INTERNATIONAL STANDARD

ISO/IEC 13719-1

Second edition
1998-10-01

Information technology — Portable Common Tool Environment (PCTE) —

Part 1:
Abstract specification

Technologies de l'information — Environnement d'outil courant portable (PCTE) —
Partie 1: Spécifications abstraites

Reference number
ISO/IEC 13719-1:1998(E)

Contents

1 Scope 1

2 Conformance 1

2.1 Conformance of binding 1
2.2 Conformance of implementation 2
2.3 Conformance of DDL texts and processors 3

3 Normative references 3

4 Definitions 4

4.1 Technical terms 4
4.2 Other terms 4

5 Formal notations 5

6 Overview of PCTE 5

6.1 PCTE structural architecture 5
6.2 Object management system 6
6.3 Object base 6
6.4 Schema management 6
6.5 Self-representation and predefined SDSs 7
6.6 Object contents 7
6.7 Process execution 8
6.8 Monitoring 8
6.9 Communication between processes 8
6.10 Notification 8
6.11 Concurrency and integrity control 8
6.12 Distribution 9
6.13 Replication 9
6.14 Security 10
6.15 Accounting 10
6.16 Implementation limits 10
6.17 Support of fine-grain objects 11
6.18 Support of object-orientation 11

7 Outline of the Standard 11

8 Foundation 13

8.1 The state 13
8.2 The object base 14

8.2.1 Objects 14
8.2.2 Attributes 15
8.2.3 Links 16

8.3 Types 17

8.3.1 Object types 17
8.3.2 Attribute types 18
8.3.3 Link types 19
8.3.4 Enumeral types 23

8.4 Types in SDS 23

8.4.1 Object types in SDS 25
8.4.2 Attribute types in SDS 25
8.4.3 Link types in SDS 26
8.4.4 Enumeral types in SDS 26

8.5 Types in working schema 26

8.5.1 Object types in working schema 27
8.5.2 Attribute types in working schema 28
8.5.3 Link types in working schema 28
8.5.4 Enumeral types in working schema 28

8.6 Types in global schema 28

8.7 Operations 29

8.7.1 Calling process 29
8.7.2 Direct and indirect effects 29
8.7.3 Errors 32
8.7.4 Operation serializability 33

9 Object management 33

9.1 Object management concepts 33

9.1.1 The basic type "object" 33
9.1.2 The common root 37
9.1.3 Datatypes for object management 37

9.2 Link operations 37
9.3 Object operations 47
9.4 Version operations 61

10 Schema management 69

10.1 Schema management concepts 69

10.1.1 Schema definition sets and the SDS directory 69
10.1.2 Types 70
10.1.3 Object types 71
10.1.4 Attribute types 72

10.1.5 Link types **73**
10.1.6 Enumeral types **74**
10.1.7 Datatypes for schema management **75**
10.2 SDS update operations **75**
10.3 SDS usage operations **105**
10.4 Working schema operations **112**

11 Volumes, devices, and archives **117**

11.1 Volume, device, and archiving concepts **117**
11.1.1 Volumes **117**
11.1.2 Administration volumes **118**
11.1.3 Devices **118**
11.1.4 Archives **119**
11.2 Volume, device, and archive operations **120**

12 Files, pipes, and devices **128**

12.1 File, pipe, and device concepts **128**
12.2 File, pipe, and device operations **132**

13 Process execution **140**

13.1 Process execution concepts **140**
13.1.1 Static contexts **140**
13.1.2 Foreign execution images **141**
13.1.3 Execution classes **141**
13.1.4 Processes **142**
13.1.5 Initial processes **149**
13.1.6 Profiling and monitoring concepts **150**
13.2 Process execution operations **150**
13.3 Security operations **166**
13.4 Profiling operations **171**
13.5 Monitoring operations **172**

14 Message queues **174**

14.1 Message queue concepts **174**
14.2 Message queue operations **177**

15 Notification **183**

15.1 Notification concepts **183**
15.1.1 Access events and notifiers **183**
15.1.2 Notification messages **184**
15.1.3 Time of sending notification messages **185**
15.1.4 Range of concerned message queues **185**
15.2 Notification operations **185**

