-to-Device Mapping entry in the Control State List may force isotropic mapping. If the current VDC extent, device viewport, and device viewport mapping would not lead to an isotropic mapping, the VDC extent is mapped onto a subset of the specified device viewport. This subset is defined by shrinking either the vertical or horizontal dimension of the current device viewport, as needed, to reach the required aspect ratio. This smaller effective device viewport is used to define the coordinate mapping from VDC to the device's coordinates. The placement of the effective viewport rectangle within the original one can be specified. This placement can be one of LEFT, RIGHT, or CENTRED when the shrinking is horizontal, and TOP, BOTTOM or CENTRED when the shrinking is vertical. These meanings are relative to the display surface. (See figure 1.)

3.3.4 VDC space and range

Graphics output functions are used to define virtual images. The coordinate data given as parameters to these functions (that is, points in the virtual image) are specified as absolute two-dimensional Virtual Device Coordinates (VDCs). VDC space is a two-dimensional Cartesian coordinate space of infinite precision and infinite extent. Only a subset of VDC space, the VDC range, is realizable by the CGI client. The VDC range comprises all coordinates representable in the format specified by the declared VDC type and limited by any applicable precision; thus, the VDC range is not directly set by the client. The VDC range is a finite discrete subset of VDC space (i.e. it does not provide a continuous range of values).

VDC space can be addressed with either integer or real coordinate data, determined by the VDC Type entry in the Control State List and controlled by the VDC TYPE function. The granularity and realizable extent of the VDC range is affected by either the VDC INTEGER PRECISION REQUIREMENT function or the VDC REAL PRECISION REQUIREMENTS function, depending on the VDC Type. The Control Description Table indicates which of integer and real types are supported for VDCs. Refer to 3.5.1 for further information on precision control.

3.3.5 VDC extent

The VDC extent is the portion of VDC space that is to be mapped onto the effective device viewport on the drawing surface of the Virtual Device. The extent is set by specifying the addresses (in VDC space) of two opposite corners of a rectangular region. Values outside the VDC extent are permitted in CGI functions.

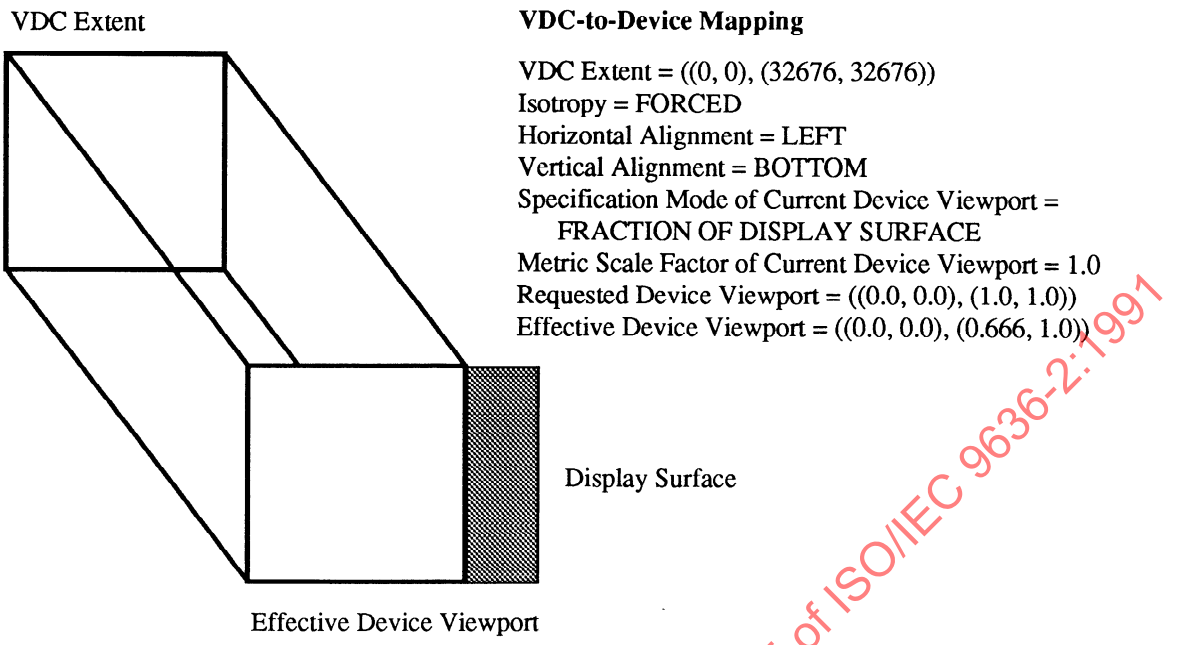


Figure 1 – An example VDC-to-Device Mapping.

The values of the coordinates for either dimension may be either increasing or decreasing from the first to the second corner. In this way, the sense of the coordinate system of VDC space relative to the drawing surface is established (see figure 2).

The transformation which maps VDC points to the drawing surface is called the VDC-to-Device Mapping. The VDC-to-Device Mapping maps the first point specifying the VDC extent onto the corner of the effective device viewport corresponding to the first point specifying the device viewport, and similarly for the second point. The mapping is linear in each dimension, but is not necessarily isotropic (e.g. a circle in VDC may not appear round to the viewer). If the values of the device viewport mapping entries do not force isotropy, an isotropic transformation can still be assured if the numerical aspect ratio of VDC extent matches the physical (not necessarily numerical) aspect ratio of the device viewport.

Angular directions are defined as follows: positive 90-degrees is defined to be the right angle from the positive x-axis to the positive y-axis (see figure 2).

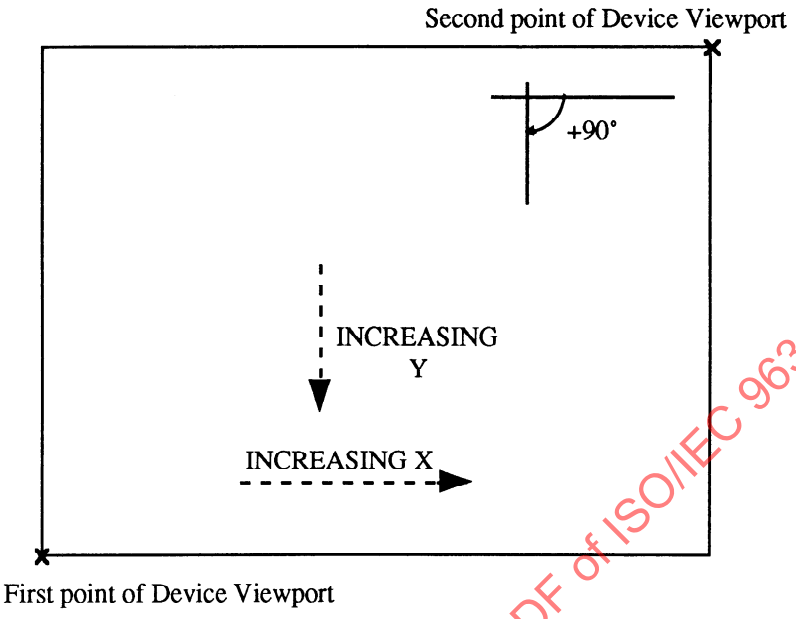
Whether changes to the VDC-to-Device Mapping take place immediately, can be simulated, or lead to an implicit regeneration, is determined by the Dynamic Modification Accepted For VDC-to-Device Mapping entry in the Output Device Description Table.

The terminology used in the description of primitives and attributes refers to increasing coordinates from the first to the second corner relative to the device viewport. If a coordinate system is chosen with decreasing coordinates from the first to the second corner in one of x or y, the rendered objects shall be mirrored. If decreasing in both x and y, the rendered objects shall be rotated by an angle of 180°.

3.3.6 VDC tailoring

The ability to specify the VDC range and the VDC extent provides the flexibility to configure the Virtual Device coordinate space to match various needs. It may be configured as an abstract, normalized coordinate range for maximum device independence. It may also be configured to match the address range and resolution of some target device (e.g. in order to avoid aliasing problems or increase performance).

If the Virtual Device coordinate space is configured to match the address range and resolution of a raster device, it may be necessary to know whether or not the pixels lie on or between the coordinates. Where pixels lie relative to the coordinates is indicated by an entry in the Output Device Description Table. The preferred behaviour is that coordinates lie between pixels.



