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

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

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.)

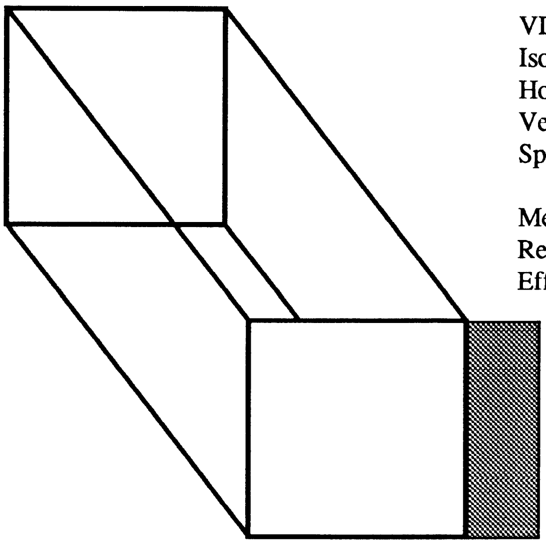
3.3.4 VDC space and range

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.



VDC-to-Device Mapping

VDC Extent = ((0, 0), (32676, 32676))
Isotropy = FORCED
Horizontal Alignment = LEFT
Vertical Alignment = BOTTOM
Specification Mode of Current Device Viewport =
FRACTION OF DISPLAY SURFACE
Metric Scale Factor of Current Device Viewport = 1.0
Requested Device Viewport = ((0.0, 0.0), (1.0, 1.0))
Effective Device Viewport = ((0.0, 0.0), (0.666, 1.0))

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).

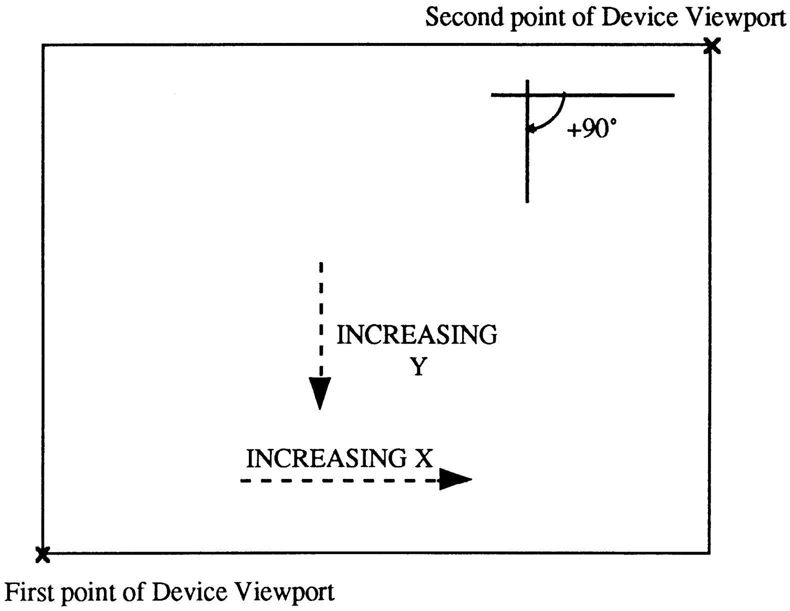
Whether changes to the VDC-to-Device Mapping take place immediately, can be simulated, or lead to an implicit regeneration, is determined by the Dynamic Modification Accepted For VDC-to-Device Mapping entry in the Output Device Description Table.

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°.

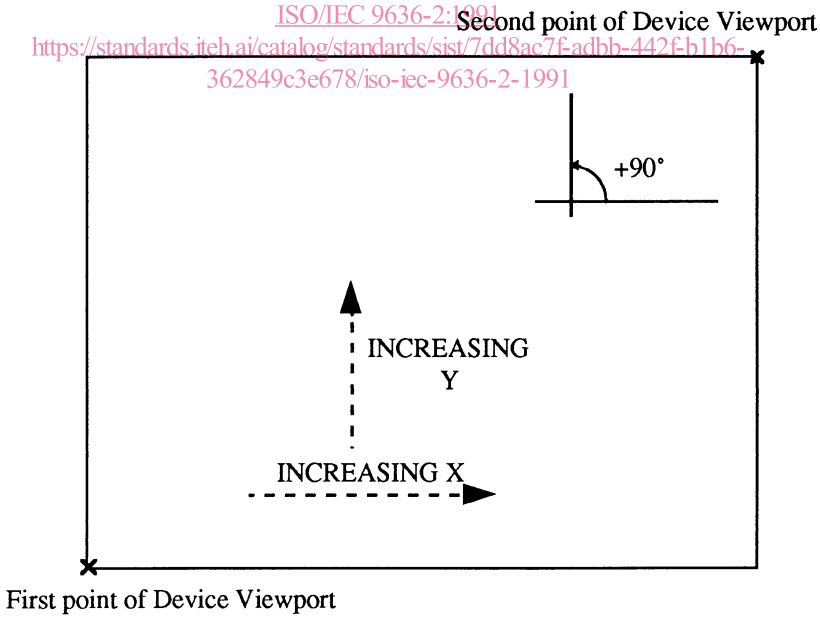
3.3.6 VDC tailoring

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.



VDC extent (0.0, 0.0), (1.0, 0.75)



VDC extent (0.0, 8.5), (11.0, 0.0)

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.