

INTERNATIONAL STANDARD

ISO/IEC
11578

First edition
1996-12-15

Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)

*Technologies de l'information — Interconnexion de systèmes ouverts
(OSI) — Appel de procédures à distance (RPC)*



Reference number
ISO/IEC 11578:1996(E)

Contents

Section 1	General	1
1.1	Scope	2
1.2	Normative references	3
1.3	Informative References	4
1.4	Definitions	5
1.4.1	Reference model definitions	5
1.4.2	Application Layer Structure definitions	5
1.4.3	Service conventions definitions	5
1.5	Abbreviations	6
1.6	Portability	8
1.7	Services and Protocols	9
1.8	Conformance Requirements	10
Section 2	Introduction to the RPC API	13
2.1	RPC Programming Model Overview	14
2.1.1	Client/Server Model	14
2.1.1.1	Interfaces	14
2.1.1.2	Remoteness	14
2.1.1.3	Binding	15
2.1.1.4	Name Services	15
2.1.1.5	Resource Models	15
2.1.1.6	Security Services	16
2.1.1.7	Server Implementation	16
2.1.2	Application/Stub/Run-time System	16
2.1.2.1	RPC Run Time	16
2.1.2.2	Stubs	16
2.1.2.3	Application Code	17
2.2	API Operations	18
2.2.1	Binding-related Operations	18
2.2.2	Name Service Operations	18
2.2.3	Endpoint Operations	18
2.2.4	Security Operations	19
2.2.5	Stub Memory Management Operations	19
2.2.6	Management Operations	19
2.2.7	UUID Operations	19
2.3	Binding	20
2.3.1	Binding Handles	21
2.3.1.1	Client and Server Binding Handles	21
2.3.1.2	Obtaining Binding Handles	21
2.3.2	String Bindings	21
2.3.3	Binding Steps	22
2.3.3.1	Server Binding Steps	22
2.3.3.2	Client Binding Steps	26
2.3.3.3	Call Routing Algorithms	26
2.3.4	Binding Methods	29

© ISO/IEC 1996

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

2.4	Name Service Interface	30
2.4.1	Name Service Model	30
2.4.2	Name Syntax Tags	30
2.4.3	Name Service Attributes	31
2.4.3.1	Server Entries	31
2.4.3.2	Group Entries	31
2.4.3.3	Profiles	31
2.4.4	Binding Searches	32
2.4.5	Search Algorithm	33
2.4.6	Name Service Caching	34
2.5	Server Model	35
2.5.1	Server Concurrency and Request Buffering	35
2.5.2	Management Interface	35
2.6	Server Resource Models	36
2.6.1	The Server-Oriented Model	36
2.6.2	The Service-Oriented Model	36
2.6.3	The Object-Oriented Model	36
2.7	Security	37
2.8	Error Handling	37
2.9	Cancel Notification	37
2.10	Stubs	39
2.10.1	IDL to Stub Data Type Mappings	39
2.10.2	Manager EPVs	39
2.10.3	Interface Handles	39
2.10.4	Stub Memory Management	39
2.11	RPC API Routine Taxonomy	40
2.11.1	Binding Operations	40
2.11.2	Interface Operations	40
2.11.3	Protocol Sequence Operations	41
2.11.4	Local Endpoint Operations	41
2.11.5	Object Operations	41
2.11.6	Name Service Interface Operations	42
2.11.6.1	NSI Binding Operations	42
2.11.6.2	NSI Entry Operations	42
2.11.6.3	NSI Group Operations	42
2.11.6.4	NSI Profile Operations	43
2.11.7	Authentication Operations	43
2.11.8	The Server Listen Operation	43
2.11.9	The String Free Operation	43
2.11.10	UUID Operations	43
2.11.11	Stub Memory Management	44
2.11.12	Endpoint Management Operations	44
2.11.13	Name Service Management Operations	44
2.11.14	Local Management Services	45
2.11.15	Local/Remote Management Services	45
2.11.16	Error Messages	45
Section 3	RPC API Manual Pages	47
3.1	RPC Data Types	47
3.1.1	Unsigned Integer Types	47
3.1.2	Signed Integer Type	47
3.1.3	Unsigned Character String	47
3.1.4	Binding Handle	48
3.1.5	Binding Vector	49
3.1.6	Boolean Type	49
3.1.7	Endpoint Map Inquiry Handle	49
3.1.8	Interface Handle	50
3.1.9	Interface Identifier	50
3.1.10	Interface Identifier Vector	51
3.1.11	Manager Entry Point Vector	51