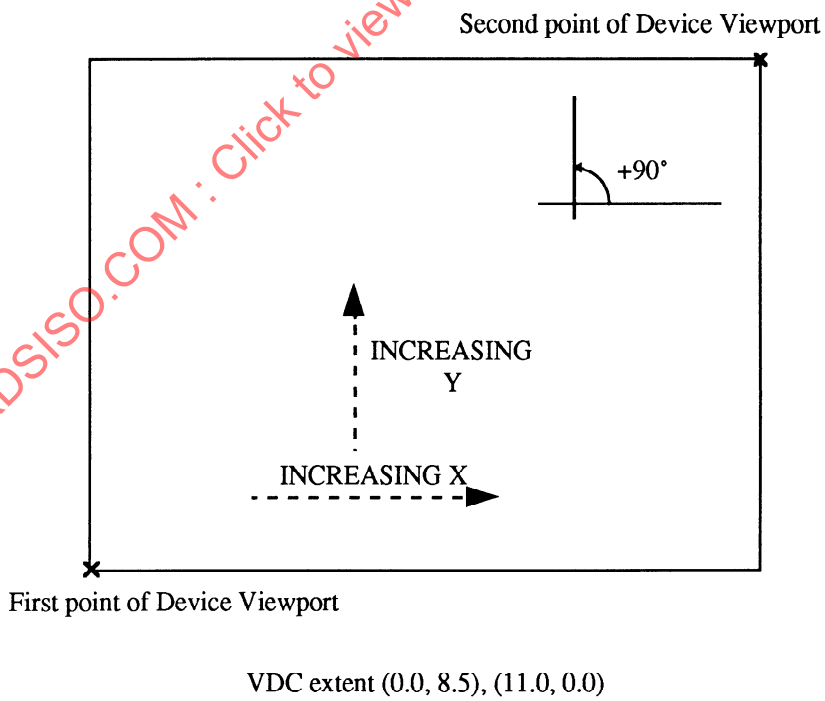


Figure 2 – VDC extent establishes the reference directions relative to the drawing surface.

3.3.7 Drawing surface clipping

Drawing surface clipping conceptually occurs in abstract DC space before the final physical rendering step. The Drawing Surface Clip Indicator and Drawing Surface Clip Rectangle entries in the Control State List provide control over drawing surface clipping. They are set by the functions DRAWING SURFACE CLIP INDICATOR and DRAWING SURFACE CLIP RECTANGLE.

- If the Drawing Surface Clip Indicator is DSCRECT, the effective drawing surface clip region is the intersection of the contents of the Drawing Surface Clip Rectangle entry and the limits of the device's drawing surface.
- If the Drawing Surface Clip Indicator is VIEWPORT, the effective drawing surface clip region is the intersection of the effective device viewport and the limits of the device's drawing surface.
- If the Drawing Surface Clip Indicator is OFF no further drawing surface clipping is performed. Implementations of CGI for physical devices which could suffer damage from an image whose extent is not controlled may always enforce some sort of clipping, regardless of the state of the Drawing Surface Clip Indicator.

Drawing surface clipping applies to all types of output primitives. It operates (conceptually) by intersecting the effective drawing surface clip region with the graphic objects after complete rendering in abstract DC space.

Note that with DSCRECT, it is possible to draw outside the device viewport if the drawing surface is larger than the device viewport, the Drawing Surface Clip Rectangle extends outside the device viewport, and the clipping associated with graphic objects does not restrict them to the device viewport (refer also to ISO/IEC 9636-3, 3.6 and ISO/IEC 9636-6, 3.5.2).

3.4 Error control

The philosophy of the specification of errors by ISO/IEC 9636, the definition of the error classes, and the actions performed by an implementation when an error occurs, are described in ISO/IEC 9636-1, 5.2.8.

This part of ISO/IEC 9636 defines the functions that control detection and reporting of errors, and that retrieve error reports from the error queue.

The function ERROR HANDLING CONTROL enables the client to selectively turn off or on error detection and reporting for each class of error. This treatment may override the defaults specified with the classes in ISO/IEC 9636-1, 5.2.8.

The function DEQUEUE ERROR REPORTS enables a client to retrieve error reports from the error queue in the CGI Virtual Device.

3.5 Miscellaneous control

3.5.1 Numerical precision requirement specifications

The functions below establish client requirements for precision in representing and communicating information for data types subject to variable precision in CGI data stream interfaces.

Table 1 – Precision functions

Data Type	Precision Function
VDC (Integer)	VDC INTEGER PRECISION REQUIREMENT
VDC (Real)	VDC REAL PRECISION REQUIREMENTS
CD	COLOUR PRECISION REQUIREMENT
CI	COLOUR INDEX PRECISION REQUIREMENT
CSN	CLIENT SPECIFIED NAME PRECISION REQUIREMENT
I	INTEGER PRECISION REQUIREMENT
IX	INDEX PRECISION REQUIREMENT
R	REAL PRECISION REQUIREMENTS

Additional encoding-specific precision functions shall be specified in encodings in order to provide the mechanism for controlling the format of the data passed across the data stream interface of a CGI implementation. All instances of the use of data type IF (Fixed precision Integer) have their own fixed precisions (not necessarily the same) and are not susceptible to precision control.

The above functions for specifying client requirements for data precision have no effect on the format of data passed across procedural interface for any binding of the CGI.

In general, there may be multiple Generator/Interpreter pairs in the path from client to target. (See ISO/IEC 9636-1, 4.2.) The client's precision requirements are passed to all these agents in the system and maintained as local state information in each. Based on the precision requirements information, each Generator can make an independent decision about what particular encoding-specific precisions must be used to satisfy the client's requirements. Note that a Generator is not required to match the required precision so long as it can employ a precision sufficient to satisfy a client's requirement. (For example, suppose the physical links between a Generator/Interpreter pair is a 32-bit data bus. It may be more efficient to use 32-bit integer precision at the local level, even though the client has indicated that 16-bit precision is sufficient for its requirements.)

In each encoding, elements exist to specify precise encoding details relative to data passed over the communications link between any one Generator/Interpreter pair. When adjustment of these parameters is required in response to a client's precision requirements request, appropriate encoding-specific elements shall be produced by each Generator to inform the single immediately downstream Interpreter about how the Generator will encode the relevant parameters on its downstream data stream, and how the Interpreter is to encode the relevant parameters on its upstream data stream. The encoding-specific elements are purely an encoding mechanism employed only on the communication link between one Generator/Interpreter pair, while the precision requirements functions defined in clause 5 result in providing the same information to all Generators and Intepreters between the client and target.

3.5.2 Escape concepts

ESCAPE functions allow non-standard device-dependent or system-dependent data to be passed across the CGI. ESCAPEs may be used at the discretion of the client, but the specification of direct effects and side effects of the use of non-standard functions are beyond the scope of ISO/IEC 9636. ISO/IEC 9636 makes the distinction between ESCAPE and GENERALIZED DRAWING PRIMITIVE based on the following definition. Any non-standard function which generates a graphic object or a part of a graphic object shall be called GENERALIZED DRAWING PRIMITIVE; any other non-standard functions shall be called ESCAPE. No other constraint on the functional intent or content of data passed by the ESCAPE mechanism will be imposed. For example, ISO/IEC 9636 does not preclude the data record of an ESCAPE from containing a transformable point list.

There are two Escape Functions defined in ISO/IEC 9636:

- ESCAPE provides communication of non-standard device-dependent or system-dependent data from the client to the Virtual Device;
- GET ESCAPE provides for the implementation of non-standard device-dependent or system-dependent soliciting functions, such as inquiry or retrieval, by providing a return parameter (data record).

3.5.3 External functions

External functions communicate information not directly related to the generation of a graphic image.

The MESSAGE function specifies a string of characters used to communicate information to an operator. This function can be used to provide special device-dependent information necessary to manage the device. Control over the position and appearance of the character string is not provided.

3.6 Inquiry concepts

Inquiry functions, as defined in clause 6, provide the client with the means to access the information in the Device Identification, Output Device, Function and Profile Support, and Control Description Tables, and the Control State List. These description tables and state list provide information about the capabilities and current state of the CGI Virtual Device.

