

INTERNATIONAL  
STANDARD

**ISO/IEC**  
**13719-1**

First edition  
1995-06-01

---

---

**Information technology — Portable  
Common Tool Environment (PCTE) —**

**Part 1:**  
Abstract specification

(standards.iteh.ai)

*Technologies de l'information — Environnement d'outil courant portable  
(PCTE)*  
<https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-fb125347366c/iso-iec-13719-1-1995>  
Partie 1: Spécification d'abstrait



Reference number  
ISO/IEC 13719-1:1995(E)

## Contents

<b>1</b>	<b>Scope</b>	<b>1</b>
<b>2</b>	<b>Conformance</b>	<b>1</b>
2.1	Conformance of binding	1
2.2	Conformance of implementation	2
<b>3</b>	<b>Normative references</b>	<b>3</b>
<b>4</b>	<b>Definitions</b>	<b>3</b>
4.1	Technical terms	3
4.2	Other terms	4
<b>5</b>	<b>Formal notations</b>	<b>4</b>
	<a href="https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995">ISO/IEC 13719-1:1995 https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667- f41b253cf79a/iso-iec-13719-1-1995</a>	
<b>6</b>	<b>Overview of PCTE</b>	<b>4</b>
6.1	PCTE structural architecture	5
6.2	Object management system	5
6.3	Object base	5
6.4	Schema management	6
6.5	Self-representation and predefined SDSs	6
6.6	Object contents	6
6.7	Process execution	7
6.8	Monitoring	7
6.9	Communication between processes	7
6.10	Notification	7
6.11	Concurrency and integrity control	7

© ISO/IEC 1995

All rights reserved. Unless otherwise specified, 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

6.12	Distribution	8
6.13	Replication	8
6.14	Security	8
6.15	Accounting	9
6.16	Implementation limits	9
<b>7</b>	<b>Outline of ISO/IEC 13719</b>	<b>9</b>
<b>8</b>	<b>Foundation</b>	<b>10</b>
8.1	The state	10
8.2	The object base	11
	8.2.1 Objects	11
	8.2.2 Attributes	12
	8.2.3 Links	13
8.3	Types	14
	8.3.1 Object types	14
	8.3.2 Attribute types	15
	8.3.3 Link types	16
	8.3.4 Enumeral types	19
8.4	Types in SDS	19
	8.4.1 Object types in SDS	21
	8.4.2 Attribute types in SDS	21
	8.4.3 Link types in SDS	22
	8.4.4 Enumeral types in SDS	22
8.5	Types in working schema	22
	8.5.1 Object types in working schema	23
	8.5.2 Attribute types in working schema	23
	8.5.3 Link types in working schema	24
	8.5.4 Enumeral types in working schema	24
8.6	Types in global schema	24
8.7	Operations	25
	8.7.1 Calling process	25
	8.7.2 Direct and indirect effects	25
	8.7.3 Errors	27
	8.7.4 Operation serializability	28
<b>9</b>	<b>Object management</b>	<b>28</b>
9.1	Object management concepts	28
	9.1.1 The basic type "object"	28
	9.1.2 The common root	31
	9.1.3 Datatypes for object management	32
9.2	Link operations	32
9.3	Object operations	40
9.4	Version operations	53

<b>10</b>	<b>Schema management</b>	<b>59</b>
10.1	Schema management concepts	59
10.1.1	Schema definition sets and the SDS directory	59
10.1.2	Types	60
10.1.3	Object types	61
10.1.4	Attribute types	61
10.1.5	Link types	63
10.1.6	Enumeral types	64
10.1.7	Datatypes for schema management	64
10.2	SDS update operations	64
10.3	SDS usage operations	89
10.4	Working schema operations	96
<b>11</b>	<b>Volumes, devices, and archives</b>	<b>100</b>
11.1	Volume, device, and archiving concepts	100
11.1.1	Volumes	100
11.1.2	Administration volumes	101
11.1.3	Devices	102
11.1.4	Archives	102
11.2	Volume, device, and archive operations	103
<b>12</b>	<b>Files, pipes, and devices</b>	<b>110</b>
12.1	File, pipe, and device concepts	110
12.2	File, pipe, and device operations	113
<b>13</b>	<b>Process execution</b>	<b>120</b>
13.1	Process execution concepts	120
13.1.1	Static contexts	120
13.1.2	Foreign execution images	121
13.1.3	Execution classes	121
13.1.4	Processes	122
13.1.5	Initial processes	128
13.1.6	Profiling and monitoring concepts	129
13.2	Process execution operations	129
13.3	Security operations	141
13.4	Profiling operations	146
13.5	Monitoring operations	147
<b>14</b>	<b>Message queues</b>	<b>149</b>
14.1	Message queue concepts	149
14.2	Message queue operations	151

iTech STANDARD PREVIEW  
(standards.iteh.ai)