16 Concurrency and integrity control 187

16.1 Concurrency and integrity control concepts 187

16.1.1 Activities 187
16.1.2 Resources and locks 190
16.1.3 Lock modes 192
16.1.4 Inheritance of locks 194
16.1.5 Establishment and promotion of locks 195
16.1.6 Implied locks 196
16.1.7 Conditions for establishment or promotion of a lock 197
16.1.8 Releasing locks 198
16.1.9 Permanence of updates 199
16.1.10 Tables for locks 200

16.2 Concurrency and integrity control operations 202

17 Replication 208

17.1 Replication concepts 208

17.1.1 Replica sets 208
17.1.2 Replicated objects 209
17.1.3 Selection of an appropriate replica 210
17.1.4 Administration replica set 211

17.2 Replication operations 212

18 Network connection 218

18.1 Network connection concepts 218

18.1.1 Execution sites 218
18.1.2 Workstations 219
18.1.3 Foreign systems 22
18.1.4 Network partitions 222
18.1.5 Accessibility 223
18.1.6 Workstation closedown 225

18.2 Network connection operations 226
18.3 Foreign system operations 231
18.4 Time operations 233

19 Discretionary security 234

19.1 Discretionary security concepts 234

19.1.1 Security groups 234
19.1.2 Access control lists 238
19.1.3 Discretionary access modes 241
19.1.4 Access control lists on object creation 243

19.2 Operations for discretionary access control operation 244
19.3 Discretionary security administration operations 248

20 Mandatory security 253

20.1 Mandatory security concepts 253

20.1.1 Mandatory classes 253
20.1.2 The mandatory class structure 255
20.1.3 Labels and the concept of dominance 256
20.1.4 Mandatory rules for information flow 258
20.1.5 Multi-level security labels 261
20.1.6 Floating security levels 264
20.1.7 Implementation restrictions 266
20.1.8 Built-in policy aspects 266

20.2 Operations for mandatory security operation 268
20.3 Mandatory security administration operations 274
20.4 Mandatory security operations for processes 279

21 Auditing 281

21.1 Auditing concepts 281

21.1.1 Audit files 281
21.1.2 Audit selection criteria 283

21.2 Auditing operations 284

22 Accounting 289

22.1 Accounting concepts 289

22.1.1 Consumers and accountable resources 289
22.1.2 Accounting logs and accounting records 290

22.2 Accounting administration operations 294
22.3 Consumer identity operations 299

23 Common binding features 300

23.1 Mapping of types 300

23.1.1 Mapping of predefined PCTE datatypes 300
23.1.2 Mapping of designators and nominators 302
23.1.3 Mapping of other values 310

23.2 Object reference operations 311
23.3 Link reference operations 314
23.4 Type reference operations 317

24 Implementation limits 320

24.1 Bounds on installation-wide limits 320
24.2 Bounds on workstation-dependent limits 321

24.3 Limit operations 322

24.3.1 Datatypes for limit operations 322

Annex A - VDM Specification Language for the Abstract Specification 323

Annex B - The Data Definition Language (DDL) 329

Annex C - Specification of Errors 339

Annex D - Auditable Events 362

Annex E - The Predefined Schema Definition Sets 370

Annex F - The fine-grain objects module 387

Annex G - The object-orientation module 400

Index of Operations 425

Index of Error Conditions 431

Index of Technical Terms 439

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 13719-1 was prepared by ECMA (as Standard ECMA-149) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

This second edition cancels and replaces the first edition (ISO/IEC 13719-1:1995), which has been technically revised.

ISO/IEC 13719 consists of the following parts, under the general title *Information technology - Portable Common Tool Environment (PCTE):*

- *Part 1: Abstract specification*
- *Part 2: C programming language binding*
- *Part 3: Ada programming language binding*
- *Part 4: IDL binding (Interface Definition Language)*

Annexes A to D and annexes F and G form an integral part of this part of ISO/IEC 13719. Annex E is for information only.

Information technology — Portable Common Tool Environment (PCTE) —

Part 1:
Abstract specification

1 Scope

This part of ISO/IEC 13719 specifies PCTE in abstract, programming-language-independent, terms. It specifies the interface supported by any conforming implementation as a set of abstract operation specifications, together with the types of their parameters and results. It is supported by a number of standard *bindings*, i.e. representations of the interface in standard programming languages.

The scope of this part of ISO/IEC 13719 is restricted to a single PCTE installation. It does not specify the means of communication between PCTE installations, nor between a PCTE installation and another system.

