INTERNATIONAL STANDARD

ISO/IEC 10728

First edition
1993-04-15

AMENDMENT 2
1996-11-15

Information technology — Information Resource Dictionary System (IRDS) Services Interface

AMENDMENT 2: Ada language binding

Technologies de l'information — Interface de services du gestionnaire de ressources du système d'informations (IRDS)

AMENDEMENT 2: Liant de langage Ada

Reference number
ISO/IEC 10728:1993/Amd.2:1996(E)

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.

Amendment 2 to International Standard ISO/IEC 10728:1993 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open systems interconnection, data management and open distributed processing*.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Information technology — Information Resource Dictionary System (IRDS) Services Interface

AMENDMENT 2: Ada language binding

Page v

Add a new entry to the Table of Contents as follows:

"Annex D - Ada language binding"

Page vi

Add a new sentence to the last paragraph in the Foreword as follows:

"Annex D is normative".

Page 1

Add a new sentence in clause 1 paragraph 2, before the last sentence.

"A language binding for the Ada language (ISO/IEC 8652) is provided in Annex D."

Add a reference in clause 2, after the reference to ISO 7185, as follows:

"ISO/IEC 8652:1995, *Information technology - Programming languages - Ada*."

Page 5

Add a new sentence in subclause 4.4.

"Data structures for use with the Ada language are defined in Annex D."

Add a new sentence in subclause 4.5.

"Ada language bindings for the services are provided in Annex D."

Page 71

Amend the first sentence of the NOTE in subclause 8.1 to read:

"For the Pascal language binding specified in this clause, the C language binding specified in Annex C, and the Ada language binding specified in Annex D, enumerated types are ... ."

Page 75

Add a new sentence at the end of clause 9 to read:

"Alternative service formats for use with the Ada language are specified in Annex D."

Add a new Annex D after Annex C (as amended by ISO/IEC 10728:1993/Amd.1:1995) as follows:

Annex D
(normative)
Ada language binding

The IRDS Services Interface language binding for the Ada language is presented in the form of an Ada package specification as provided in D.4 using the general rules as set out in D.2 below. D.3 provides a package called Hardware_Dependencies which isolates hardware, software, and system dependencies in order to improve portability of the binding.

D.1 Strategy for the Ada language binding

In this binding the data names and data structures defined in clause 8 have been adhered to except where the Ada language provides a preferred alternative construct. The binding defined in the package specification in D.4 provides an interface to the SQL types through a package that isolates hardware dependencies for Ada types. Time and date abstractions are obtained from package Calendar. It is expected that the package body will provide the necessary transformations to SQL and IRDS types.

In this binding, the procedure names and their parameters defined in clause 9 have been adhered to except as follows:

1. All names have been spelled out for understanding except for IRDS, IRD, and SQL.

2. All names have underscores between logical words for clarity.

3. All names of data structure types have the explicit word "type" added to the simple name for clarity. Excluded are Ada predefined types of Boolean, Positive, Character, String, and Time. Names ending in "type" which were not types have been renamed to avoid confusion.

4. The "Irds" prefix has been removed from all types, objects, and procedure names. The desired effect is achieved through the use of a package called "IRDS". The fully qualified name will thus include IRDS [e.g., IrdsCreateIRDDefinition becomes IRDS.Create_IRD_Definition, IrdsNameLim becomes IRDS.Name_Limit].

5. The subprogram names and parameter names in D.4 below shall have the same meaning as the mapped names defined in clause 9.

6. The use of the Service Return Code identified in subclause 8.2.5 is not used in preference to the use of Ada exceptions. A discussion on how exceptions are used is provided in the Ada binding discussion for subclause 8.2.5 within D.4 below.

7. Strings are passed as an access type (with a pointer) allowing unbounded lengths. A list structure is used for the Column List Parameters (subclause 8.2.6) again using access types. Access types end with the suffix "Pointer_Type."

8. Time abstractions are imported from package Calendar to facilitate time operations by Ada applications.

9. Use of name User_Id vice User as object of type User_Id_Type to conform with naming convention.

10. The following mappings for SQL data types to Ada data types have been used:

SQL Data Type	Ada Data Type
CHARACTER	Character_Type [Note 1]
CHARACTER VARYING	Text_Pointer_Type [Note 2]
NATIONAL CHARACTER	National_Character_Type [Note 1]
NATIONAL CHARACTER VARYING	National_Text_Pointer_Type [Note 2]
REAL	Real_Type [Note 1]
DOUBLE PRECISION	Long_Float_Type [Note 1]
FLOAT	Float_Type [Note 1]
INTEGER	Integer_Type [Note 1]
SMALL INTEGER	Short_Integer_Type [Note 1]
NUMERIC	Fixed_Type [Note 1]
DECIMAL	Decimal_Type [Note 1]
DATE	Calendar.Time [Note 3]
TIME	Calendar.Time [Note 3]
TIME STAMP	Calendar.Time [Note 3]
INTERVAL	Interval_Type [Note 4]

Notes

1. These types are imported from a separate package Hardware_Dependencies used to insulate the IRDS binding from hardware differences, thus improving portability of the application code. This package is provided in subclause D.3.

