INTERNATIONAL STANDARD

ISO/IEC 9636-1

First edition
1991-12-15

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

Part 1:
Overview, profiles, and conformance

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

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

Partie 1: Résumé, profils et conformité

Reference number
ISO/IEC 9636-1:1991(E)

Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8c145c5-318a-4579-bbee-a86129749cd5/iso-iec-9636-1-1991

© ISO/IEC 1991
All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

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 9636-1 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*

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Annexes A and B form an integral part of this part of ISO/IEC 9636. Annexes C, D, E and F are for information only.

ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

Introduction

Purpose

The Computer Graphics Interface (CGI) specifies a standard interface between device-independent and device-dependent parts of a graphics system. ISO/IEC 9636 specifies sets of functions for control and data exchange over this interface. This interface may be implemented as a software-to-software interface (as a data stream encoding used in a network environment or as a procedural binding to one or more software packages), or as a software-to-hardware interface (as a data stream encoding to one or more devices presenting a standardized Computer Graphics Interface as their device protocol). Related standards specify data stream encodings (for use in the software-to-software case, over a network, and in the software-to-hardware case), and procedural bindings (for use in the software to software case).

ISO/IEC 9636 includes a reference model and a description of the CGI's relationship to other standards.

ISO/IEC 9636 only defines graphics functions, graphics control functions, and functions to control data representations and deferral in data stream encodings. ISO/IEC 9636 does not define the protocols to be used to convey these functions between the parts of a (potentially distributed) system.

Benefits

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Intrinsic The CGI will simplify the development and implementation of graphics systems. ISO/IEC 9636 will encourage a uniform access to the graphics devices within an installation. As new devices are made available, graphics device drivers adhering to this interface can be installed for use by existing programs.

Interchange ISO/IEC 9636 promotes the exchange of software between installations. By isolating the device-dependent aspects of any graphics system, modularity is encouraged, which promotes increased portability. The standard set of functions, access mechanisms, and terminology will allow developers and users to move between installations with minimal retraining.

ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4970-bbee-a06129749cd5/iso-iec-9636-1-1991

Educational The standard set of functions uses a standard terminology. This allows both the academic and industrial communities to develop instructional programs concentrating on programming techniques and methodologies based on these standard functions.

Economic In view of the trend towards lower hardware and higher software costs, the following benefits accrue from ISO/IEC 9636:

- It encourages transporting of software between installations, thereby reducing costs associated with “reinvention”;
- It protects the large software investment made by both users and vendors because the software will not be rendered obsolete by the introduction of new devices;
- It allows developers of new software to focus on higher-level graphics functions and applications instead of device-level functions;
- It reduces maintenance of software systems because the standard encourages modularity;
- It increases vendor independence for the user because any system designed to use a particular device can more easily be changed to use some other device;
- It allows vendors to develop and market devices that will easily interface to the customer's system;
- It enables users, manufacturers and vendors to take advantage of new, lower-cost graphics hardware designs. The total system's hardware cost may be reduced because system redesign may not be necessary.

Design requirements

To realize the benefits described above, a number of design principles have been adopted:

a) The Computer Graphics Interface should provide a suitable set of functions for the description of a wide range of pictorial information;

b) The Computer Graphics Interface should provide a suitable set of functions for the necessary CGI session control of a wide range of graphics devices;

c) The Computer Graphics Interface should address the more usual and essential features found on graphical devices directly and should provide access to less common facilities;

d) The design of the Computer Graphics Interface should not preclude extension of ISO/IEC 9636 at a later stage to cover facilities currently not standardized;

e) The Computer Graphics Interface should be usable from GKS (Graphical Kernel System - ISO 7942). In particular, the CGI should include functional capability to support the various levels of a GKS workstation in an efficient and concise manner, without compromising the ability of the interface to support non-GKS systems in an efficient and concise manner;

f) The Computer Graphics Interface should be compatible with the Computer Graphics Metafile - ISO 8632. In particular, those CGM elements not associated with the file-oriented aspects of the CGM shall have corresponding CGI functions which have identical abstract names and parameterization;

g) ISO/IEC 9636 should address the needs of different applications that have conflicting requirements for

- allocation of processing burden between host and device;
- speed of generation and interpretation of functions;
- ease of transfer through different transport mechanisms.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

Design criteria

The above requirements were used to formulate the following design criteria:

a) Completeness
In any area of ISO/IEC 9636, the functionality specified by ISO/IEC 9636 should be complete in itself.