3.1.12	Name Service Handle	51
3.1.13	Protocol Sequence String	52
3.1.14	Protocol Sequence Vector	52
3.1.15	Statistics Vector	53
3.1.16	String Binding	53
3.1.17	String UUID	54
3.1.18	UUIDs	54
3.1.19	UUID Vector	54
	<code>rpc_binding_copy()</code>	56
	<code>rpc_binding_free()</code>	57
	<code>rpc_binding_from_string_binding()</code>	58
	<code>rpc_binding_inq_auth_client()</code>	59
	<code>rpc_binding_inq_auth_info()</code>	61
	<code>rpc_binding_inq_object()</code>	63
	<code>rpc_binding_reset()</code>	64
	<code>rpc_binding_server_from_client()</code>	65
	<code>rpc_binding_set_auth_info()</code>	66
	<code>rpc_binding_set_object()</code>	68
	<code>rpc_binding_to_string_binding()</code>	69
	<code>rpc_binding_vector_free()</code>	70
	<code>rpc_ep_register()</code>	71
	<code>rpc_ep_register_no_replace()</code>	74
	<code>rpc_ep_resolve_binding()</code>	76
	<code>rpc_ep_unregister()</code>	78
	<code>rpc_if_id_vector_free()</code>	79
	<code>rpc_if_inq_id()</code>	80
	<code>rpc_mgmt_ep_elt_inq_begin()</code>	81
	<code>rpc_mgmt_ep_elt_inq_done()</code>	84
	<code>rpc_mgmt_ep_elt_inq_next()</code>	85
	<code>rpc_mgmt_ep_unregister()</code>	87
	<code>rpc_mgmt_inq_com_timeout()</code>	89
	<code>rpc_mgmt_inq_dflt_protect_level()</code>	90
	<code>rpc_mgmt_inq_if_ids()</code>	91
	<code>rpc_mgmt_inq_server_princ_name()</code>	93
	<code>rpc_mgmt_inq_stats()</code>	95
	<code>rpc_mgmt_is_server_listening()</code>	97
	<code>rpc_mgmt_set_authorization_fn()</code>	99
	<code>rpc_mgmt_set_cancel_timeout()</code>	101
	<code>rpc_mgmt_set_com_timeout()</code>	102
	<code>rpc_mgmt_set_server_stack_size()</code>	104
	<code>rpc_mgmt_stats_vector_free()</code>	105
	<code>rpc_mgmt_stop_server_listening()</code>	106
	<code>rpc_network_inq_protseqs()</code>	107
	<code>rpc_network_is_protseq_valid()</code>	108
	<code>rpc_ns_binding_export()</code>	110
	<code>rpc_ns_binding_import_begin()</code>	112
	<code>rpc_ns_binding_import_done()</code>	114
	<code>rpc_ns_binding_import_next()</code>	115
	<code>rpc_ns_binding_inq_entry_name()</code>	118
	<code>rpc_ns_binding_lookup_begin()</code>	120
	<code>rpc_ns_binding_lookup_done()</code>	122
	<code>rpc_ns_binding_lookup_next()</code>	123
	<code>rpc_ns_binding_select()</code>	126
	<code>rpc_ns_binding_unexport()</code>	127
	<code>rpc_ns_entry_expand_name()</code>	129
	<code>rpc_ns_entry_inq_resolution()</code>	130
	<code>rpc_ns_entry_object_inq_begin()</code>	132
	<code>rpc_ns_entry_object_inq_done()</code>	133
	<code>rpc_ns_entry_object_inq_next()</code>	134