2. Text_Pointer_Type is defined in subclause 8.2.1 as an access type of String_Type. String_Type is an array of Character_Type. National_Text_Pointer_Type is defined in subclause 8.2.1 as an access type of National_String_Type. National_String_Type is an array of National_Character_Type.

3. Time abstractions are imported from the Ada package Calendar. The type Calendar.Time includes abstractions for year, month, day, and seconds. It is also useful as a date-time stamp.

4. This is a newly defined record; See Ada binding for subclause 8.2.1.

Use of this language binding requires the use of an Ada compiler that conforms to ISO/IEC 8652:1995. All validated Ada compilers that conform to ISO/IEC 8652:1995 meet this conformance requirement. A validated Ada compiler that conforms to ISO 8652:1987 only meets the requirement if it has the appropriate packages to support enhanced character sets.

D.2 General Rules

1. Those data names in D.4 below that map to data names in clause 8 shall have the same meanings as is defined in clause 8. Equivalent Ada Types are used to build data structures. It is expected that the package body of the IRDS interface will perform the necessary transformations to convert objects of Ada types to objects of SQL types and vice versa.

2. Several IRDS sessions can proceed in parallel using Ada tasking. Subclause 9.4 defines the sequence of permitted service invocations within a Current_Session_Id. The IRDS interface must be capable of serializing concurrent calls to its Application Program Interface (API). IRDS implementations should support the use of the language-defined library package Ada.Exceptions. This would provide IRDS applications with the capability to obtain auxiliary information regarding distinct exceptions when an error occurs.

3. An implementation shall provide a package Hardware_Dependencies to isolate the IRDS binding from hardware implementations of all Ada types except Boolean, Positive, Character, String, and Time. This approach provides a means to improve portability between IRDS implementations.

D.3 IRDS Ada Binding Hardware_Dependencies

```
--~--------------------------------------------------------------------------
--      Package Hardware_Dependencies
--
--      This version supports the Ada IRDS binding conforming to:
--              ISO/IEC 10728:1993(E)
--              IRDS Services Interface
--
--      Copyright ISO 1996
--~--------------------------------------------------------------------------

--~--------------------------------------------------------------------------
package Hardware_Dependencies is

-- Implementations are insulated from hardware dependencies by defining
-- objects and types which might vary from implementation to implementation in
-- the package Hardware_Dependencies. All known implementation dependent
-- interfaces are isolated herein.

-- This package contains:
--      1.  Values for constants set in clauses 8.1.1, 8.1.2, and 8.1.3.
--      2.  Specification of dependent types used in the IRDS Ada Binding.

--~--------------------------------------------------------------------------
--~--------------------------------------------------------------------------
--      1.  Values for constants set in clauses 8.1.1, 8.1.2, and 8.1.3.
--~--------------------------------------------------------------------------

        --~------------------------------------------------------------------
        --~------------------------------------------------------------------
        -- Subclause 8.1.1  Name Length Limits

        Name_Limit              : constant := implementation defined;
        Variation_Limit         : constant := implementation defined;

        --~------------------------------------------------------------------
        -- Subclause 8.1.2  Attribute Length Limits

        Text_Limit              : constant := implementation defined;

        -- Note that as access types are used, Text_Limit is not needed in
        -- this language binding. Knowledge of this constant may be useful.
```

```
          --~-----------------------------------------------------------------------
          -- Subclause 8.1.3  Control Identifier Length Limits

          Session_Id_Limit            : constant := implementation defined;
          Cursor_Id_Limit             : constant := implementation defined;

          Implementation_Dictionary_Name_Length
                                      : constant := implementation defined;

          Key_Length                  : constant := implementation defined;

--~-----------------------------------------------------------------------
--~-----------------------------------------------------------------------
--        2.  Specification of dependent types used in the IRDS Ada Binding.
--~-----------------------------------------------------------------------

-- The Ada binding should not result in a loss of precision. Implementations
-- have flexibility in choosing hardware representation based on requirements.
-- Interfaces to SQL databases using IRDS should consider use of the Ada
-- Package SQL_Standard defined in ISO/IEC 9075:1992.
-- The data types specified in this package represent the contract between the
-- types supported by IRDS and the types supported by the application program.
-- These types might vary based on hardware, system, or application
-- requirements. Ranges of these types should be explicitly declared based on
-- an understanding of the ranges supported by the chosen implementation of
-- IRDS.

          --~-----------------------------------------------------------------------
          -- IRDS character Types
          --~-----------------------------------------------------------------------

          -- The SQL standard supports an implementation-defined selection of a
          -- character type rather than simply mapping to Ada's character type.
          -- The definition of Character_Type is insulated in this package. This
          -- permits implementations to use Latin-1 (ISO 8859/1), Unicode (ISO
          -- 10646), or other character set.

          type Character_Type is implementation defined;
          type National_Character_Type is implementation defined;
          type String_Type is array (implementation defined) of Character_Type;
          type National_String_Type is
                array (implementation defined) of National_Character_Type;
```

```
      --~---------------------------------------------------------------------
      -- IRDS Integer Types
      --~---------------------------------------------------------------------

      -- The SQL standard supports 2 implementation defined integer types:

      type Integer_Type is implementation defined;
            -- Integer_Type must support range 0 .. 9999999. See subclause 8.2.1.

      type Short_Integer_Type is implementation defined;

      --~---------------------------------------------------------------------
      -- IRDS Float Types
      --~---------------------------------------------------------------------

      -- The SQL standard supports 3 implementation defined float types.

      type Float_Type is implementation defined;
      type Long_Float_Type is implementation defined;
      type Real_Type is implementation defined;

      --~---------------------------------------------------------------------
      -- IRDS Fixed Types
      --~---------------------------------------------------------------------

      -- The SQL standard supports 2 implementation defined fixed types:

      type Fixed_Type is implementation defined;
      type Decimal_Type is implementation defined;

end Hardware_Dependencies;
--~---------------------------------------------------------------------
```

