Standard ECMA-228
October 1995

ECMA
Standardizing Information and Communication Systems

Portable Common Tool Environment (PCTE) - Extensions for support of Fine-Grain Objects - C Programming Language Binding

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Standard ECMA-228
October 1995

ECMA

Standardizing Information and Communication Systems

Portable Common Tool Environment (PCTE) - Extensions for support of Fine-Grain Objects - C Programming Language Binding

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Gino Lauri - ECMA-228.DOC - 05.12.95 09,31

Brief History

Software engineering tools are increasingly manipulating large amounts of objects accessed by several application developers in the context of integrated software development environments. With PCTE, defined in Standard ECMA-149, the software community has all the basic functionalities required to develop such repositories. In early 1993, however, it soon appeared that not all objects manipulated by software tools need to be shared with the same level of flexibility but, on another hand, very often require performances which seem hard to achieve with all properties associated with PCTE objects in general. Typically, a given tool may need to manipulate a large set of objects which are most often used by this tool only at a given time (therefore allowing simplified concurrent access mechanisms), with very short access times.

The abstract specification of the PCTE extensions for the support of fine-grain objects has its origin in a joint project of the North American PCTE Initiative (later the Object Management Group PCTE Special Interest Group) and ECMA TC33, later joined by ISO/IEC JTC1/SC22/WG22 - PCTE. This ECMA Standard is the result of a collaborative effort by all these bodies.

This ECMA Standard has been adopted by the ECMA General Assembly in October 1995.

Table of contents

1 Scope

(1) This document defines the standard binding of the Portable Common Tool Environment (PCTE) extensions for the support of fine-grain objects, as specified in Standard ECMA-227, to the C programming language. It forms an extension to Standard ECMA-158.

2 Conformance

(1) An implementation of PCTE conforms to this Standard if it conforms to both ECMA-158 and to ECMA-227, as defined in 2.2 of that Standard, where the binding referred is taken to be the C binding defined in clauses 1 to 5 and 8 to 10 of this Standard.

(2) The C language binding defined in this Standard conforms to ECMA-227, as defined in 2.1 of that Standard.

3 Normative references

(1) The following Standards contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the Standards indicated below.

(2)	ECMA-149	Portable Common Tool Environment (PCTE) - Abstract Specification (3rd edition, December 1994)
(3)	ECMA-158	Portable Common Tool Environment (PCTE) - C Programming Language Binding (3rd edition, December 1994)
(4)	ECMA-227	Portable Common Tool Environment (PCTE) - Extensions for support of Fine-Grain Objects - Abstract Specification (October 1995)
(5)	ISO/IEC 9899:1990	Information technology - Programming languages, their environments and system software interfaces - C programming language

4 Definitions

(1) All technical terms used in this Standard, other than a few in widespread use, are defined in the body of this Standard or in ECMA-149, ECMA-158, ECMA-227, or ISO/IEC 9899.

5 Formal notations

(1) The notations used in this Standard are the same as those used in ECMA-158.

6 Outline of the Standard

(1) Clause 7 describes the strategy used to develop this binding specification.

(2) Clause 8 contains the mapping of datatypes used in ECMA-227.

(3) Clause 9 defines the binding of new operations defined in ECMA-227.

(4) Clause 10 defines the binding of the new error conditions specified in annex C of ECMA-227.

7 Binding strategy

(1) The binding strategy used in this Standard is the same as that used in ECMA-158.

8 Datatype mapping

(1) The datatype mapping used in this Standard is the one used in ECMA-158. The support of fine-grain objects does not require new data types.

(2) An implementation of this Standard must extend the global header file <Pcte/pcte.h> as follows:

```
(3)     /* The header   <Pcte/pcte.h> */

(4)     #ifndef          PCTE_INCLUDED
        #define                     PCTE_INCLUDED          1

(5)     /* All include files of article 8.7.1 of ECMA-158 are included here */

        #include         <Pcte/types.h>
        #include         <Pcte/sequences.h>
        ...
        ...
        #include         <Pcte/accounting.h>

(6)     /* include file used for cluster management */

        #include         <Pcte/clusters.h>

(7)     #endif
```

9 New operations on clusters

```
(1)     /* The header <Pcte/clusters.h> */

(2)     #ifndef          PCTE_CLUSTERS_INCLUDED
        #define                     PCTE_CLUSTERS_INCLUDED   1

(3)     #include         <Pcte/types.h>
        #include         <Pcte/references.h>
        #include         <Pcte/sequences.h>
        #include         <Pcte/security.h>

        /* CLUSTER_CREATE : see 11.3.1 of ECMA-227 */

(4)     int Pcte_cluster_create (
                  Pcte_object_reference           volume,
                  Pcte_natural                    cluster_id,
                  Pcte_atomic_access_rights       *access_mask,
                  Pcte_string                     *cluster_characteristics,
                  Pcte_object_reference           *new_cluster
        );

        /* CLUSTER_DELETE : see 11.3.2 of ECMA-227 */

(5)     int Pcte_volume_delete (
                        Pcte_object_reference           cluster
        );

        /* CLUSTER_LIST_OBJECTS : see 11.3.3 of ECMA-227 */

(6)     int  Pcte_cluster_list_objects (
                  Pcte_object_reference           cluster,
                  Pcte_type_references            types,
                  Pcte_object_references          *objects
        );

(7)     #endif
```

10 New error conditions

(1) An implementation of this Standard must extend the header file <Pcte/errors.h> as follows:

```
(2)         /* The header <Pcte/errors.h> */

(3)         #ifndef  PCTE_ERRORS_INCLUDED
            #define          PCTE_ERRORS_INCLUDED        1

(4)         typedef enum
                    PCTE_NO_ERROR,

(5)                 /* All errors defined in 25.1 of ECMA-158 are here */

                    PCTE_ACCESS_MODE_IS_INCOMPATIBLE,
                    PCTE_ACCESS_MODE_IS_NOT_ALLOWED,
                    ...
                    ...
                    PCTE_VALUE_TYPE_IDENTIFIER_DOES_NOT_MATCH,

(6)                 /* New error conditions for fine-grain support */

                    PCTE_OBJECT_CANNOT_BE_CLUSTERED,
                    PCTE_OBJECT_IS_FINE_GRAIN,
                    PCTE_CLUSTER_EXISTS,
                    PCTE_CLUSTER_HAS_OTHER_LINKS,
                    PCTE_CLUSTER_IS_UNKNOWN

            Pcte_error_type;
```

This Standard ECMA-228 is available free of charge from:

ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

Fax: +41 22 849.60.01
Internet: helpdesk@ecma.ch

This Standard can also be downloaded as file E228-doc.exe and E228-psc.exe from FTP.ECMA.CH.