

INTERNATIONAL STANDARD

ISO/IEC
10728

First edition
1993-04-15

AMENDMENT 4
1998-05-01

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

AMENDMENT 4: RPC IDL binding

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

AMENDEMENT 4: Connexion RPC IDL



Reference number
ISO/IEC 10728:1993/Amd.4:1998(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 4 to 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 1998

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

Introduction

This Amendment to ISO/IEC 10728 defines the RPC IDL interface to an ISO IRDS.

An ISO IRDS RPC IDL binding makes ISO IRDS repository facilities available to an RPC environment. This:

- a) increases the availability of ISO IRDS functionality;
- b) provides IRDS access at low cost to a wider community.

The range of languages from which the ISO IRDS Services Interface can be accessed is increased to all of those that can access services defined by the RPC IDL. Tool vendors, tool builders, data providers and data consumers will benefit from this broader accessibility of an ISO IRDS.

The general principles set out in ISO/IEC TR10182:1993, *Information technology — Programming languages, their environments and system software interfaces — Guidelines for language bindings*, have been considered during the development of this Amendment.

This page intentionally left blank

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 10728:1993/Amd 4:1998

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

AMENDMENT 4: RPC IDL binding

Page v

Contents

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

“Annex F- RPC IDL binding”

Page 1

Clause 1

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

“A language binding for RPC IDL is provided in Annex F.”

Page 1

Clause 2

Add a new Normative Reference.

“ISO/IEC 11578:1996 Information technology - Open Systems Interconnection
- Remote Procedure Call (RPC) ”

Page 5

Subclause 4.4

Add a new sentence in subclause 4.4.

“Data structures for use with RPC IDL are defined in Annex F.”

Page 5

Subclause 4.5

Add a new sentence in subclause 4.5

“RPC IDL bindings for the services are provided in Annex F.”

Page 71

Subclause 8.1

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

For the Pascal language binding specified in this clause, the C language binding specified in Annex C, the Ada language binding specified in Annex D, the CORBA IDL binding specified in Annex E and the RPC IDL binding specified in Annex F, enumerated types are ...”

Page 75

Clause 9

Amend the first sentence of the second paragraph of Clause 9 to read:

“The service formats are specified in this clause using ISO standard Pascal. Alternative service formats for use with the C language binding are specified in Annex C, for the Ada language binding in Annex D, the CORBA IDL binding in Annex E and the RPC IDL binding in Annex F.”

Page 105

Add a new Annex F as follows:

Annex F (normative)

RPC IDL Binding

The IRDS Services Interface language bindings for the IDL Interface Definition Language are presented in the form of a .idl file as set out in clause F.3 below.

F.1 Strategy for the Language Binding

In this binding the data names and data structures defined in clause 8 have been adhered to except where the RPC IDL does not provide an appropriate construct.

Use of this language binding requires the use of a RPC IDL compiler that conforms to ISO/IEC 11578.

F.2 General Rules

1. Those data names in F.3 below that also appear in Clause 8 shall have the same meaning as is defined in Clause 8. The same rules for the use of separators as defined in Clause 8 shall apply.
2. The function and parameter names in F.3 below shall have the same meaning as is defined in Clause 9.
3. The Service Return Codes returned shall be those defined by Clause 9 and Annex A and they shall have the same meaning.
4. The following mappings from SQL data types to RPC IDL data types have been used:

SQL DATA TYPE	RPC IDL DATA TYPE
CHARACTER	char *
CHARACTER VARYING	char *
NATIONAL CHARACTER	char *
NATIONAL CHARACTER VARYING	char *
REAL	float
DOUBLE PRECISION	double
FLOAT	float
INTEGER	long
SMALLINT	short
NUMERIC	long
DECIMAL	long
DATE	IrdsDate
TIME	IrdsTime
TIMESTAMP	IrdsTimeStamp
INTERVAL	IrdsInterval

5. Every function returns an int which is to be set to the value of the NumStates field of the RetCode returned by the function.

