ECMA

Standardizing Information and Communication Systems

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

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

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 Ada programming language. It forms an extension to Standard ECMA-162.

2 Conformance

(1) An implementation of PCTE conforms to this Standard if it conforms to both ECMA-162 and to ECMA-227, as defined in 2.2 of that Standard, where the binding referred is taken to be the Ada binding defined in clauses 1 to 5 and 8 and 9 of this Standard.

(2) The Ada 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-162   Portable Common Tool Environment (PCTE) - Ada 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 8652: 1995   Information technology - Programming languages, their environments and system software interfaces - Ada programming language (referring to ANSI-MIL-STD 1815A - Reference manual for the Ada 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-162, ECMA-227, or ISO/IEC 8652.

5 Formal notations

(1) The notations used in this Standard are the same as those used in ECMA-162.

6 Outline of the Standard

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

(2) Clause 8 defines the bindings of the new operations in 11.3 of ECMA-227.

(3) Clause 9 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-162.

8 New operations on clusters

(1) **with** Pcte, Pcte_error, Pcte_discretionary;

(2) **package** Pcte_cluster **is**

(3) **use** Pcte, Pcte_error;

-- CLUSTER_CREATE - see 11.3.1 of ECMA-227

(4) **procedure** create (
```
on_same_volume_as        : in Pcte.object_reference;
cluster_identifier       : in Pcte.natural;
access_mask              : in Pcte_discretionary.object.atomic_access_rights;
cluster_characteristics  : in Pcte.string;
new_cluster              : in out Pcte.object_reference;
status                   : in Pcte_error.handle := EXCEPTION_ONLY);
```

-- CLUSTER_DELETE - see 11.3.2 of ECMA-227

(5) **procedure** delete (
```
cluster    : in Pcte.object_reference;
status     : in Pcte_error.handle := EXCEPTION_ONLY);
```

-- CLUSTER_LIST_OBJECTS - see 11.3.3 of ECMA-227

(6) **procedure** list_objects (
```
cluster    : in Pcte.object_reference;
types      : in Pcte.type_references.sequence
objects    : in out Pcte.object_references.sequence
status     : in Pcte_error.handle := EXCEPTION_ONLY);
```

(7) **end** Pcte_cluster;

9 New errors

(1) The following new values are added to the enumeration type Pcte_error.error_code defined in clause 25 of ECMA-162, with the code letters shown. For upward compatibility from ECMA-162 they are added at the end, after the Ada-binding-defined errors.

(2) OBJECT_CANNOT_BE_CLUSTERED   -- U

(3) OBJECT_IS_FINE_GRAIN   -- U

(4) CLUSTER_EXISTS   -- U

(5) CLUSTER_HAS_OTHER_LINKS   -- U

(6) CLUSTER_IS_UNKNOWN   -- I