	<code>rpc_ns_group_delete()</code>	136
	<code>rpc_ns_group_mbr_add()</code>	137
	<code>rpc_ns_group_mbr_inq_begin()</code>	139
	<code>rpc_ns_group_mbr_inq_done()</code>	141
	<code>rpc_ns_group_mbr_inq_next()</code>	142
	<code>rpc_ns_group_mbr_remove()</code>	144
	<code>rpc_ns_mgmt_binding_unexport()</code>	146
	<code>rpc_ns_mgmt_entry_create()</code>	149
	<code>rpc_ns_mgmt_entry_delete()</code>	150
	<code>rpc_ns_mgmt_entry_inq_if_ids()</code>	151
	<code>rpc_ns_mgmt_handle_set_exp_age()</code>	153
	<code>rpc_ns_mgmt_inq_exp_age()</code>	154
	<code>rpc_ns_mgmt_set_exp_age()</code>	156
	<code>rpc_ns_profile_delete()</code>	158
	<code>rpc_ns_profile_elt_add()</code>	159
	<code>rpc_ns_profile_elt_inq_begin()</code>	161
	<code>rpc_ns_profile_elt_inq_done()</code>	164
	<code>rpc_ns_profile_elt_inq_next()</code>	165
	<code>rpc_ns_profile_elt_remove()</code>	167
	<code>rpc_object_inq_type()</code>	169
	<code>rpc_object_set_inq_fn()</code>	171
	<code>rpc_object_set_type()</code>	172
	<code>rpc_protseq_vector_free()</code>	174
	<code>rpc_server_inq_bindings()</code>	175
	<code>rpc_server_inq_if()</code>	177
	<code>rpc_server_listen()</code>	178
	<code>rpc_server_register_auth_info()</code>	180
	<code>rpc_server_register_if()</code>	182
	<code>rpc_server_unregister_if()</code>	185
	<code>rpc_server_use_all_protseqs()</code>	187
	<code>rpc_server_use_all_protseqs_if()</code>	189
	<code>rpc_server_use_protseq()</code>	191
	<code>rpc_server_use_protseq_ep()</code>	193
	<code>rpc_server_use_protseq_if()</code>	195
	<code>rpc_sm_allocate()</code>	197
	<code>rpc_sm_client_free()</code>	198
	<code>rpc_sm_destroy_client_context()</code>	199
	<code>rpc_sm_enable_allocate()</code>	200
	<code>rpc_sm_free()</code>	201
	<code>rpc_sm_get_thread_handle()</code>	202
	<code>rpc_sm_set_client_alloc_free()</code>	203
	<code>rpc_sm_set_thread_handle()</code>	204
	<code>rpc_sm_swap_client_alloc_free()</code>	205
	<code>rpc_string_binding_compose()</code>	206
	<code>rpc_string_binding_parse()</code>	207
	<code>rpc_string_free()</code>	209
	<code>uuid_compare()</code>	211
	<code>uuid_create()</code>	212
	<code>uuid_create_nil()</code>	213
	<code>uuid_equal()</code>	214
	<code>uuid_from_string()</code>	215
	<code>uuid_is_nil()</code>	216
	<code>uuid_to_string()</code>	217
Section	4 Interface Definition Language	219
4.1	Notation	219
4.2	IDL Language Specification	220
4.2.1	IDL Lexemes	220
4.2.1.1	Keywords and Reserved Words	220
4.2.1.2	Identifiers	220

4.2.1.3	IDL Punctuation	220
4.2.1.4	Alternate Representation of Braces	221
4.2.1.5	White Space	221
4.2.2	Comments	221
4.2.3	Interface Definition Structure	221
4.2.4	Interface Header	221
4.2.4.1	The uuid Attribute	222
4.2.4.2	The version Attribute	222
4.2.4.3	The endpoint Attribute	222
4.2.4.4	The local Attribute	222
4.2.4.5	The pointer_default Attribute	222
4.2.5	Interface Body	223
4.2.6	Import Declaration	223
4.2.7	Constant Declaration	223
4.2.7.1	Syntax	223
4.2.7.2	Semantics and Restrictions	224
4.2.8	Type Declarations and Tagged Declarations	224
4.2.9	Base Types	225
4.2.9.1	Syntax	225
4.2.9.2	Integer Types	225
4.2.9.3	The char Types	226
4.2.9.4	The boolean Type	226
4.2.9.5	The byte Type	226
4.2.9.6	The void Type	226
4.2.9.7	The handle_t Type	226
4.2.10	Constructed Types	226
4.2.11	Structures	226
4.2.12	Unions	226
4.2.12.1	Syntax	226
4.2.12.2	Semantics and Restrictions	227
4.2.13	Enumerated Types	228
4.2.14	Pipes	228
4.2.14.1	Syntax	228
4.2.14.2	Semantics and Restrictions	228
4.2.15	Arrays	228
4.2.15.1	Syntax	228
4.2.15.2	Semantics and Restrictions	229
4.2.15.3	Arrays of Arrays	229
4.2.16	Type Attributes	229
4.2.16.1	Syntax	229
4.2.16.2	Semantics and Restrictions	229
4.2.16.3	The transmit_as Attribute	229
4.2.16.4	The handle Attribute	230
4.2.16.5	The string Attribute	230
4.2.16.6	The context_handle Attribute	230
4.2.17	Field Attributes	230
4.2.17.1	Syntax	231
4.2.17.2	Semantics and Restrictions	231
4.2.17.3	The ignore Attribute	231
4.2.18	Field Attributes in Array Declarations	231
4.2.18.1	Conformant Arrays	231
4.2.18.2	Varying and Conformant Varying Arrays	232
4.2.18.3	Relationships Between Attributes	233
4.2.18.4	Negative Size and Length Specifications	233
4.2.19	Field Attributes in String Declarations	233
4.2.19.1	The first_is, last_is and length_is Attributes	233
4.2.19.2	The max_is Attribute	233
4.2.19.3	The size_is Attribute	234
4.2.20	Pointers	234