Concepts **Inquiry concepts**

Details about the relationship between a description table or state list and the corresponding inquiry function(s) are described in ISO/IEC 9636-1, 5.2.7.

4 Interactions with other parts of ISO/IEC 9636

4.1 Interactions with more than one part of ISO/IEC 9636

This part of ISO/IEC 9636 has significant and far-reaching interactions with each of the other functional parts of ISO/IEC 9636. In general, the functions of this part of ISO/IEC 9636 define the environment in which the functions of all of the other parts execute. None of the state restrictions defined in the other parts of ISO/IEC 9636 apply to the functions of this part of ISO/IEC 9636.

The functions of this part are divided into four categories: Virtual Device management, Coordinate space control, Error, and Miscellaneous.

4.1.1 Virtual Device management

Of the functions in this category, INITIALIZE and TERMINATE have explicit effects on the state defined in all other functional parts.

PREPARE DRAWING SURFACE, END PAGE, EXECUTE DEFERRED ACTIONS, and DEFERRAL MODE control the general display facilities, and as such are limited to only the parts concerned with direct graphical output, ISO/IEC 9636-3 and ISO/IEC 9636-6. In devices implementing the functions of ISO/IEC 9636-6 the PREPARE DRAWING SURFACE, END PAGE, EXECUTE DEFERRED ACTIONS, and DEFERRAL MODE functions act upon the currently selected drawing bitmap.

4.1.2 Coordinate space control

The coordinate space control functions provide control of the coordinate space mapping, VDC precision, and drawing surface clipping. They apply most directly to graphical output, ISO/IEC 9636-3 and ISO/IEC 9636-6, but also serve as the model of similar input related controls. These additional controls are elaborated in ISO/IEC 9636-5. The coordinate spaces are used in support of the segment functions in ISO/IEC 9636-4.

Mirroring as a result of the VDC-to-Device Mapping does not apply to raster operation functions.

For a CGI implementation which supports the functionality of ISO/IEC 9636-6, the VDC-to-Device Mapping entries of the current drawing bitmap determine the VDC-to-Device Mapping. The corresponding values of the Control State List in such implementations are actually the corresponding entries in the current drawing bitmap's Bitmap State List. Inquiry of the Control State List shall therefore always return values for VDC-to-Device Mapping and Drawing Surface Clip entries from the current drawing bitmap's Bitmap State List. See ISO/IEC 9636-6, 4.1.

NOTE – These functions have side effects if used when a compound graphic object is under construction (see ISO/IEC 9636-3, 4.2.2).

4.1.3 Error

The error control functions defined in this part of ISO/IEC 9636 apply equally to errors arising from functions defined in all parts of ISO/IEC 9636.

4.1.4 Miscellaneous

These functions control integer, real, index, colour, colour index, and client specified name precision. Control of data precision, defined in this part of ISO/IEC 9636, applies to all functions that carry data subject to such precision change across the CGI.

ESCAPE and GET ESCAPE, defined in this part of ISO/IEC 9636, have no defined bounds in terms of functional impact (except with respect to the creation of graphic objects, see 3.5.2). ESCAPEs can be defined which affect any and all elements of the Virtual Device.

5 Abstract specification of functions

5.1 Introduction

5.1.1 Control functions

This clause defines the abstract functional specification of the Control functions of the CGI.

The format used throughout this clause to define the CGI function set is designed to separate function behaviour from implementation. Each function is named, the parameters are described, data types are listed, and a description of implicit relationships is added to clarify how the function fits into the system.

The descriptions of individual functions are grouped on the basis of functional association, as follows:

- Virtual Device management functions, which allows the CGI's client and the Virtual Device to initiate and terminate sessions of dialogue and to manage the graphics image on the Virtual Device.
- Coordinate space control functions, for the establishment of coordinate information, placement of the picture on the drawing surface, and for the management of drawing surface clipping.
- Error functions, which involve the detection of errors both at and below the CGI.
- Miscellaneous control functions, for establishing data stream numeric precisions, accessing implementation specific functionality, and accessing CGI external functions.
- Control inquiry functions, which provide access to the description tables and state lists concerned with function and Profile support, device description and CGI control information.

5.1.2 Validity of returned information

For all of the functions specified in this clause which solicit a response from the Virtual Device, a response validity flag is returned as INVALID if an error was detected in executing the function. In such cases, other output parameters are undefined and no meaning should be applied to any other of these parameter values.

5.1.3 Data types employed

The abstract specifications of functions detail the functions in terms of input and output parameters. The data type of each parameter is selected from a standard set and is identified in the functional specification by a standard abbreviation.

The data types and the abbreviations used in this part of ISO/IEC 9636 are taken from the complete list of data types given in ISO/IEC 9636-1, 5.2.10.

5.2 Virtual Device management functions

5.2.1 INITIALIZE

Parameters:

None

Effect:

This function resets all state list entries to their default values described in ISO/IEC 9636-2, ISO/IEC 9636-3, ISO/IEC 9636-4, ISO/IEC 9636-5, and ISO/IEC 9636-6. All dynamically created entities are deleted. These include bundles, attribute name sets, LID state lists, echo output entities, bitmaps, the error queue, the event queue, and segments. This function is only defined to affect states specified in ISO/IEC 9636-2, ISO/IEC 9636-3, ISO/IEC 9636-4,

ISO/IEC 9636-5, and ISO/IEC 9636-6. Effects on states outside the CGI (such as modes of operation) are implementation-dependent. An implementation of the CGI shall always be able to recognize the INITIALIZE function; thus, the INITIALIZE function is not explicitly subject to any state restrictions.

NOTE – It may be necessary that an agent outside the CGI employ functionality not specified in ISO/IEC 9636 to ensure that the implementation is able to recognize the INITIALIZE function. For example, the network connection to an interpreter implementation may need to be reset before the CGI interpreter is able to respond to further commands.

5.2.2 TERMINATE

Parameters:

None

Effect:

The function signals the end of the current CGI session. The Virtual Device will ignore any further CGI functions until it receives an INITIALIZE function.

ISO/IEC 9636 specifies no other effects of this function.

5.2.3 EXECUTE DEFERRED ACTIONS

Parameters:

None

Effect:

This function ensures that all pending operations buffered within the Virtual Device are completely processed. This means that a subsequent soliciting function will not be processed until all actions pending at the time of the EXECUTE DEFERRED ACTIONS have been performed.

A hard-copy device is not required to print the image if it would adversely affect the final output (see 5.2.6).

5.2.4 DEFERRAL MODE

Parameters:

In	deferral mode	(ASTI, BNI, ASAP)	E

Effect:

This function sets the Deferral Mode entry in the Control State List to the value specified by the parameter. See 3.2.3 on deferral for a description of how deferral modes affect the Virtual Device behaviour.

5.2.5 PREPARE DRAWING SURFACE

Parameters:

In	clear drawing surface	(UNCONDITIONAL, CONDITIONAL)	E

Effect:

This function clears the drawing surface and sets the Device Drawing Surface State to CLEAN. The drawing surface is set to the background colour (see ISO/IEC 9636-3, 3.4 for a description of background colour).

If the *clear drawing surface* parameter is set to CONDITIONAL and the Control State List entry for Device Drawing Surface State is CLEAN, the drawing surface is not cleared. Otherwise, the drawing surface is cleared unconditionally.

5.2.6 END PAGE

Parameters:

None

Effect:

This function first performs an EXECUTE DEFERRED ACTIONS. It does not otherwise affect the display of a soft-copy device. It then advances the medium on a hard-copy device unless it is known that the display medium has not been marked. For hard-copy devices, the complete picture is produced as a result of END PAGE.

5.3 Coordinate space control functions

5.3.1 VDC TYPE

Parameters:

In	VDC type selector	(INTEGER, REAL)	E

Effect:

This function sets the VDC Type entry in the Control State List to the value of the parameter. This entry indicates to the Virtual Device whether subsequent parameters of data type VDC or P will be represented by integer or real values. Changing VDC Type sets all state lists entries specified in VDC to their default values specified in the new type.

NOTE – This function has side effects if it is used when a compound object is under construction (see ISO/IEC 9636-3, 4.2.2).

Errors:

Error identifier: 3:201
Cause: Selected VDC type not supported
Reaction: All functions with VDC parameters are ignored until a supported VDC type is selected, or until INITIALIZE.