b) Conciseness
Redundant functions or parameters should be avoided.

c) Consistency
Contradictory functions should be avoided.

d) Extensibility
The ability to add new functions and generality to ISO/IEC 9636 should not be precluded.

e) Fidelity
The results and behaviour of functions should be well defined.

f) Implementability
A function should be able to be efficiently supported on most host systems andor graphics hardware.

g) Orthogonality
Independent functions for separate and noninteracting activities should be provided.

h) Predictability
The recommended or proper use of a standard function should guarantee the results of using that particular function.

i) Standard practice
Only those functions that reflect existing practice, that are necessary to support existing practice, or that are necessary to support standards being developed concurrently should be standardized.

j) Usefulness
Functions should be powerful enough to perform useful tasks.

k) Well-structured
The number of assumptions that functions make about each other should be minimized. A function should have a well-defined interface and a simply stated unconditional purpose. Multi-purpose functions and side effects should be avoided.

Parts of the CGI functional specification

ISO/IEC 9636, the functional specification of the Computer Graphics Interface, consists of a number of parts presenting portions of the CGI functionality, including an overview in this part of ISO/IEC 9636.

Table 1 – Parts of the CGI Functional Specification

Part No.	Title
ISO/IEC 9636-1	Overview, profiles, and conformance
ISO/IEC 9636-2	Control
ISO/IEC 9636-3	Output
ISO/IEC 9636-4	Segments
ISO/IEC 9636-5	Input and echoing
ISO/IEC 9636-6	Raster

This part of ISO/IEC 9636 gives a general overview and introduction to the basic concepts and principles of ISO/IEC 9636. It includes a reference model, the relationship to other standards, and profiles. In addition, it contains overviews of each of the subsequent parts. This part of ISO/IEC 9636 thus establishes the framework for all the parts of ISO/IEC 9636; it does not contain functional descriptions.

The functional capability provided by the CGI is separate from the specification of any particular encoding format or language binding.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)

This page intentionally left blank

ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/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

1 Scope

ISO/IEC 9636 establishes the conceptual model, functional capability, and minimum conformance requirements of the Computer Graphics Interface (CGI). It specifies design requirements for encodings of the CGI. ISO/IEC 9636 defines a set of CGI functions that is expected to satisfy the following needs of a majority of the computer graphics community:

a) provide an interface standard for computer graphics software package implementors;

b) provide an interface standard for computer graphics device manufacturers and suppliers;

c) provide an inquiry and response mechanism for graphics device capabilities, characteristics, and states;

d) provide a standard graphics escape mechanism to access non-standard graphics device capabilities;

e) allow for future functional extension of the CGI.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145e5-318c-4770-bbee-a06129749cd5/iso-iec-9636-1-1991

In addition to the CGI functionality, device classes, and Foundation and Constituency Profiles are defined. The device classes included in the CGI are output (OUTPUT), input (INPUT), and output/input (OUTIN). Profiles allow subsets of the CGI functions and features to be defined to suit particular well identified groups of users. There is also provision for Constituency Profiles to be registered after ISO/IEC 9636 is published. The Computer Graphics Interface (CGI) is a standard functional and syntactical specification of the control and data exchange between device-independent graphics software and an implementation of a CGI Virtual Device.

The syntax of the CGI, presented in ISO/IEC 9636, is an encoding-independent and binding-independent specification. Any similarity of the examples or function specifications to a particular encoding technique or language is coincidental unless explicitly stated otherwise.

The functions specified provide for the representation of a wide range of two-dimensional pictures and for control over their display on a wide range of graphics devices. The functions are split into groups that perform device and CGI session control, specify the data representations used, control the display of the picture, perform basic drawing actions, control the attributes of the basic drawing actions, acquire data from input devices, and provide access to non-standard device capabilities.

This part of ISO/IEC 9636 gives an overview of ISO/IEC 9636, explains the relationship between its parts and their relation to other standards, describes a reference model for graphics systems, and defines certain Foundation and Constituency Profiles. ISO/IEC 9636-2, ISO/IEC 9636-3, ISO/IEC 9636-4, ISO/IEC 9636-5, and ISO/IEC 9636-6 specify the CGI functions for different functional areas using an abstract notation.

ISO/IEC 9637 and ISO/IEC 9638 define standard data stream encodings, procedural library bindings, and single entry point procedural bindings of the CGI.