A number of features are not completely defined in this part of ISO/IEC 13719, some freedom being allowed to the implementor. Some of these are *implementation limits*, for which constraints are defined (see clause 24). The other implementation-dependent and implementation-defined features are specified in the appropriate places in this Standard.

PCTE is an interface to a set of facilities that forms the basis for constructing environments supporting systems engineering projects. These facilities are designed particularly to provide an infrastructure for programs which may be part of such environments. Such programs, which are used as aids to systems development, are often referred to as tools.

This part of ISO/IEC 13719 also includes (in annex B) a language standard for the PCTE Data Description Language (DDL), suitable for writing PCTE schema definition sets.

2 Conformance

2.1 Conformance of binding

A binding conforms to this part of ISO/IEC 13719 if and only if:

- it consists of a set of operational interfaces and datatypes, with a mapping from the operations and datatypes of this part of ISO/IEC 13719;
- each operation of this part of ISO/IEC 13719 is mapped to one or more sequences of one or more operations of the binding (distinct operations need not be mapped to distinct sets of sequences of binding operations);
- each datatype of this part of ISO/IEC 13719 is mapped to one or more datatypes of the binding;
- each named error of this part of ISO/IEC 13719 is mapped to one or more error values (status values, exceptions, or the like) of the binding;
- the conditions of clause 23 on common binding features are satisfied;

- the conditions for conformance of an implementation to the binding are defined, are achievable, and are not in conflict with the conditions in 2.2 below.

2.2 Conformance of implementation

The functionality of PCTE is divided into the following modules:

- The core module consists of the datatypes and operations defined in clauses 8 to 19 (except 13.1.6, 13.4, and 13.5) and 23.
- The mandatory access control module consists of the datatypes and operations defined in clause 20.
- The auditing module consists of the datatypes and operations defined in clause 21.
- The accounting module consists of the datatypes and operations defined in clause 22.
- The profiling module consists of the datatypes defined in 13.1.6 and the operations defined in 13.4.
- The monitoring module consists of the datatype Address defined in 13.1.6 and operations defined in 13.5.
- The fine-grain objects module consists of the following extensions defined in annex F:
  - . extensions to the semantics of operations to cater for fine-grain objects;
  - . new operations;
  - . new error conditions;
  - . additions to the predefined SDS system.
- The object-orientation module consists of the following extensions defined in annex G:
  - . additions to the predefined SDSs metasds and system;
  - . an extension to the semantics of the operation SDS_REMOVE_TYPE to cater for the new classes of type;
  - . new operations;
  - . new error conditions.

An implementation of PCTE conforms to this part of ISO/IEC 13719 if and only if it implements the core module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with mandatory access control level 1 or 2 if it implements the core module and in addition:

- for level 1: the mandatory access control module except the floating security levels features defined in 20.1.6;
- for level 2: the mandatory access control module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with auditing if and only if it implements the core module and in addition the auditing module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with accounting if and only if it implements the core module and in addition the accounting module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with profiling if and only if it implements the core module and in addition the profiling module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with monitoring if and only if it implements the core module and in addition the monitoring module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with fine-grain objects if and only if it implements the core module and in addition, implements the fine-grain objects module.

An implementation of PCTE conforms to this part of ISO/IEC 13719 with object-orientation if and only if it implements the core module and in addition the object-orientation module.

By 'an implementation implements a module' is meant that, for the clauses of the module:

- the implementation conforms to a binding of this part of ISO/IEC 13719 which itself conforms to this part of ISO/IEC 13719 and which is itself an International Standard;
- if an operation of this part of ISO/IEC 13719 is mapped to a set of sequences of operations in the binding:
  - . case 1: operation_A; operation_B; .. operation_F;
  - . case 2: operation_G; operation_H; ...operation_M;
  - . etc.

  then in each case the sequence of invocations of the operations of the implementation must have the effect of the original operation of this part of ISO/IEC 13719;
- the relevant limits on quantities specified in clause 24 are no more restrictive than the values specified there;
- the implementations of the implementation-defined features in this part of ISO/IEC 13719 are all defined.

An implementation of PCTE does not conform to this part of ISO/IEC 13719 if it implements any of the following, whether or not the PCTE entity mentioned is in a module which the implementation implements:

- an operation with same name as a PCTE operation but with different effect;
- an SDS with the same name as a PCTE predefined SDS but with different contents;
- an error condition with the same name as a PCTE error condition but with different meaning.