5.3.2 VDC INTEGER PRECISION REQUIREMENT

Parameters:

In	log2 magnitude upper bound	(7..31)	I

Effect:

The VDC Integer Precision Requirement entry in the Control State List is set to the value specified. It affects the manner in which Generators and Interpreters encode parameters of data type VDC or P when VDC type is INTEGER.

See 5.5.1 and 3.5.1.

NOTE – The precision requirement for non-coordinate integers is specified separately by the function INTEGER PRECISION REQUIREMENT.

Errors:

Error identifier: 3:202
Cause: Specified VDC precision requirements are not achievable
Reaction: All functions with VDC parameters are ignored until supportable VDC precision requirements are specified, or until INITIALIZE.

5.3.3 VDC REAL PRECISION REQUIREMENTS

Parameters:

In	log2 magnitude upper bound (u)		I
In	log2 non-zero magnitude minimum (l)		I
In	log2 relative precision requirement (p)	(≤ 0)	I
In	log2 minimum magnitude for relative precision (m)		I
In	log2 typical magnitude (t)		I

Effect:

The VDC Real Precision Requirements entries in the Control State List are set to the values specified. These entries affect the manner in which Generators and Interpreters encode parameters of data type VDC or P when VDC type is REAL.

See 5.5.2 and 3.5.1.

NOTE – The precision requirements for non-coordinate reals are specified separately by the function REAL PRECISION REQUIREMENTS.

Errors:

Error identifier: 3:202
Cause: Specified VDC precision requirements are not achievable
Reaction: All functions with VDC parameters are ignored until supportable VDC precision requirements are specified, or until INITIALIZE.

5.3.4 VDC EXTENT

Parameters:

In	first corner (P), second corner (P')	($P_x \neq P'_x, P_y \neq P'_y$)	2P

Effect:

This function sets the VDC Extent entry in the Control State List. The two specified corners define a rectangular region in VDC space which together with the Effective Viewport and device viewport mapping entries defines the VDC-to-Device Mapping.

This region is mapped onto the Effective Viewport such that the *first corner* of VDC Extent is mapped to the "first corner" of the Effective Viewport, and the *second corner* of VDC Extent is mapped to the "second corner" of the Effective Viewport. The mapping is performed according to the specifications of the current device viewport mapping entries in the Control State List. The Effective Viewport may itself be placed on the physical device inverted in either axis, leading to mirroring or a 180° rotation.

See 3.3.3, 3.3.5, and 5.3.1.

NOTE – Changing the VDC Type sets the VDC Extent to the default for the chosen type. The range of coordinates used by CGI functions is not restricted to the VDC Extent. The limit on the range of the VDC Extent is determined by the current integer or real precision for VDCs, depending on VDC Type. This function has side effects if it is used when a compound object is under construction (see ISO/IEC 9636-3, 4.2.2).

5.3.5 DEVICE VIEWPORT

Parameters:

In	first corner (P), second corner (P')	($P_x \neq P'_x, P_y \neq P'_y$)	2VP

Effect:

This function sets the Requested Device Viewport and Effective Viewport entries in the Control State List. The two specified corners define the opposite corners of a rectangular region, the Requested Device Viewport, on the device's display surface.

The Effective Viewport is that area of the Requested Device Viewport onto which the VDC Extent rectangle is mapped. If the current device viewport mapping entries force an isotropic mapping and the aspect ratio of the VDC extent is not

equal to that of the Requested Device Viewport, the Effective Viewport will be smaller than the Requested Device Viewport on one or the other axis, but not both.

If the current device viewport mapping entries do not force an isotropic mapping, the Effective Viewport will be the same as the Requested Device Viewport.

If the Requested Device Viewport exceeds the available drawing surface, the Requested Device Viewport is still used to determine the VDC-to-Device Mapping.

Mirroring or 180° rotation of the image on the Virtual Device may be achieved by specifying the corners in some way other than that the first is below and to the left of the second.

See 3.3.3 and 3.3.5.

NOTE – This function has side effects if it is used when a compound object is under construction (see ISO/IEC 9636-3, 4.2.2).

Errors:

Error identifier: 3:206
Cause: Device viewport mirroring not supported
Reaction: Non-mirrored viewport is used.

5.3.6 DEVICE VIEWPORT SPECIFICATION MODE

Parameters:

In	VC mode	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	E
In	metric scale factor	(> 0)	R

Effect:

This function sets the Device Viewport Specification Mode entry in the Control State List. This entry determines how parameters of data type VC (Viewport Coordinate) and VP (Viewport Point) will be interpreted by the CGI. These parameters may be specified in one of three modes: FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, or PHYSICAL DEVICE COORDINATES.

When the *VC mode* is FRACTION OF DISPLAY SURFACE, the value (0.0, 0.0) corresponds to the lower left corner and the value (1.0, 1.0) corresponds to the upper right corner of the default device viewport. (The default device viewport is the largest axis-aligned rectangular area visible on the display surface.) Numbers outside of the range [0.0..1.0] may be specified (see the DEVICE VIEWPORT function). The *metric scale factor* is ignored.

When the *VC mode* is MILLIMETRES WITH SCALE FACTOR, the *metric scale factor* represents the distance in millimetres on the display surface corresponding to a VC change of 1.0. One VC represents one millimetre multiplied by the *metric scale factor*. The value (0.0, 0.0) corresponds to the lower left corner and the values increase positively to the right and upwards.

When the *VC mode* is PHYSICAL DEVICE COORDINATES, the native units and handedness of the physical device are used. These can be determined from the Output Device Description Table. The *metric scale factor* is ignored.

Errors:

Error identifier: 3:203
Cause: Selected viewport specification mode not supported
Reaction: All functions using VC parameters are ignored until a supported mode is selected, or until INITIALIZE.

5.3.7 DEVICE VIEWPORT MAPPING

Parameters:

In	isotropy flag	(NOT FORCED, FORCED)	E
In	horizontal alignment flag	(LEFT, CENTRE, RIGHT)	E
In	vertical alignment flag	(BOTTOM, CENTRE, TOP)	E

Effect:

This function sets the Isotropy, Horizontal Alignment, and Vertical Alignment entries in the Control State List. These entries determine how the coordinate mapping is derived from the VDC Extent and the Requested Device Viewport. The VDC-to-Device Mapping is redefined. See 5.3.5 for a detailed explanation.

The alignment parameters are only significant if the *isotropy flag* is FORCED. If the CGI forces isotropy, the Effective Viewport may be smaller than the Requested Device Viewport. The alignment parameters determine how it will be positioned within the rectangle defined by the Requested Device Viewport.

See 3.3.3 and 3.3.5.

NOTE – This function has side effects if it is used when a compound object is under construction (see ISO/IEC 9636-3, 4.2.2).

5.3.8 DRAWING SURFACE CLIP RECTANGLE

Parameters:

In	first corner (P), second corner (P')	($P_x \neq P'_x$, $P_y \neq P'_y$)	2VP

Effect:

This function sets the Drawing Surface Clip Rectangle entry in the Control State List. For information on how it affects subsequent rendering of a graphic object, see 3.3.7.

5.3.9 DRAWING SURFACE CLIP INDICATOR

Parameters:

In	clip indicator	(OFF, DSCRECT, VIEWPORT)	E

Effect:

This function sets the Drawing Surface Clip Indicator entry in the Control State List. For information on how it affects subsequent rendering of a graphic object, see 3.3.7.

5.4 Error functions

5.4.1 DEQUEUE ERROR REPORTS

Parameters:

In	number of reports requested	(1..n)	I
Out	response validity	(INVALID, VALID)	E
Out	number of reports remaining in queue	(0..n)	I
Out	list of error reports		nER

Effect:

This function removes the oldest error reports from the error queue and returns them in the *list of error reports* parameter. The number of reports removed and returned is no greater than either the number of queued error reports or the input parameter *number of reports requested.*

The *number of reports remaining in queue* corresponds to the state after removal of the returned error reports.

See ISO/IEC 9636-1, 5.2.8 for details of error reporting.