D.4 IRDS Package Specification

```
-----------------------------------------------------------------------------
--      Package IRDS
--
--      This version conforms to:
--              ISO/IEC 10728:1993(E)
--              IRDS Services Interface
--
--      Copyright ISO 1996
-----------------------------------------------------------------------------

with Calendar;
with Hardware_Dependencies; use Hardware_Dependencies;
package IRDS is

-- This package contains:
--      1.  Service Data Structures -- From ISO/IEC 10728, Clause 8
--      2.  Service Formats and Descriptions -- From ISO/IEC 10728, Clause 9

-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
--    1. Service Data Structures -- From ISO/IEC 10728, Clause 8
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------

          ----------------------------------------------------------------------
          ----------------------------------------------------------------------
          -- Subclause 8.1.1  Name Length Limits

          -- Name_Limit and Variation_Limit are in package Hardware_Dependencies.

          ----------------------------------------------------------------------
          -- Subclause 8.1.2  Attribute Length Limits

          -- Text_Limit is in package Hardware_Dependencies.

          -- Note that as access types are used, Text_Limit is not needed in
          -- this language binding. Knowledge of this constant may be useful.

          ----------------------------------------------------------------------
          -- Subclause 8.1.3  Control Identifier Length Limits

          -- Session_Id_Limit, Cursor_Id_Limit,
          -- Implementation_Dictionary_Name_Length, and Key_Length are in
          -- package Hardware_Dependencies.
```

```
--~----------------------------------------------------------------------
-- Subclause 8.1.4  Data Types

type Data_Type is (                                -- See subclause 8.2.1 for:
        IRDS_Data_Character,                       -- Character_Type
        IRDS_Data_Character_Varying,               -- Text_Pointer_Type
        IRDS_Data_National_Character,              -- National_Character_Type
        IRDS_Data_National_Character_Varying,
                                 -- National_Text_Pointer_Type
        IRDS_Data_Real,                            -- Real_Type
        IRDS_Data_Double,                          -- Long_Float_Type
        IRDS_Data_Float,                           -- Float_Type
        IRDS_Data_Integer,                         -- Integer_Type
        IRDS_Data_Small_Integer,                   -- Short_Integer_Type
        IRDS_Data_Numeric,                         -- Fixed_Type
        IRDS_Data_Decimal,                         -- Decimal_Type
        IRDS_Data_Date,                            -- Calendar.Time
        IRDS_Data_Time,                            -- Calendar.Time
        IRDS_Data_Time_Stamp,                      -- Calendar.Time
        IRDS_Data_Interval,                        -- Interval_Type
        IRDS_Data_IRDS_Key);                       -- IRDS_Key_Type

--~----------------------------------------------------------------------
-- Subclause 8.1.5  IRD Content Status Classes

type IRD_Content_Status_Class_Type is (
                    IRD_Content_Status_Class_Uncontrolled,
                    IRD_Content_Status_Class_Controlled,
                    IRD_Content_Status_Class_Archived);

--~----------------------------------------------------------------------
-- Subclause 8.1.6  Close Type Parameter

type Close_Type is (    Request_IRDS_Commit,
                        Request_IRDS_Rollback);

--~----------------------------------------------------------------------
```