2.3 Conformance of DDL texts and processors

A DDL definition conforms to this part of ISO/IEC 13719 if it conforms to the syntax and obeys the constraints of the DDL definition in annex B.

A DDL processor conforms to this part of ISO/IEC 13719 if it accepts any conforming DDL definition and processes it in conformance with the meaning of DDL as defined in annex B.

3 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 13719. 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 13719 are encouraged to investigate the possibility of applying the most recent editions

of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 2022:1994, *Information technology - Character code structure and extension techniques.*

ISO 8601:1988, *Data elements and interchange formats - Information interchange - Representation of dates and times.*

ISO/IEC 8859-1:1998, *Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1.*

ISO/IEC 10646-1:1993, *Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane.*

ISO/IEC 11404:1996, *Information technology - Programming languages, their environments and system software interfaces - Language-independent datatypes.*

ISO/IEC 13817-1:1996, *Information technology - Programming languages, their environments and system software interfaces - Vienna Development Method - Specification Language - Part 1: Basic language.*

ISO/IEC 14977:1996, *Information technology - Syntactic metalanguage - Extended BNF.*

4 Definitions

4.1 Technical terms

All technical terms used in this part of ISO/IEC 13719, other than a few in widespread use, are defined in the text, usually in a formal notation. All identifiers defined in VDM-SL or in DDL (see clause 5) are technical terms; apart from those, a defined technical term is printed in italics at the point of its definition, and only there. For the use of technical terms defined in VDM-SL and DDL see clause A.3 and clause B.9 respectively. All defined technical terms are listed in an index, with references to their definitions.

4.2 Other terms

For the purposes of this part of ISO/IEC 13719, the following definitions apply.

4.2.1 implementation-defined: Possibly differing between PCTE implementations, but defined for any particular PCTE implementation.

4.2.2 implementation-dependent: Possibly differing between PCTE implementations and not necessarily defined for any particular PCTE implementation.

4.2.3 binding-defined: Possibly differing between language bindings, but defined for any particular language binding.

4.2.4 datatype: The type of a parameter or result of an operation defined in this part of ISO/IEC 13719, or used to define such a type. Where, as in clause 23, it is necessary to distinguish these types from datatypes defined elsewhere, the term *PCTE datatype* is used.

4.2.5 operation: a name plus a signature that is used in the context of an invocation to trigger the execution of a specific method.

4.2.6 interface: a set of operations; interfaces are a convenient way to group operations so that they can be referred to together, e.g. to define other interfaces by inheritance.

4.2.7 method: the set of actions triggered by an operation.

5 Formal notations

Four formal notations are used in this part of ISO/IEC 13719.

For datatypes and for operation signatures, a small subset of the *Vienna Development Method Specification Language* or *VDM-SL* is used; it is defined in annex A. This subset of VDM-SL is also used to define some types used for operation parameters and results.

The *Data Definition Language* or *DDL* is used to define types; it is defined in annex B. Where a concept is defined in both VDM-SL and DDL, the same identifier is used.

To define the error conditions detected by operations, a parameterized notation is used; it is defined in annex C.

The BSI syntactic notation (BS 6154 : 1981) is used to define the syntax of VDM-SL and DDL, and in a few other places where the syntax of strings is defined.

6 Overview of PCTE

PCTE is designed to support program portability by providing machine-independent access to a set of facilities. These facilities, which are described in this part of ISO/IEC 13719, are designed particularly to provide an infrastructure for programs to support systems engineering projects.

The PCTE architecture is described in two dimensions: the *structural architecture* and the *functional architecture*. The structural architecture is described in 6.1, and shows how a PCTE installation is built of a system of communicating workstations and how the software providing the PCTE interfaces is structured. The functional architecture is described in 6.2 onwards, and gives an outline of the functional components of PCTE and the facilities they provide.

6.1 PCTE structural architecture

The preferred structural architecture for a PCTE installation is a set of workstations and associated resources communicating over a network, though other architectures are possible. There is no hierarchy or ordering of workstations within a PCTE installation. If a workstation is part of a PCTE installation then the PCTE installation appears to the workstation's user as a conceptually single machine, although each workstation can act as an autonomous unit. Such a user has access to the total resources of a PCTE installation, subject to the necessary access controls.