F.3 IDL Interface Definition

The following is the IDL interface definition to an ISO IRDS.

```
local interface irds
{
    /* clause 8.1.3 - these definitions are used in 8.2.3 below */

    /* The values n1, n2, n3, n4 in this clause are to be replaced by the
    implementor by suitable implementor defined values as in clause 8.1.3 of
    ISO/IEC 10728. */

    const short IrdsSessIdLim = n1;          /* n1 e.g 255 */
                                              /* IrdsSessIdLim used in 8.2.3 below */
    const short IrdsCurIdLim = n2;          /* n2 e.g 255 */
                                              /* IrdsCurIdLim used in 8.2.3 below */
    const short IrdsImpDicNameLen = n3;      /* n3 e.g 255 */
                                              /* IrdsImpDicNameLen used in 8.2.3 below */
    const short IrdsKeyLen = n4;             /* n4 e.g 255 */
                                              /* IrdsKeyLen used in 8.2.1 below */

    /* Clause 8.1.4 Data Types */

typedef enum
{
    IrdsDataTypeChar,          /* SQL CHARACTER */
    IrdsDataTypeCharaVar,     /* SQL CHARACTER VARYING */
    IrdsDataTypeNatCharaVar,  /* SQL NATIONAL CHARACTER VARYING */
    IrdsDataTypeReal,         /* SQL REAL */
    IrdsDataTypeDouble,       /* SQL DOUBLE PRECISION */
    IrdsDataTypeFloat,        /* SQL FLOAT */
    IrdsDataTypeInteger,      /* SQL INTEGER */
    IrdsDataTypeSmallint,     /* SQL SMALLINT */
    IrdsDataTypeNumeric,      /* SQL NUMERIC */
    IrdsDataTypeDecimal,      /* SQL DECIMAL */
    IrdsDataTypeDate,         /* SQL DATE */
    IrdsDataTypeTime,         /* SQL TIME */
    IrdsDataTypeTimestamp,    /* SQL TIMESTAMP */
    IrdsDataTypeInterval,     /* SQL INTERVAL */
    IrdsDataTypeIrdsKey,      /* SQL IRDS KEY */
} IrdsDataType;

/* Clause 8.1.5 IRD Content Status Classes */
typedef enum
{
    IrdsDcsClsUcntl,          /* Uncontrolled */
    IrdsDcsClsCntl,           /* Controlled */
    IrdsDcsClsArch            /* Archived */
} IrdsDcsCls;

/* Clause 8.1.6 Close Type parameter */
typedef enum
{
    RequestIrdsCommit,        /* COMMIT */
    RequestIrdsRollback       /* ROLLBACK */
} IrdsCloseType;
```

/* Clause 8.2.1 Column data types */

typedef struct

```
{
    char        Year[4];
    char        Sep1;
    char        Month[2];
    char        Sep2;
    char        Day[2];
} IrdsDate;
```

typedef struct

```
{
    char        Hour[2];
    char        Sep1;
    char        Minute[2];
    char        Sep2;
    char        Second[2];
    char        Sep3;
    char        Fraction[3];
} IrdsTime;
```

typedef struct

```
{
    IrdsDate    Date;
    char        SepT;
    IrdsTime    Time;
} IrdsTimestamp;
```

typedef struct

```
{
    char        Days[7];
    char        SepI;
    IrdsTime    Time;
} IrdsInterval;
```

typedef char IrdsKey[IrdsKeyLen];

/* IrdsKeyLen is defined in 8.1.3 above */

/* Clause 8.2.2 Object Names */

typedef char IrdsSQLName[128];

/* 128 is set by ISO/IEC 9075:1992 database Language SQL */

typedef char IrdsName[IrdsNameLim];

/* IrdsNameLim is defined in 8.1.1 above */

typedef char IrdsVarName[IrdsVarLim];

/* IrdsVarLim is defined in 8.1.1 above */

typedef char UserId[IrdsNameLim];

/* IrdsNameLim is defined in 8.1.1 above */


```

/* Clause 8.2.4 Diagnostics Area */

typedef struct
{
    char          StateClass[2];
    char          StateSubClass[3];
} IrdState;

/* IrdState is used in 8.2.5 below */

typedef struct
{
    short          IrdStateSeq;
    IrdState       IrdReturnedState;
    IrdSQLName     IrdConstraintSchema;
    IrdSQLName     IrdConstraintName;
    IrdSQLName     IrdSchemaName;
    IrdSQLName     IrdTableName;
    short          IrdColumnNumber;
    IrdSQLName     IrdColumnName;
} IrdStateRec;

/* IrdStateRec is used in the Get Diagnostics
   Service in 9.1.8 below */

/* Clause 8.2.5 Service Return Code */
typedef struct
{
    short NumStates;
    IrdState State;
} IrdRetCode;

/* In the C binding the column list actually is a list.
   In this IDL binding the list is a sequence.
*/

typedef union IrdDataTypeUnion switch(IrdDataType ColType)
{
    case IrdDataTypeChar:
    case IrdDataTypeCharVar:
    case IrdDataTypeNatChar:
    case IrdDataTypeNatCharVar: [string, ptr] char* ColValText;
    case IrdDataTypeReal:      double      ColValReal;
    case IrdDataTypeFloat:     float       ColValFloat;
    case IrdDataTypeInteger:    long        ColValLongInteger;
    case IrdDataTypeSmallint:   short       ColValShortInteger;
    case IrdDataTypeNumeric:    long        ColValLongNumeric;
    case IrdDataTypeDate:       IrdDate     ColValDate;
    case IrdDataTypeTime:       IrdTime     ColValTime;
    case IrdDataTypeTimestamp:  IrdTimestamp ColValTimestamp;
    case IrdDataTypeInterval:   IrdInterval ColValInterval;
    case IrdDataTypeIrdKey:     [string, ptr] char* ColValIrdKey;
} IrdDataTypeUnion;