1.1 Relationship of CGI to a computing environment

ISO/IEC 9636 describes graphical services provided by a Virtual Graphics Device. The model for description of these services is expressed in terms of graphical capabilities of a single instance of a hypothetical graphics device. In all but the simplest of computing environments, CGI functions alone will not be sufficient to provide complete control over a device. Additional functions, not included in ISO/IEC 9636, will likely be needed. Examples of such functions include

- means to configure (sets of) physical devices to be accessed as CGI Virtual Devices;
- means to control a device capable of offering CGI-defined services as well as other, non-CGI-defined services, such as those implied by ISO 2022 and ISO 6429;
- means to differentiate among separate instances of CGI Virtual Devices in the same computing environment;
- means of defining or determining communication paths from CGI clients to CGI Virtual Devices.

In some cases, other standards exist that describe the functions required. For example, various communications standards address the needs of the last point above. In other cases, no standards may exist, but the tasks indicated are outside the scope of ISO/IEC 9636.

1.2 Position of CGI in a managed environment

There exists a large and growing family of computer controlled display systems that have the ability to act as if they are multiple individual display devices. Resources, most notably the visible drawing surface resources, are coordinated by the display system so that multiple non-cooperating client programs can each access the services of a separate individual device while all are actually running in a single managed environment.

The graphical capabilities of the CGI Virtual Device may suffice, in some instances, as the basis for implementing a complex, multiple-client display system. However, the complete needs of such a system are quite complex, include many non-graphical services, and (as current practice shows) are quite technology dependent. The CGI does not, therefore, purport to be a generally sufficient interface on which a managed display environment may be built. Rather, within a managed environment, the CGI will be one of the managed interfaces in a way not visible to the CGI client without recourse to services not part of ISO/IEC 9636. The use of the CGI as a managed interface within a managed display environment is not limited to raster devices.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

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 646 : 1983 *Information processing – ISO 7-bit coded character set for information interchange.*

ISO 2022 : 1986 *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.*

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

ISO 6429 : 1988 *Information processing – Control functions for 7-bit and 8-bit coded character sets.*

ISO 7942 : 1985 *Information processing systems – Computer graphics – Graphical Kernel System (GKS) functional description.*

ISO 7498 : 1984 *Information processing systems – Open systems interconnection – Basic reference model.*

ISO 8632-1 : 1987 *Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information – Part 1: Functional specification.*

ISO 8632-2 : 1987 *Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information – Part 2: Character encoding.*

ISO 8632-3 : 1987 *Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information – Part 3: Binary encoding.*

ISO 8805 : 1988 *Information processing systems – Computer graphics – Graphical Kernel System for Three Dimensions (GKS-3D) functional description.*

ISO 9282-1 : 1988 *Information processing – Coded representation of pictures – Part 1: Encoding principles for picture representation in a 7-bit or 8-bit environment.*

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

ISO/IEC 9592-1 1989 *Information processing systems – Computer graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS) – Part 1: Functional description.*

ISO/IEC 9636-2 : 1991 *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 2: Control.*

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.*

ANSI/IEEE 754 – *Standard for Binary Floating Point Arithmetic.*

[1] To be published.

3 Definitions and abbreviations

3.1 Organization of the definitions

The following list of definitions describes the usage of words or phrases which have been given special technical meaning in the context of ISO/IEC 9636. These meanings are not in conflict with dictionary meanings or usage in other ISO and IEC standards. However, there are usually additional semantics associated with the words or phrases which are essential to understanding ISO/IEC 9636.

The list of definitions is arranged in alphabetical order of main entry. Occasionally a term will be used with or without optional qualification in the text; in such cases, the optional qualification appears in parentheses in the term. Under each main entry, certain derived terms may also be covered; all such derived terms are underlined. With a few exceptions, where the appropriate main heading may not be immediately obvious, derived terms do not have separate main entries. Where a phrase appears in italic text, its definition may be as another main entry or it may be a derived term – in which case it will be found either under an entry for one of its constituents (not necessarily its first), under another compound entry with the same first constituent, or under an entry for another word with the same root.

3.2 Definitions

For the purposes of ISO/IEC 9636, the following definitions apply.