<https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-419223cf79a/iso-iec-13719-1-1995>

<b>15</b>	<b>Notification</b>	<b>157</b>
15.1	Notification concepts	157
15.1.1	Access events and notifiers	157
15.1.2	Notification messages	158
15.1.3	Time of sending notification messages	158
15.1.4	Range of concerned message queues	158
15.2	Notification operations	159
<b>16</b>	<b>Concurrency and integrity control</b>	<b>160</b>
16.1	Concurrency and integrity control concepts	160
16.1.1	Activities	160
16.1.2	Resources and locks	162
16.1.3	Lock modes	164
16.1.4	Inheritance of locks	166
16.1.5	Establishment and promotion of locks	167
16.1.6	Implied locks	168
16.1.7	Conditions for establishment or promotion of a lock	169
16.1.8	Releasing locks	169
16.1.9	Permanence of updates	170
16.1.10	Tables for locks	170
16.2	Concurrency and integrity control operations	173
<b>17</b>	<b>Replication</b>	<b>178</b>
17.1	Replication concepts	178
17.1.1	Replica sets	178
17.1.2	Replicated objects	179
17.1.3	Selection of an appropriate replica	180
17.1.4	Administration replica set	181
17.2	Replication operations	181
<b>18</b>	<b>Network connection</b>	<b>187</b>
18.1	Network connection concepts	187
18.1.1	Execution sites	187
18.1.2	Workstations	187
18.1.3	Foreign systems	190
18.1.4	Network partitions	190
18.1.5	Accessibility	191
18.1.6	Workstation closedown	192
18.2	Network connection operations	193
18.3	Foreign system operations	198
18.4	Time operations	199
<b>19</b>	<b>Discretionary security</b>	<b>200</b>
19.1	Discretionary security concepts	200
19.1.1	Security groups	200
19.1.2	Access control lists	203

19.1.3	Discretionary access modes	206
19.1.4	Access control lists on object creation	208
19.2	Operations for discretionary access control operation	209
19.3	Discretionary security administration operations	212
<b>20</b>	<b>Mandatory security</b>	<b>217</b>
20.1	Mandatory security concepts	217
20.1.1	Mandatory classes	217
20.1.2	The mandatory class structure	218
20.1.3	Labels and the concept of dominance	219
20.1.4	Mandatory rules for information flow	220
20.1.5	Multi-level security labels	224
20.1.6	Floating security levels	226
20.1.7	Implementation restrictions	228
20.1.8	Built-in policy aspects	228
20.2	Operations for mandatory security operation	230
20.3	Mandatory security administration operations	235
20.4	Mandatory security operations for processes	239
<b>21</b>	<b>Auditing</b>	<b>241</b>
21.1	Auditing concepts	241
21.1.1	Audit files	241
21.1.2	Audit selection criteria	243
21.2	Auditing operations	244
	<a href="https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995">standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995</a>	
<b>22</b>	<b>Accounting</b>	<b>248</b>
22.1	Accounting concepts	248
22.1.1	Consumers and accountable resources	248
22.1.2	Accounting logs and accounting records	249
22.2	Accounting administration operations	252
22.3	Consumer identity operations	257
<b>23</b>	<b>Common binding features</b>	<b>257</b>
23.1	Mapping of types	257
23.1.1	Mapping of predefined PCTE datatypes	257
23.1.2	Mapping of designators and nominators	260
23.1.3	Mapping of other values	266
23.2	Object reference operations	267
23.3	Link reference operations	269
23.4	Type reference operations	272
<b>24</b>	<b>Implementation limits</b>	<b>274</b>
24.1	Bounds on installation-wide limits	274