5.4.2 ERROR HANDLING CONTROL

Parameters:

In	list of error class/handling pairs	((1..9), (ON, REPORTING OFF, DETECTION OFF))	n[IN,E]

Effect:

This function informs the Virtual Device of the level of error reporting and error detection required by the client. The level of error control is given on a error class by error class basis.

If reporting is disabled, no error reports for the indicated class will be entered into the error queue.

If detection is disabled, reporting is also disabled and the Virtual Device may disable its error detection logic.

NOTE – The CGI implementation may elect to keep its error detection logic enabled for its own purposes, such as robustness, or if the implementor feels that no gain (e.g. performance) would result if error detection for that error class were disabled.

5.5 Miscellaneous control functions

5.5.1 INTEGER PRECISION REQUIREMENT

Parameters:

In	log2 magnitude upper bound	(7..31)	I

Effect:

The Integer Precision Requirement entry in the Control State List is set to the value specified. It affects the manner in which Generators and Interpreters encode parameters of data type I (Integer).

The precision requirement is expressed with an integer parameter as a base 2 logarithm of the relevant number size. Specifying a value *n* for *log2 magnitude upper bound*, indicates that the client will not attempt to use any integer whose magnitude exceeds $2^n - 1$. Thus, for a binary encoding, a specification of $n = 15$ corresponds to a (minimum sufficient) 16-bit integer precision.

See 3.5.1.

NOTE – Integer precision requirements specifications have little impact on the behaviour of Generator/Interpreter pairs using character or clear text encodings. The precision requirement for integers used as coordinate data is specified separately by the function VDC INTEGER PRECISION REQUIREMENT.

Errors:

Error identifier: 3:204
Cause: Specified precision requirement not achievable
Reaction: All functions with affected parameters are ignored until a supportable precision requirement is specified, or until INITIALIZE.

5.5.2 REAL PRECISION REQUIREMENTS

Parameters:

In	log2 magnitude upper bound (u)		I
In	log2 non-zero magnitude minimum (l)		I
In	log2 relative precision requirement (p)	(≤ 0)	I
In	log2 minimum magnitude for relative precision (m)		I
In	log2 typical magnitude (t)		I

Effect:

The Real Precision Requirements entries in the Control State List are set to the values specified. These entries affect the manner in which Generators and Interpreters encode parameters of data type R (Real).

In terms of a binary floating point representation for which the base of the exponent is 2, u and l are related to required exponent range and p is related to the number of bits required in the fraction. Values of p are non-positive. The smallest difference that can be represented between a number A and $A+d$ shall not exceed $2^p A$. This relative precision requirement does not have to apply to numbers whose magnitudes are less than 2^m. This latter consideration can justify the use of fixed-point representation. For example, $u = 15$, $l = -16$, $m = 0$, and $p = -16$ would be satisfied by 32-bit fixed-point precision with a 16-bit fraction component.

The parameter *log2 typical magnitude* is intended for use by Generators of a character encoding and may be used to set the default exponent. The parameter *log2 non-zero magnitude minimum* is also relevant in a character encoding to control the number of binary places used to encode data. For large numbers, these precision parameters can justify a decrease in the number of binary places encoded (after the binary point).

See 3.5.1.

NOTE – The precision requirements for reals used as coordinate data are specified separately by the function VDC REAL PRECISION REQUIREMENTS.

Errors:

Error identifier: 3:204
Cause: Specified precision requirement not achievable
Reaction: All functions with affected parameters are ignored until a supportable precision requirement is specified, or until INITIALIZE.

5.5.3 INDEX PRECISION REQUIREMENT

Parameters:

In	log2 magnitude upper bound	I

Effect:

The Index Precision Requirement entry in the Control State List is set to the value specified. It affects the manner in which Generators and Interpreters encode parameters of data type IX (index).

See 3.5.1.

NOTE – The precision requirement for indices used for colour selection is specified separately by the function COLOUR INDEX PRECISION REQUIREMENT.

Errors:

Error identifier: 3:204
Cause: Specified precision requirement not achievable
Reaction: All functions with affected parameters are ignored until a supportable precision requirement is specified, or until INITIALIZE.

5.5.4 COLOUR PRECISION REQUIREMENT

Parameters:

In	log2 upper bound	(1..32)	I

Effect:

The Colour Precision Requirement entry in the Control State List is set to the value specified. It affects the manner in which Generators and Interpreters encode parameters of data type CD (colour direct). It specifies the minimum number of bits required to specify each component of the RGB triple.

See 3.5.1.

Errors:

Error identifier: 3:204
Cause: Specified precision requirement not achievable
Reaction: All functions with affected parameters are ignored until a supportable precision requirement is specified, or until INITIALIZE.

5.5.5 COLOUR INDEX PRECISION REQUIREMENT

Parameters:

In	log2 upper bound	(8..32)	I

Effect:

The Colour Index Precision Requirement entry in the Control State List is set to the value specified. It affects the manner in which Generators and Interpreters encode parameters of data type CI (colour index). It specifies the minimum number of bits required to specify each colour index.

See 3.5.1.

NOTE – The precision requirement for indices other than colour indices is specified separately by the function INDEX PRECISION REQUIREMENT.

Errors:

Error identifier: 3:204
Cause: Specified precision requirement not achievable
Reaction: All functions with affected parameters are ignored until a supportable precision requirement is specified, or until INITIALIZE.

5.5.6 CLIENT SPECIFIED NAME PRECISION REQUIREMENT

Parameters:

In	log2 magnitude upper bound	(7..31)	I

Effect:

The Client Specified Name Precision Requirement entry in the Control State List is set to the value specified. It affects the manner in which Generators and Interpreters encode parameters of data type CSN (client specified name).

See 3.5.1.

Errors:

Error identifier: 3:204
Cause: Specified precision requirement not achievable
Reaction: All functions with affected parameters are ignored until a supportable precision requirement is specified, or until INITIALIZE.

5.5.7 MESSAGE

Parameters:

In	action required flag	(NO ACTION REQUIRED, ACTION REQUIRED)	E
In	message text		S

Effect:

The MESSAGE function specifies a string of characters used to communicate information to a CGI operator through a path separate from normal graphical output. Character set selection for MESSAGE is independent of that for the TEXT primitive and is not specified by ISO/IEC 9636.

If the *action required flag* is ACTION REQUIRED, a subsequent soliciting function will not return until after any such required action has been performed by the operator.

Errors:

Error identifier: 3:205
Cause: Message action detection not supported
Reaction: The next function after MESSAGE will be processed immediately; the action required flag is not interpreted.

5.5.8 ESCAPE

Parameters:

In	escape function identifier	(n ≠ 0)	I
In	data record		D

Effect:

This function allows use of device capabilities not specified by ISO/IEC 9636.

The *escape function identifier* parameter specifies the particular escape function. Non-negative escape function identifiers are reserved for registration, negative escape function identifiers are available for private use.

An ESCAPE can be used to specify non-standard attributes of a primitive.

NOTE – ESCAPE is designed, in particular, for access to non-standard control features of graphics devices, as opposed to non-standard geometric primitives. The GENERALIZED DRAWING PRIMITIVE function is designed for specification of non-standard primitives.

Errors:

Error identifier: 3:207
Cause: Requested escape not available.
Reaction: Function has no effect.

5.5.9 GET ESCAPE

Parameters:

In	escape function identifier	(n ≠ 0)	I
In	data record		D
Out	response validity	(INVALID, VALID)	E
Out	data record		D

Effect:

This soliciting function permits the return of non-standard information to the user.

The *escape function identifier* specifies the particular escape function information to be returned. Non-negative escape function identifiers are reserved for registration, negative escape function identifiers are available for private use.

If the Virtual Device encounters an unimplemented GET ESCAPE escape function identifier, it will return an empty output data record and a response validity value of INVALID.

Errors:

Error identifier: 3:207
Cause: Requested escape not available.
Reaction: Function has no effect.

5.5.10 STATE LIST INQUIRY SOURCE

Parameters:

In	source	(CURRENT, DEFAULT)	E

Effect:

This function sets the entry State List Inquiry Source in the Control State List.

When one of the functions listed below is invoked, its effect is determined by the current value of the State List Inquiry Source entry in the Control State List. If the current value is CURRENT, then the inquiry function shall return the current entry values of the relevant state list. If the current value is DEFAULT, then the default entry values of the relevant state list will be returned.