3.2.1 acknowledgement

An action performed automatically by a *logical input device* signals to the *operator* that an *input operation* has been completed. Initially enabled by default, this action is controllable by the *client* using acknowledgement control. In a *remote echoing* situation, the acknowledgement has to be explicitly performed by the client. Various acknowledgement may be available and the client can select from these different styles of signalling.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

3.2.2 active (for a LID)

A *logical input device* is active if events or sampling have been enabled or if request or echo request input is in progress; otherwise, it is not active.

3.2.3 allowed latitude

There are a number of cases in which ISO/IEC 9636 identifies a limited set of allowed behaviours for functions or features and allows implementors latitude to provide precisely one of this set. In such cases, a *description table* entry gives a *client* the opportunity to inquire the implemented behaviour. ISO/IEC 9636 always identifies one of these allowed behaviours as a preferred behaviour. This method of regulating implementation variability contrasts with that referred to as *implementation dependence*. Where ISO/IEC 9636 allows latitude in the value of a state list default, then a particular value is identified as the preferred default. An example is the default VDC Type, which has allowed latitude in its value, and a preferred default of INTEGER.

3.2.4 appending (to a segment)

See under *segment definition*.

3.2.5 arming (a trigger)

An action which enables a *trigger* to be *fired* by the *operator* to signal the occurrence of something significant. A trigger cannot be fired unless it has been armed. Triggers are armed for a *logical input device* while events are enabled for it and while a request or echo request input operation is in progress.

3.2.6 aspect ratio

The ratio, using a uniform metric, of the width to the height (i.e. x to y) of a rectangular area, such as a *VDC extent* or a *device viewport*.

3.2.7 aspect source flag (ASF)

Aspect source flags which are themselves *attribute values* indicate the source of other attribute values which determine *aspects* of *primitives* or *objects*. If the value of a particular aspect source flag is INDIVIDUAL, then the attribute value for that particular aspect is *associated* with the primitive or object. If the value of a particular aspect source flag is BUNDLED, then the corresponding *bundle index* attribute value associated with the primitive or object is used to access a *bundle table* to find the attribute value determining the aspect in question.

3.2.8 aspects (of primitive and object)

Characteristics determining the visual (rendered) appearance of *primitives* or *objects* are known as aspects. A particular aspect of a particular primitive or object is determined by an *attribute value* which may be directly *associated* with it or indirectly associated with it by means of a *bundle index* as determined by the appropriate *aspect source flag*. There are other characteristics of primitives and objects which are determined by attribute values, but if these are not concerned with visual appearance or are only specifiable directly (i.e. they cannot be bundled), they are not referred to as aspects.

3.2.9 association (of attribute values)

This refers to various steps in the *graphic object pipeline* where the connection between *graphic objects* and *attribute values* relevant to their *rendering* is established. For *aspects* relevant to a given *primitive type*, the corresponding attribute values are associated during the initial step of object creation and later, just before rendering, for those aspects for which the (already associated) *aspect source flag* values are bundled. For objects stored in segments, the associated object transformation attribute value can be modified by the concatenation of a segment transformation, which is a segment attribute, or a copy transformation, which is a parameter to the copy segment function. Associated attribute values can be replaced by means of the copy segment function and the inheritance filter.

3.2.10 association (of triggers with LIDs)

see under *trigger association.*

3.2.11 attribute function

A function which sets an *attribute value* in a *state list*. Unqualified, the term is used for functions which set *primitive* or *object* attribute values. If other attributes are intended, the term will be appropriately qualified, e.g. segment attribute function or raster attribute function.

3.2.12 attribute value

Certain characteristics of entities are determined by attribute values. When used without qualification, the term implies *primitive* or *object* attribute value. All other uses of attribute value will be appropriately qualified such as *segment attribute value*, *raster attribute value*, etc. Many primitive or object attribute values are directly concerned with rendered appearance and determine *aspects*. Others may not be directly or at all concerned with rendered appearance (such as *ASFs* and pick identifiers). When a single primitive forms a single object, there is no distinction between primitive attribute value and object attribute value. However, when several primitives contribute to one object (*compound text* or *closed figure*), the primitive attribute values are associated with each contributing primitive (see *local attribute value*) while object attribute values are associated with the object as a whole (see *global attribute value*). As an example, auxiliary colour is a primitive (local) attribute value for *edges* but an object (global) attribute value for *interiors* of *fill objects.*

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbea-a061297d9cd5/iso-iec-9636-1-1991

3.2.13 auxiliary colour

The auxiliary colour attribute value *associated* with an *object* is used in conjunction with the transparency attribute value. See under *transparency.*