	4.2.20.1	Syntax	234
	4.2.20.2	Semantics and Restrictions	234
	4.2.20.3	Attributes Applicable to Pointers	235
	4.2.20.4	Varying Arrays of Pointers	235
	4.2.20.5	Restrictions on Pointers	236
	4.2.21	Pointers as Arrays	237
	4.2.21.1	Pointers with the string Attribute	237
	4.2.21.2	Possible Ambiguity Resolved	237
	4.2.22	Operations	237
	4.2.22.1	The idempotent Attribute	238
	4.2.22.2	The broadcast Attribute	238
	4.2.22.3	The maybe Attribute	238
	4.2.23	Parameter Declarations	238
	4.2.23.1	Syntax	238
	4.2.23.2	Semantics and Restrictions	238
	4.2.23.3	Directional Attributes	238
	4.2.23.4	Aliasing in Parameter Lists	239
	4.2.24	Predefined Types	239
	4.2.25	The error_status_t Type	239
	4.2.26	International Character Types	239
	4.2.27	Anonymous Types	240
	4.3	The Attribute Configuration Source	241
	4.3.1	Comments	241
	4.3.2	Identifiers	241
	4.3.3	Syntax	241
	4.3.4	Include Declaration	242
	4.3.5	Specifying Binding Handles	243
	4.3.5.1	The explicit_handle Attribute	243
	4.3.5.2	The implicit_handle Attribute	243
	4.3.5.3	The auto_handle Attribute	243
	4.3.6	The represent_as Attribute	244
	4.3.7	The code and nocode Attributes	244
	4.3.8	The in_line and out_of_line Attributes	244
	4.3.9	Return Statuses	245
	4.3.9.1	The comm_status Attribute	245
	4.3.9.2	The fault_status Attribute	245
	4.3.9.3	Interaction of the comm_status and fault_status Attributes	246
	4.3.10	The heap Attribute	246
	4.3.11	The enable_allocate Attribute	246
	4.4	IDL Grammar Synopsis	247
	4.4.1	Grammar Synopsis	247
	4.4.2	Alphabetic Listing of Productions	250
	4.5	IDL Constructed Identifiers	253
	4.6	IDL and ACS Reserved Words	254
Section	5	Stubs	255
	5.1	The Application/Stub Interface	255
	5.1.1	Parameters	255
	5.1.1.1	Parameter Memory Management	255
	5.1.1.2	Client-side Allocation	256
	5.1.1.3	Server-side Allocation	256
	5.1.1.4	Aliasing	257
	5.1.2	Default Manager EPVs	257
	5.1.3	Interface Handle	257
	5.1.4	Pipes	257
	5.1.4.1	Processing of in Pipes	259
	5.1.4.2	Processing of out Pipes	260
	5.1.4.3	Processing of in,out Pipes	261
	5.1.5	IDL and ACS Type Attributes	261
	5.1.5.1	The IDL transmit_as Attribute	261