Note that INQUIRE MISCELLANEOUS CONTROL STATE (the inquiry function used to return the value of the State List Inquiry Source entry) is not one of the functions affected.

Inquiry functions affected by the State List Inquiry Source entry:

INQUIRE CONTROL STATE
INQUIRE CURRENT PRECISIONS REQUIREMENTS
INQUIRE VDC TO DEVICE MAPPING
INQUIRE ERROR HANDLING
INQUIRE LINE ATTRIBUTES
INQUIRE LIST OF LINE BUNDLE INDICES
INQUIRE LINE REPRESENTATION
INQUIRE MARKER ATTRIBUTES
INQUIRE LIST OF MARKER BUNDLE INDICES
INQUIRE MARKER REPRESENTATION
INQUIRE TEXT ATTRIBUTES
INQUIRE LIST OF TEXT BUNDLE INDICES
INQUIRE TEXT REPRESENTATION
INQUIRE FILL ATTRIBUTES
INQUIRE LIST OF FILL BUNDLE INDICES
INQUIRE FILL REPRESENTATION
INQUIRE PATTERN DIMENSIONS
INQUIRE PATTERN
INQUIRE LIST OF PATTERN INDICES
INQUIRE EDGE ATTRIBUTES
INQUIRE LIST OF EDGE BUNDLE INDICES
INQUIRE EDGE REPRESENTATION
INQUIRE OUTPUT STATE
INQUIRE OBJECT CLIPPING
INQUIRE LIST OF ATTRIBUTE SET NAMES IN USE
INQUIRE COLOUR STATE
INQUIRE LIST OF COLOUR TABLE ENTRIES
INQUIRE FONT LIST
INQUIRE CHARACTER SET LIST
LOOKUP ASPECT SOURCE FLAGS
INQUIRE SEGMENT STATE
INQUIRE LIST OF INHERITANCE FILTER SETTINGS
INQUIRE CLIPPING INHERITANCE
INQUIRE LIST OF SEGMENT IDENTIFIERS IN USE
INQUIRE INDIVIDUAL SEGMENT STATE LIST
INQUIRE COMMON LOGICAL INPUT DEVICE STATE
INQUIRE LIST OF ASSOCIATED TRIGGERS
INQUIRE ECHO DATA RECORD
INQUIRE INPUT DEVICE DATA RECORD
INQUIRE LOCATOR STATE

INQUIRE STROKE STATE
INQUIRE VALUATOR STATE
INQUIRE CHOICE STATE
INQUIRE PICK STATE
INQUIRE STRING STATE
INQUIRE RASTER INPUT STATE
INQUIRE GENERAL STATE
INQUIRE EVENT INPUT STATE
INQUIRE LIST OF CURRENTLY EXISTING ECHO ENTITIES
INQUIRE ECHO ENTITY STATE
INQUIRE ECHO OUTPUT DATA RECORD
INQUIRE RASTER STATE
INQUIRE LIST OF NON-DISPLAYABLE BITMAP IDENTIFIERS
INQUIRE LIST OF DISPLAYABLE BITMAP IDENTIFIERS
INQUIRE BITMAP STATE

6 Control inquiry functions

6.1 Introduction

6.1.1 Control inquiry functions

This clause defines the abstract functional specification of the Control Inquiry functions of the CGI.

The abstract names of these functions begin either with INQUIRE or LOOKUP. Those functions whose name start with INQUIRE will only return the values of one or more entries in a description table or state list. The functions whose names start with LOOKUP are used to determine if support exists for certain functions, Profiles, ESCAPEs, and GET ESCAPEs.

See ISO/IEC 9636-1, 5.2.7.

6.1.2 Data types employed

The abstract specifications of functions detail the functions in terms of input and output parameters. The data type of each parameter is selected from a standard set and is identified in the functional specification by a standard abbreviation. Both the data type and the abbreviation are taken from the complete list in ISO/IEC 9636-1, 5.2.10.

6.1.3 Validity of returned information

For all the inquiry functions specified in this clause, if any of the inquired information is available, the response validity flag is returned as VALID and the values specified in the output parameters are returned. In the case of a VALID response, if there is any possibility that any of the individual parameters within the response are not valid, then the response itself will contain additional (always valid) parameters which indicate which of the other returned parameters are valid.

If the inquired information is not available or the inquiry function is unsupported, the response validity flag is returned as INVALID and the specified output parameters are undefined. No other meaning should be applied to these other output parameters.

6.2 Device Identity Description Table

6.2.1 INQUIRE DEVICE IDENTIFICATION

Parameters:

In	maximum characters per string	(1..n)	I
Out	response validity	(INVALID, VALID)	E
Out	device class	(OUTPUT,INPUT,OUTIN)	E
Out	device identification		S

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.3 Output Device Description Table

6.3.1 INQUIRE DEVICE DESCRIPTION

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	hard/soft copy class	(HARD, SOFT)	E
Out	display type	(VECTOR, RASTER, OTHER)	E
Out	dynamic modification accepted for VDC-to-Device Mapping	(IRG, CBS, IMM)	E
Out	spontaneous change possible in display surface description	(NO, YES)	E
Out	display surface bottom-left corner		DP
Out	display surface upper-right corner		DP
Out	display surface width (millimetres)		R
Out	display surface height (millimetres)		R
Out	pixel location relative to coordinates	(ON, BETWEEN)	E

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.4 Function and Profile Support Description Table

6.4.1 LOOKUP FUNCTION SUPPORT

Parameters:

In	list of function identifiers		nFN
Out	response validity	(INVALID, VALID)	E
Out	list of support indicators	(NO, YES)	nE

Effect:

This function returns information about support for functions defined in all parts of ISO/IEC 9636. It is used to determine whether or not each function in the given *list of function identifiers* is supported. The number of elements in and the ordering of the returned list correspond to the *list of function identifiers*. See ISO/IEC 9636-1, annex A for the complete list of function identifiers specified by ISO/IEC 9636. This function will not return any information regarding private function support.

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.4.2 LOOKUP PROFILE SUPPORT

Parameters:

In	list of profile identifiers		nPRN
Out	response validity	(INVALID, VALID)	E
Out	list of support indicators	(NO, YES, UNRECOGNIZED)	nE

Effect:

This function returns information about the support of each profile in the given *list of profile identifiers*. The number of elements in and the ordering of the returned list correspond to the *list of profile identifiers*. The UNRECOGNIZED

response is indecisive; the implementation does not include explicit knowledge of the profile, and it may not satisfy the profile.

See ISO/IEC 9636-1, annex B for the complete list of profile identifiers specified by ISO/IEC 9636. See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.4.3 INQUIRE LIST OF PROFILE SUPPORT INDICATORS

Parameters:

In	number of list elements requested	(0..n)	I
In	index (within the list) of first element to return	(1..n)	I
Out	response validity	(INVALID, VALID)	E
Out	total number of list elements in description table	(1..n)	I
Out	list of profile/support indicators	(NO, YES)	n[PRN,E]

Effect:

See ISO/IEC 9636-1, annex B for the complete list of profile identifiers specified by ISO/IEC 9636. See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.5 Control Description Table

6.5.1 INQUIRE SUPPORTED VDC TYPES

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	VDC type support	(INTEGER, REAL, BOTH)	E

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.5.2 INQUIRE DEVICE CONTROL CAPABILITY

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	number of valid device viewport specification mode elements	(1..3)	I
Out	array of supported device viewport specification modes	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	3E
Out	message action detection support	(NO, YES)	E
Out	maximum number of characters for message	(-1,0..n)	I
Out	device viewport mirroring support	(NO, YES)	E
Out	number of valid deferral mode elements	(1..3)	I
Out	array of supported deferral modes	(ASTI, BNI, ASAP)	3E
Out	size of error queue	(-1,3..n)	I

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.5.3 LOOKUP ESCAPE SUPPORT

Parameters:

In	list of escape function identifiers		nI
Out	response validity	(INVALID, VALID)	E
Out	list of support indicators	(NO, YES)	nE

Effect:

This function returns information about whether each escape function in the given *list of escape function identifiers* is supported. The number of elements in and the ordering of the returned list correspond to the *list of escape function identifiers*.