3.2.14 background colour

The colour used in preparing the *drawing surface.*

3.2.15 background colour (of mapped bitmap)

A *control value* that determines the full depth value to which the background *pixels* of a *mapped bitmap* are expanded when they participate in *raster operation functions.*

3.2.16 binding

A binding is a (concrete) realization of an abstract functional specification in a programming language. This may be in terms of a procedural library binding to a standard programming language with approximately one call per function or a procedural single entry point binding of the abstract functions with all input and output parameters. Procedural single entry point bindings are based on data stream encodings; see *encoding*. In the absence of any clear context implying the contrary, "procedural binding" or "binding" may be taken to refer to a procedural library binding.

3.2.17 bitblt

BIT aligned BLock Transfer. The transfer or combination of the *pixel values* in rectangular *bitmap regions* of *bitmaps*. The CGI provides two and three operand bitblt functions where there are one or two source bitmap regions respectively, and a destination bitmap region. Arbitrary logical combinations and some arithmetic combinations are supported by *drawing mode*. See *raster operation functions.*

3.2.18 bitmap

A resource of the *Virtual Device* which may be viewed by the *client* as a rectangular array of *pixels*. This resource is available only on devices offering the functional capability defined in ISO/IEC 9636-6. Some bitmaps are inherent resources of the device; others can be defined by the client. CGI provides for full depth bitmaps, where the values assignable to the individual pixels span the total colour or grey scale capability of the device, and for mapped bitmaps, where the values assignable to a pixel are only "foreground" or "background"; it does not provide arbitrary variable depth bitmap capability. Any bitmap may be selected as the drawing bitmap which is the destination (*drawing surface*) for the *rendering* of (graphic) *objects* as well as for *raster operation functions*. Bitmaps which are displayable bitmaps may also be selected as the display bitmap.

3.2.19 bitmap region

see *region*.

3.2.20 blocked (of a queue)

A queue is said to be blocked if the *client* has taken specific action to prevent the entry of elements into it. Contrast with *overflow* where there may or may not be implicit system action to "block" a queue in overflow state.

3.2.21 boundary

The boundary of a *fill object* consists of explicit boundary portions, implicit boundary portions, and clip boundary portions. The boundary of a fill *primitive* consists of the mathematical locus of its defining perimeter. During *closed figure* construction, the mathematical locus of each line primitive used (including CONNECTING EDGE) and the boundary of each fill primitive used defines an explicit boundary portion. Implicit boundary portions are those added automatically during closed figure construction to ensure closure. (The interpretation of the significance of edge out flags in the POLYGON SET function means that it can also introduce both explicit boundary portions and implicit boundary portions.) When a fill object undergoes locus clipping, pieces of the original closed boundary lying outside the effective clip region are discarded and clip boundary portions are added to maintain closure. Drawing surface clipping does not contribute any clip boundary portions. Clip and implicit boundary portions have no associated *attribute values*. Edge portions correspond to the explicit boundary portions and are rendered in accordance with their associated edge attribute values. Contrast with *edge*, and see also *interior*, *realized edge*, and *realized interior*.

3.2.22 break action

A specific action by the *operator* to indicate that an *input operation* should be aborted. Contrast with *firing (a trigger)* and with *timeout*.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

3.2.23 bundle

Virtual Devices have a conceptual resource known as bundle tables. Bundle tables contain *attribute values* determining *aspects* of *primitives* or *objects*. There are different bundle types – one each for line, marker, text, (fill object) edge, and fill (object interior) aspects. A bundle table consists of all bundles of a given type; each of which is uniquely identified by its index. For a particular primitive or object, those aspects for which an attribute value from a bundle table is to be used must have the appropriate *aspect source flag* set to BUNDLED. In this case, the bundle which contains the relevant values is specified by another attribute value, the bundle index into the table of the appropriate type.

3.2.24 cell (of CELL ARRAY or pattern)

A subspace of *VDC space* may be divided into a regular array of smaller spaces referred to as cells. Cells occur in CELL ARRAY and in *patterns* used for interior filling of *fill objects*. In general, the subspace and the individual cells may be rectangular or skewed (i.e. a parallelogram) and of arbitrary *aspect ratio*. Some devices may not provide full support for arbitrary sized or skewed cells.