```

```

struct IrdsSingleColSpec
{
    IrdsSQLName      ColName;
    boolean          ColNull;
    IrdsDataTypeUnion IrdsColVal;
};

typedef struct
{
    short                      NumCols;
    char                      StateSubClass[3];
    [size_is(NumCols)] IrdsSingleColSpec ColSpecList[];
} IrdsColList;

/* Clause 9.1.1 Create IRD Definition Service */

short IrdsCreateIRDefinition
(
    [in] UserId      IrdsUser,
    [in] IrdsImpDicName IrdDefName,
    [out] IrdsSessId  CurrSessId,
    [out] IrdsRetCode RetCode
);

/* Clause 9.1.2 Drop IRD Definition Service */

short IrdsDropIRDefinition
(
    [in] UserId      IrdsUser,
    [in] IrdsImpDicName IrdDefName,
    [out] IrdsRetCode RetCode
);

/* Clause 9.1.3 Open IRDS Service */

short IrdsOpen
(
    [in] UserId      IrdsUser,
    [in] IrdsImpDicName IrdDefName,
    [in] IrdsName      IrdDicName,
    [in] boolean       WillUpdate,
    [out] handle_t *    CurrSessId,
    [out] IrdsRetCode  RetCode
);

} /* End of definition of interface IrdsServicesInterfaceProcessor */

[explicit_handle] interface IrdsSession
{
    /* Clause 9.1.4 Prepare Service */

    short IrdsPrepare
    (
        [in] handle_t      IrdsSessId,
        [out] IrdsRetCode  RetCode
    );

```

```
/* Clause 9.1.5 Commit Service */

short IrdsCommit
(
    [in] handle_t      IrdsSessId,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.1.6 Rollback Service */

short IrdsRollback
(
    [in] handle_t      IrdsSessId,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.1.7 Close IRDS Service */

short IrdsClose
(
    [in] handle_t      IrdsSessId,
    [in] IrdsCloseType CloseType,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.1.8 Get Diagnostics Service */
short IrdsGetDiagnostics
(
    [in] handle_t      IrdsSessId,
    [in] short          StateNum,
    [out] IrdsStateRec  StateRec,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.2.1 Set Context Service */

short IrdsSetContext
(
    [in] handle_t      IrdsSessId,
    [in] IrdsName       SessWkgSetName,
    [in] IrdsName       SessWkgSetVerIrd,
    [in] boolean        WillUpdate,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.2.2 Add Object Service */

short IrdsAddObject
(
    [in] handle_t      IrdsSessId,
    [in] IrdsName       ObjType,
    [in] IrdsCollList   NewCols,
    [out] IrdsRetCode   RetCode
);
```

```

/* Clause 9.2.3 Open Cursor Service */
short IrdsOpenCursor
(
    [in] handle_t      IrdsSessId,
    [in] boolean       UseContext,
    [in, string, ptr] char*   ObjSelExpr,
    [in] IrdsName       WkgSetName,
    [in] IrdsName       WkgSetVerId,
    [in] boolean       FullContext,
    [in] boolean       WillUpdate,
    [out] IrdsCurId    ObjCurId,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.2.4 Retrieve Object Service */

short IrdsRetrieveObject
(
    [in] handle_t      IrdsSessId,
    [in] IrdsCurId    ObjCurId,
    [in, out] IrdsColList RequestedCols,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.2.5 Modify Object Service */

short IrdsModifyObject
(
    [in] handle_t      IrdsSessId,
    [in] IrdsCurId    ObjCurId,
    [in] IrdsColList   ModifiedCols,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.2.6 Delete Object Service */

short IrdsDeleteObject
(
    [in] handle_t      IrdsSessId,
    [in] IrdsCurId    ObjCurId,
    [out] IrdsRetCode   RetCode
);

/* Clause 9.2.7 IrdsDeclassifyObject Service */

short IrdsDeclassifyObject
(
    [in] handle_t      IrdsSessId,
    [in] IrdsCurId    ObjCurId,
    [out] IrdsRetCode   RetCode
);

```