The PCTE database (called the *object base*) is partitioned into volumes. Volumes are dynamically allocated to (*mounted on*) particular workstations, and, once mounted, are globally available in that PCTE installation.

The program writer does not need to be aware of the distribution architecture, but the PCTE interfaces do provide all the facilities needed to configure a PCTE installation and control its distribution. The PCTE interfaces appear to the tool writer as available within a PCTE installation irrespective of the tool's physical location within a PCTE installation and independent of any particular network topology.

6.2 Object management system

An aspect of PCTE that is of major importance to the process of constructing and integrating portable tools is the provision of the object base and a set of functions to manipulate the various objects in the object base. The object base is the repository of the data used by the tools of a PCTE installation, and the *Object Management System* or *OMS* of PCTE provides the functions used to access the object base.

In a general sense, the users and programs of the PCTE installation have the ability to manage entities that are known to, and can be designated in, a particular PCTE installation. These may be files in the traditional sense, or peripherals, interprocess message queues or pipes, or the description of processes themselves or of the static context of a process. Tools supporting user applications establish classes of objects defined by the user: these can represent information items such as project milestones, tasks, and change requests.

6.3 Object base

The basic OMS model is derived from the Entity Relationship data model and defines *objects* and *links* as being the basic items of a PCTE object base.

Objects are entities (in the Entity Relationship sense) which can be designated, and can optionally have:

- *Contents*: a storage of data representing the traditional file concept;
- *Attributes*: primitive values representing specific properties of an object which can be named individually;
- *Links*: representations of associations between objects. Links may have attributes, which may be used to describe properties of the associations or as keys to distinguish between links of the same type from the same object.

Designation of links is the basis for the designation of objects: the principal means for accessing objects in most OMS operations is to navigate the object base by traversing a sequence of links.

6.4 Schema management

Entities used by the user and those used by the system that are represented by objects in the object base can be treated in a uniform manner, and facilities to control their structure, to store and to designate these objects, are provided by PCTE.

The object base of each PCTE installation is governed by a typing mechanism. All entities in the object base are typed and the data must conform to the corresponding type rules. Type rules are defined for objects, for links, and for attributes.

PCTE is designed to allow, but not to require, distributed and devolved management of the object base. To this end the definition of the typing rules which govern an object, a link, or an attribute in the object base may be split up among a number of *schema definition sets* (or *SDSs*). Some properties of an object, a link, or an attribute must be the same in every SDS which contributes to the definition of the typing rules for that object, link, or attribute: these are properties of the *type*. Other properties may differ for different SDSs: these are properties of the *type in SDS*.

Each SDS provides a consistent and self-contained view of the data in the object base. A process, at any one time, views the data in the object base through a *working schema*. A working schema is obtained as a composition of SDSs in an ordered list. The effect of such a composition is to provide a union of all the types contained in the listed SDSs. A uniform naming algorithm, dependent on the ordering of the SDSs, is applied to all the contained types.

The object base of a PCTE installation has a notional *global schema*, composed of all the SDSs. The global schema is not directly represented in the object base, and the concept is used mainly to state certain consistency constraints on the object base as a whole.

Child types of object types can be defined with the effect of implicit inheritance of all properties of their parent types. Additionally, child types can have properties of their own.

6.5 Self-representation and predefined SDSs

Many of the entities in a PCTE installation are represented by objects in the object base. The types of these objects are defined in *predefined SDSs*, which are available in any conforming implementation; for example processes are represented by objects of type "process" which is defined in the predefined SDS 'system'. This property of PCTE is called *self-representation*. In general, in this part of ISO/IEC 13719, the name of an entity is used also to refer to the object that represents it.

In some cases an object of a type representing some kind of entity requires initializing, or must be created by a particular operation, before it can be used in operations to represent an entity of that kind. Such an object which has been initialized or correctly created is referred to as a *known* entity of that kind (i.e. known to the PCTE installation); any other object of that type is referred to as an *unknown* entity. For example an object of type "process" created by PROCESS_CREATE is a known process, while one created by OBJECT_CREATE is an unknown process.

6.6 Object contents

A set of operations is provided to access the contents of some types of objects (files, pipes, and devices). These operations provide conventional input-output facilities on files and pipes and control of input and output on devices. These contents are not interpreted by PCTE.

Other types of objects (accounting logs and audit files) have contents with structure that is defined by PCTE and for access to which special operations are provided.