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.5.4 LOOKUP GET ESCAPE SUPPORT

Parameters:

In	list of get escape identifiers		nI
Out	response validity	(INVALID, VALID)	E
Out	list of support indicators	(NO, YES)	nE

Effect:

This function returns information about whether or not each get escape function in the given *list of get escape identifiers* is supported. The number of elements in and the ordering of the returned list correspond to the *list of get escape identifiers*.

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.6 Control State List

6.6.1 INQUIRE CONTROL STATE

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	device drawing surface state	(DIRTY, CLEAN)	E
Out	deferral mode	(ASTI, BNI, ASAP)	E
Out	VDC type	(INTEGER, REAL)	E
Out	device viewport specification mode	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	E
Out	device viewport metric scale factor	(> 0)	R
Out	number of queued error reports	(0..n)	I

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.6.2 INQUIRE CURRENT PRECISION REQUIREMENTS

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	VDC integer precision log2 magnitude upper bound	(7..31)	I
Out	VDC real precision log2 magnitude upper bound		I

Control inquiry functions			Control State List
Out	VDC real precision log2 non-zero magnitude minimum		I
Out	VDC real precision log2 relative precision requirement	(≤ 0)	I
Out	VDC real precision log2 minimum magnitude for relative precision		I
Out	VDC real precision log2 typical magnitude		I
Out	integer precision log2 magnitude upper bound	(7..31)	I
Out	real precision log2 magnitude upper bound		I
Out	real precision log2 non-zero magnitude minimum		I
Out	real precision log2 relative precision requirement	(≤ 0)	I
Out	real precision log2 minimum magnitude for relative precision		I
Out	real precision log2 typical magnitude		I
Out	index precision log2 magnitude upper bound	(7..31)	I
Out	colour precision log2 upper bound	(1..32)	I
Out	colour index precision log2 upper bound	(8..32)	I
Out	client specified name precision log2 magnitude upper bound	(7..31)	I

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.6.3 INQUIRE VDC TO DEVICE MAPPING

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	VDC extent		2P
Out	isotropy	(NOT FORCED, FORCED)	E
Out	horizontal alignment	(LEFT,CENTRE,RIGHT)	E
Out	vertical alignment	(BOTTOM,CENTRE,TOP)	E
Out	specification mode of current device viewport	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	E
Out	metric scale factor of current device viewport	(> 0)	R
Out	requested device viewport		2VP
Out	effective viewport		2VP
Out	drawing surface clip indicator	(OFF, DSCRECT, VIEWPORT)	E
Out	specification mode of drawing surface clip rectangle	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	E
Out	metric scale factor of drawing surface clip rectangle	(> 0)	R
Out	drawing surface clip rectangle		2VP

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.6.4 INQUIRE ERROR HANDLING

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	array of error handling flags	(ON, REPORTING OFF, DETECTION OFF)	9E

Effect:

The *array of error handling flags* is ordered in increasing sequence correspond to error class 1 upto and including error class 9. See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

6.6.5 INQUIRE MISCELLANEOUS CONTROL STATE

Parameters:

Out	response validity	(INVALID, VALID)	E
Out	state list inquiry source	(CURRENT, DEFAULT)	E

Effect:

See 6.1.2 and ISO/IEC 9636-1, 5.2.7.

7 CGI description tables and state lists

The information in the description tables and state lists is available to a client by means of inquiry functions. The abstract specifications of these functions are found in this part of ISO/IEC 9636, clause 6.

7.1 Description tables

The description tables define the configuration of a given Virtual Device. This part of ISO/IEC 9636 describes the overall Device Description Tables, the Function and Profile Support Description Table, and the Control Description Table.

If this part of ISO/IEC 9636 allows latitude for certain description table entries, the preferred entries are designated by the entry being underlined.

Device classification information available in the Device Identification Description Table is: device class (OUTPUT, INPUT, or OUTIN) and a device-dependent device identification string.

The Function and Profile Support Description Table contains information regarding the function and resources support available in the CGI implementation.

Other description tables contain entries regarding hardware and technology dependencies. For devices of class OUTPUT or OUTIN, the Output Device Description Table includes physical device description data such as display surface size, native device coordinates, and display type. For devices of class INPUT or OUTIN, the physical device description data corresponding to the input side of the device are contained in the Logical Input Device Description tables (see ISO/IEC 9636-5). In general, the information in the description tables is static and cannot be changed by the client. Note that the display surface description can vary (will not be static) if the value of the Spontaneous Change Possible In Display Surface Description entry is YES.

The Control Description Table contains information on the VDC types supported by the device as well as the supported ESCAPE and GET ESCAPE identifiers. It also provides quantitative information regarding the capacity of relevant elements of the Virtual Device, such as the size of error queue. Such quantitative entries are static and indicate the number of constructs available immediately following INITIALIZE. The values do not change as resources are consumed.

Table 2 – Device Identification Description Table

Entry	Possible Values	Data Type
Device Class	(OUTPUT,INPUT,OUTIN)	E
Device Identification	device dependent	S

Table 3 – Output Device Description Table

Entry	Possible Values	Data Type
Hard/Soft Copy Class	(HARD, SOFT)	E
Display Type	(VECTOR, RASTER, OTHER)	E
Dynamic Modification Accepted for VDC-to-Device Mapping	(IRG, CBS, IMM)	E
Spontaneous Change Possible in Display Surface Description	(NO, YES)	E
Display Surface Bottom-Left Corner	*(imp.dep)*	DP
Display Surface Upper-Right Corner	*(imp.dep)*	DP
Display Surface Width	*(imp.dep)*	R (Note 1)
Display Surface Height	*(imp.dep)*	R (Note 1)
Pixel Location Relative to Coordinates (Note 2)	(ON, BETWEEN)	E

NOTES
1) These are measured in millimetres.
2) The Pixel Location Relative to Coordinates entry is only applicable if the Display Type is RASTER.

Table 4 – Function and Profile Support Description Table

Entry	Possible Values	Data Type
Function Support:		
List of Supported Functions	(Note 1)	nFN
Profile Support:		
List of Profile Support Pairs each containing:		
Profile Identifier	(Note 2)	PRN
Support Indicator	(NO, YES)	E

NOTES

1) The list encompasses all functions defined in all implemented parts of ISO/IEC 9636. See ISO/IEC 9636-1, annex A for the complete list and range of standardized function identifiers.

2) See ISO/IEC 9636-1, annex B for the range of standardized profile identifiers.

Table 5 – Control Description Table

Entry	Possible Values	Data Type
Supported VDC Types:		
VDC Type Support	(INTEGER, REAL, BOTH)	E
Device Control Capability:		
Number of Valid Device Viewport Specification Mode Elements	(1..3)	I
Array of Supported Device Viewport Specification Modes	(FRACTION OF DISPLAY, SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	3E
Message Action Detection Support	(NO, YES)	E
Maximum Number of Characters for Message	(-1, 0..n) (Note 1)	I
Device Viewport Mirroring Support	(NO, YES)	E
Number of Valid Deferral Mode Elements	(1..3)	I
Array of Supported Deferral Modes	(ASTI, BNI, ASAP)	3E
Size of Error Queue	(-1,3..n)(Note 1)	I
ESCAPE support:		
List of Supported Escape Functions	(-n..-1, 1..n)	nI
GET ESCAPE support:		
List of Supported Get Escape Functions	(-n..-1, 1..n)	nI

NOTE –

1) For entries denoting the maximum number of supported entities, -1 indicates either no restriction on the number or that support is only limited by available memory (for an implementation using dynamic memory allocation).

7.2 State lists

State lists contain the details of the current state of the Virtual Device. The Control State List described in this clause relates to states controlled by functions described in this part of ISO/IEC 9636.