24.2	Bounds on workstation-dependent limits	275
<b>Annex A</b>	<b>(normative) VDM Specification Language for the abstract specification</b>	<b>278</b>
<b>Annex B</b>	<b>(normative) The Data Definition Language (DDL)</b>	<b>283</b>
<b>Annex C</b>	<b>(normative) Specification of errors</b>	<b>292</b>
<b>Annex D</b>	<b>(normative) Auditable events</b>	<b>311</b>
<b>Annex E</b>	<b>(informative) The predefined schema definition sets</b>	<b>318</b>
<b>Index of Error Conditions</b>		<b>337</b>
<b>Index of Technical Terms</b>		<b>344</b>

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

[ISO/IEC 13719-1:1995](https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995)

<https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995>

## 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 the European Computer Manufacturers Association (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.

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*

Annexes A to D form an integral part of this part of ISO/IEC. 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 part of ISO/IEC 13719.

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.

### 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.

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.

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.

### 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 International Standard 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,	<i>Information technology — Character code structure and extension techniques.</i>
ISO 8601 : 1988,	<i>Data elements and interchange formats — Information interchange — Representation of dates and times.</i>
ISO 8859-1 : 1987,	<i>Information processing — 8-bit single-byte coded graphic character sets — Part 1 : Latin alphabet No. 1.</i>
ISO/IEC 10646-1 : 1993,	<i>Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1 : Architecture and Basic Multilingual Plane.</i>
ISO/IEC 11404 : — <sup>1)</sup> ,	<i>Information technology — Programming languages, their environments and system software interfaces — Language-independent datatypes.</i>
ISO/IEC 13303-1 : — <sup>1)</sup> ,	<i>Information technology — Programming languages, their environments and system software interfaces — Vienna Development Method/Specification language — Part 1 : Basic Language.</i>
BS 6145 : 1981	<i>Method of Defining Syntactic Metalanguage.</i>

### 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 5.2) 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.

1) To be published.

## 4.2 Other terms

For the purposes of this International Standard, 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.

## 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 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.

## 6.7 Process execution

PCTE is an interface to support programs. When a program is *run*, this is either the *execution* of the program itself, or the execution of an interpreter which interprets the program. An execution of a program is a *process*. Processes are represented by objects in the object base, so the hierarchy of processes, the environment in which a process runs, the parameters it has been passed, and the various stages of the program execution can be controlled, manipulated and examined.

These facilities can be used also to control processes running on *foreign systems*. A *foreign system* can be a foreign development system, a target system running a real-time operating system, or even a PCTE workstation in another PCTE installation.

## 6.8 Monitoring

PCTE provides three sets of features to support debugging and monitoring of processes.

- To measure the amount of time spent in selected parts of the code.
- To observe, and modify, the execution of a child process.
- To measure the processor usage of the calling process.

## 6.9 Communication between processes

PCTE provides a number of different mechanisms for communicating between processes. The principal ones supplied are:

- the objects, links and attributes in the database;
- message queues; [ISO/IEC 13719-1:1995](https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995)
- pipes. <https://standards.iteh.ai/catalog/standards/sist/03dd9a72-dd30-4a53-8667-f41b253cf79a/iso-iec-13719-1-1995>

Message queues and pipes are essentially special forms of object. Thus both pipes and message queues are special cases of the general use of the object base for *interprocess communication*.

Pipes and message queues also provide communication between PCTE processes and foreign processes running on foreign systems (if the foreign systems allow it).

## 6.10 Notification

In PCTE there is a mechanism that allows the designation of objects so that certain types of access result in a message being posted in a message queue which can be accessed by the process requesting the notification.

The notification mechanism allows a process to specify events, corresponding to operations on objects, of which it wants to be notified.

## 6.11 Concurrency and integrity control

The object base is subject to concurrent access by users, and is liable to underlying system failure.

PCTE provides locking facilities to control the strength of object base concurrency and consistency, ranging from unprotected behaviour, through protected behaviour, to protected atomic and serializable transaction activities. PCTE ensures object base consistency and object base integrity for atomic and serializable transactions.

Each user carrying out a transaction on the object base sees some grouping of operations as an atomic operation which transforms the object base from one consistent state to another. If transactions are run one at a time then each transaction sees the consistent state left by its