3.2.25 cell (of character cell)

The subspace of *VDC space* determined by text *attribute values* (and the positioning information for the containing text *object*) which is conceptually "occupied" by a character. This is a parallelogram; but since attributes can change within a text object and proportional spacing and kerning may be relevant, character cells may not be a uniform array as in the above definition. A character cell is relevant for positioning of subsequent character cells and also for where *auxiliary colour* is *rendered* in the remainder of the cell after the rendering of the actual glyph in *foreground colour* (in this case, text colour).

3.2.26 CGI Generator

Any agent which forms an *encoding* or *binding* of CGI functions and passes this to a *CGI interpreter*. The same agent may also interpret any responses.

3.2.27 CGI Interpreter

An implementation of a CGI *Virtual Device* which receives an *encoding* or *binding* of the CGI functions, performs the appropriate actions, and provides any required responses.

3.2.28 character set
A set of displayable glyphs represented by the individual values occurring in a text *primitive*. The particular values in the strings are dependent on the character set or the alternate character set which is referenced by the character set index *attribute value* of the text primitive and are subject to character coding announcer control. Character set is concerned with (logical) alphabet and special glyphs (national character sets, Katakana, etc.) and is orthogonal to *font* which determines the typeface and style of the actual *rendering* of the glyphs (e.g. italic, bold).

3.2.29 class
see *input class* and *device class*.

3.2.30 client
Any agent invoking functions of the abstract functional capability defined in ISO/IEC 9636; i.e. any entity invoking a *procedural binding* or generating an output data stream *encoding*. Contrast with *operator* and *CGI interpreter*.

3.2.31 clip boundary portion
see *boundary*.

3.2.32 clipping
A process by which portions of (graphic) *objects* which fall outside an *associated* object clip rectangle or portions of a *rendering* of a picture or raster operation which fall outside a *drawing surface* clip rectangle are discarded. Whether or not such discarding takes place is determined by the object or drawing surface clip indicator. The object clip rectangle and indicator are object *attribute values* whereas the drawing surface clip rectangle and indicator are *control values*. See *object clipping mode*.

3.2.33 closed figure
A *fill object* constructed on the device side of the CGI interface from a series of interface functions bounded by a BEGIN FIGURE and an END FIGURE. A closed figure consists of one or more *regions*. The *boundary* of the closed figure and of each region is constructed from explicit line, fill, and GDP *primitives* and *implicit boundary portions* added as necessary to ensure the boundary remains closed. Edge *attribute values* are *associated* with individual primitives used to construct the closed figure; these are *local attributes values*. (Interior) fill attribute values are associated with the fill object as a whole; these are *global attributes values*.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9636-1:1991
https://standards.iteh.ai/catalog/standards/sist/d8e145a5-318e-4570-bbee-a06129749cd5/iso-iec-9636-1-1991

3.2.34 closure point
In *closed figure* and POLYGON SET construction, when the definition of the *boundary* of a *region* has been started but is not yet complete, the first point specified for the boundary of the region is referred to as the current closure point. The occurrence of an edge flag of CLOSE VISIBLE or CLOSE INVISIBLE in POLYGON SET or the invocation of a fill primitive, NEW REGION, or END FIGURE function while in state FIGURE OPEN (for closed figures) causes the automatic creation of a line segment constituting a portion of the boundary from the last line primitive end point to the current closure point. This line segment is either an *implicit boundary portion* or an edge portion, depending on whether CONNECTING EDGE precedes the closure.

3.2.35 colour selection mode
A mode determining whether colour specification is DIRECT by means of a *(direct) colour value* or INDEXED by means of an index value into a *colour table*. The colour values in both DIRECT colour specification and appearing as entries in the colour table are specified using an RGB additive colour model. Colour selection mode is independent of the setting of any colour *ASFs*.

3.2.36 colour table
A table of *(direct) colour values* used to map from an index value to a direct colour value when the *colour selection mode* is INDEXED.

3.2.37 (direct) colour value
A 3-tuple specifying the relative amounts of red, green, and blue light which must be combined to achieve the desired colour stimulus. See *colour selection mode*.

3.2.38 compound object
A generic term for both *compound text* and *closed figures*.

3.2.39 compound text
An *object* constructed on the device side of the CGI interface from a series of interface functions bounded by a TEXT or RESTRICTED TEXT with a not-final/final flag value of NOT FINAL and an APPEND TEXT with a flag value of FINAL. (Such construction might also be started, continued, or completed by *GDPs*.) The text attributes are subdivided into those