Table 6 – Control State List

Entry	Possible Values	Data Type	Default
Control State:			
Device Drawing Surface State	(DIRTY, CLEAN)	E	(Note 8)
Deferral Mode	(ASTI, BNI, ASAP)	E	ASTI
VDC Type	(INTEGER, REAL)	E	(Note 7)
Device Viewport Specification Mode	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES) (Note 1)	E	FRACTION OF DISPLAY SURFACE
Device Viewport Metric Scale Factor	(> 0)	R	1.0
Number of Queued Error Reports	(0..n)	I	0
Current Precision Requirements:			
VDC Integer Precision Log2 Magnitude Upper Bound	(7..31)	I	*(imp.dep)*
VDC Real Precision Log2 Magnitude Upper Bound		I	*(imp.dep)*
VDC Real Precision Log2 Non-Zero Magnitude Minimum		I	*(imp.dep)*
VDC Real Precision Log2 Relative Precision Requirement	(≤ 0)	I	*(imp.dep)*
VDC Real Precision Log2 Minimum Magnitude for Relative Precision		I	*(imp.dep)*
VDC Real Precision Log2 Typical Magnitude		I	*(imp.dep)*
Integer Precision Log2 Magnitude Upper Bound	(7..31)	I	*(imp.dep)*
Real Precision Log2 Magnitude Upper Bound		I	*(imp.dep)*
Real Precision Log2 Non-Zero Magnitude Minimum		I	*(imp.dep)*
Real Precision Log2 Relative Precision Requirement	(≤ 0)	I	*(imp.dep)*
Real Precision Log2 Minimum Magnitude For Relative Precision		I	*(imp.dep)*
Real Precision Log2 Typical Magnitude		I	*(imp.dep)*
Index Precision Log2 Magnitude Upper Bound	(7..31)	I	*(imp.dep)*
Colour Precision Log2 Upper Bound	(1..32)	I	*(imp.dep)*
Colour Index Precision Log2 Upper Bound	(8..32)	I	*(imp.dep)*
Client Specified Name Precision Log2 Magnitude Upper Bound	(7..31)	I	*(imp.dep)*
VDC-to-Device Mapping:			
VDC Extent		2P	(Note 4)
Isotropy	(NOT FORCED, FORCED)	E	FORCED
Horizontal Alignment	(LEFT, CENTRE, RIGHT)	E	LEFT
Vertical Alignment	(BOTTOM, CENTRE, TOP)	E	BOTTOM
Specification Mode of Current Device Viewport	(Note 1) (FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	E	FRACTION OF DISPLAY SURFACE
Metric Scale Factor of Current Device Viewport	(> 0)	R	1.0
Requested Device Viewport	(Note 2)	2VP	(0.0, 0.0), (1.0, 1.0)
Effective Viewport	(Note 2)	2VP	(Note 5)

Table 6 – Control State List — (concluded)

Entry	Possible Values	Data Type	Default
Drawing Surface Clipping:			
Drawing Surface Clip Indicator	(OFF, DSCRECT, VIEWPORT)	E	VIEWPORT
Specification Mode of Drawing Surface Clip Rectangle	(FRACTION OF DISPLAY SURFACE, MILLIMETRES WITH SCALE FACTOR, PHYSICAL DEVICE COORDINATES)	E	FRACTION OF DISPLAY SURFACE
Metric Scale Factor of Drawing Surface Clip Rectangle	(> 0)	R	1.0
Drawing Surface Clip Rectangle		2VP	(Note 3)
Error Handling:			
Array of Error Handling Flags (for error classes 1 through 9)	(ON, REPORTING OFF, DETECTION OFF)	9E	(Note 6)
Miscellaneous Control State:			
State List Inquiry Source	(CURRENT, DEFAULT)	E	CURRENT

NOTES

1) Applies to both the requested device viewport and the effective viewport.
2) The requested device viewport and the effective viewport will only differ when isotropy is forced.
3) Drawing Surface Clip Rectangle defaults to a rectangle bounded by Display Surface Bottom Left Corner and Display Surface Top Right Corner as specified in the Output Device Description Table of this part of ISO/IEC 9636.
4) (0, 0), (32767, 32767) if VDC Type is INTEGER. (0.0, 0.0), (1.0, 1.0) if VDC Type is REAL.
5) The default effective viewport depends on the aspect ratio of the physical display surface and is therefore device-dependent.
6) For error classes 1 through 7, the default is REPORTING OFF. For classes 8 and 9, the default is DETECTION OFF.
7) The preferred default VDC type is INTEGER.
8) If an implementation is able to determine that the display surface is unmarked at initialization time, it should set the entry to CLEAN. Otherwise, the entry should be set to DIRTY.

Annex A

(normative)

Formal Grammar of the Functional Specification

A.1 Introduction

This grammar is a formal definition of the Control portion of standard CGI syntax. It shows all productions regardless of the encoding scheme. The terminal symbols correspond to the CGI basic abstract data types. Encoding and representation details of these can be found in ISO/IEC 9637-1 and ISO/IEC 9637-2.

A.2 Notation used

<symbol>	- nonterminal
<SYMBOL>	- terminal
<symbol>*	- 0 or more occurrences
<symbol>+	- 1 or more occurrences
<symbol>o	- 0 or 1 occurrences
<symbol>(n)	- exactly n occurrences; n=2,3,...
<symbol-1> ::= <symbol-2>	- symbol-1 has the syntax of symbol-2
<symbol-1> | <symbol-2>	- symbol-1 or alternatively symbol-2
<symbol: meaning>	- symbol with the stated meaning
comment	- explanation of a symbol or a production
returned: <symbol>*	- output parameter(s)

A.3 Control functions

```
<control function>  ::=   <virtual device management function>
                        | <inquiry function>
                        | <coordinate space function>
                        | <error function>
                        | <miscellaneous control function>
```

A.3.1 Virtual Device management functions

```
<virtual device management function>  ::=   <INITIALIZE>
                                            | <TERMINATE>
                                            | <PREPARE DRAWING SURFACE>
                                                <clear drawing surface>
                                            | <EXECUTE DEFERRED ACTIONS>
                                            | <END PAGE>
                                            | <SET DEFERRAL MODE>
                                                <deferral mode>

<clear drawing surface: enumerated>  ::=  <UNCONDITIONAL> | <CONDITIONAL>

<deferral mode: enumerated>          ::=  <ASTI> | <BNI> | <ASAP>
```

A.3.2 Coordinate space functions

```
<coordinate space function>  ::=  <VDC TYPE>
```

<vdc type>
| <VDC EXTENT>
<point>(2)
| <VDC INTEGER PRECISION REQUIREMENT>
<BDT_INTEGER: log2 magnitude upper bound>
| <VDC REAL PRECISION REQUIREMENTS>
<BDT_INTEGER: log2 magnitude upper bound>
<BDT_INTEGER: log2 non-zero magnitude minimum>
<BDT_INTEGER: log2 relative precision requirement>
<BDT_INTEGER: log2 minimum magnitude for relative precision>
<BDT_INTEGER: log2 typical magnitude>
| <DEVICE VIEWPORT>
<viewport point>(2)
| <DEVICE VIEWPORT SPECIFICATION MODE>
<viewport specification mode>
<BDT_REAL: metric scale factor>
| <DEVICE VIEWPORT MAPPING>
<isotropy flag>
<horizontal alignment flag>
<vertical alignment flag>
| <DRAWING SURFACE CLIP RECTANGLE>
<viewport point>(2)
| <DRAWING SURFACE CLIP INDICATOR>
<drawing surface clip indicator>

<point> ::= <vdc value>(2)

<vdc value> ::= <BDT_REAL> | <BDT_INTEGER>

<viewport point> ::= <vc value>(2)

<vc value> ::= <BDT_REAL> | <BDT_INTEGER>

<vdc type: enumerated> ::= <INTEGER> | <REAL>

<viewport specification mode: enumerated> ::= <FRACTION OF DISPLAY SURFACE>
| <MILLIMETRES WITH SCALE FACTOR>
| <PHYSICAL DEVICE COORDINATES>

<isotropy flag: enumerated> ::= <NOT FORCED> | <FORCED>

<horizontal alignment flag: enumerated> ::= <LEFT> |<CENTRE> | <RIGHT>

<vertical alignment flag: enumerated> ::= <BOTTOM> | <CENTRE> | <TOP>

<drawing surface clip indicator: enumerated> ::= <OFF> | <DSCRECT> | <VIEWPORT>

A.3.3 Error functions

<error function> ::= <dequeue error reports>
| <error handling control>

<dequeue error reports> ::= <DEQUEUE ERROR REPORTS>
<BDT_INTEGER: number requested>
returned:
<response validity>
<BDT_INTEGER: number of reports remaining>
<error report>*

<error handling control> ::= <ERROR HANDLING CONTROL>
<error control>(9)

<response validity> ::= <validity flag>