	5.1.5.2	The IDL handle Attribute	262
	5.1.5.3	Interaction of IDL transmit_as and IDL handle Attributes	263
	5.1.5.4	The ACS represent_as Attribute	263
	5.1.5.5	Interaction of the ACS represent_as Attribute and the IDL handle Attribute	264
	5.1.5.6	Interaction of the ACS represent_as Attribute with the IDL transmit_as Attribute	264
	5.1.6	Context Handle Rundown	264
	5.2	Interoperability Requirements on Stubs	266
	5.2.1	Operation Numbers	266
	5.2.2	Error Handling During Floating-Point Unmarshalling	266
Section	6	Remote Procedure Call Model	267
	6.1	Client/Server Execution Model	268
	6.1.1	RPC Interface and RPC Object	268
	6.1.1.1	RPC Interfaces	268
	6.1.1.2	RPC Objects	268
	6.1.2	Interface Version Numbering	269
	6.1.2.1	Rules for Changing Version Numbers	269
	6.1.2.2	Definition of an Upwardly Compatible Change	269
	6.1.2.3	Non-upwardly Compatible Changes	269
	6.1.3	Remote Procedure Calls	269
	6.1.4	Nested RPCs	270
	6.1.5	Execution Semantics	270
	6.1.6	Context Handles	271
	6.1.7	Threads	271
	6.1.8	Cancels	273
	6.2	Binding, Addressing and Name Services	275
	6.2.1	Binding	275
	6.2.2	Endpoints and the Endpoint Mapper	276
	6.2.2.1	Client Operation	276
	6.2.2.2	Server Operation	277
	6.2.3	NSI Interface	277
	6.2.3.1	Common Declarations	277
	6.2.3.2	Protocol Towers	278
	6.2.3.3	The server_name Object Attributes	279
	6.2.3.4	The group Object Attributes	281
	6.2.3.5	The profile Object Attributes	282
	6.2.3.6	Encoding	283
	6.2.3.7	Name Service Class Values	283
	6.3	Error Handling Model	284
Section	7	RPC Service Definition	285
	7.1	Call Representation Data Structure	285
	7.2	Service Primitives	285
	7.2.1	Invoke	286
	7.2.2	Result	287
	7.2.3	Cancel	288
	7.2.4	Error	289
	7.2.5	Reject	290
Section	8	Statechart Specification Language Semantics	291
	8.1	The Elements of Statecharts	291
	8.2	State Hierarchies	293
	8.3	Concurrency	293
	8.4	Graphical Expressions	294
	8.4.1	Default Entrances	294
	8.4.2	Conditional Connectors	294
	8.4.3	Terminal Connectors	294
	8.5	Semantics that Require Special Consideration	295
	8.5.1	Implicit Exits and Entrances (Scope of Transitions)	295
	8.5.2	Conflicting Transitions	295

	8.5.3	Execution Steps and Time	295
	8.5.4	Synchronisation and Race Conditions	296
	8.6	Summary of Language Elements	297
	8.6.1	Event Expressions	297
	8.6.2	Condition Expressions	298
	8.6.3	Action Expressions	298
	8.6.4	Data Item Expressions	300
	8.6.4.1	Atomic Numeric Expressions	300
	8.6.4.2	Compound Numeric Expressions	300
	8.6.4.3	String Expressions	300
Section	9	RPC Protocol Definitions	301
	9.1	Conformance	301
	9.2	RPC Stub to Run-time Protocol Machine Interactions	302
	9.2.1	Client Protocol Machines	302
	9.2.2	Server Protocol Machines	303
	9.3	Connection-oriented Protocol	304
	9.3.1	Client/Server	304
	9.3.2	Association Group	304
	9.3.3	Association	304
	9.3.3.1	Association Management Policy	305
	9.3.3.2	Primary and Secondary Endpoint Addresses	305
	9.3.4	Call	305
	9.3.5	Transport Service Requirements	305
	9.4	Connection-oriented Protocol Machines	306
	9.4.1	CO_CLIENT_ALLOC	306
	9.4.2	CO_CLIENT_GROUP	306
	9.4.3	CO_CLIENT	306
	9.4.3.1	ASSOCIATION	307
	9.4.3.2	CONTROL	307
	9.4.3.3	CANCEL	307
	9.4.3.4	CALL	307
	9.4.4	CO_SERVER_GROUP	307
	9.4.5	CO_SERVER	307
	9.4.5.1	ASSOCIATION	308
	9.4.5.2	CONTROL	308
	9.4.5.3	CANCEL	308
	9.4.5.4	WORKING	308
	9.5	Connectionless Protocol	309
	9.5.1	Client/Server	309
	9.5.2	Activity	309
	9.5.3	Call	309
	9.5.4	Maintaining Execution Context and Monitoring Liveness	309
	9.5.5	Serial Numbers	310
	9.5.6	Transport Service Requirements	310
	9.6	Connectionless Protocol Machines	311
	9.6.1	RPC Stub to Run Time Protocol Machine Interactions	311
	9.6.2	CL_CLIENT	311
	9.6.2.1	CONTROL	311
	9.6.2.2	AUTHENTICATION	311
	9.6.2.3	CALLBACK	311
	9.6.2.4	PING	311
	9.6.2.5	CANCEL	311
	9.6.2.6	DATA	312
	9.6.3	CL_SERVER	312
	9.6.3.1	CONTROL	312
	9.6.3.2	AUTHENTICATION	312
	9.6.3.3	CANCEL	312
	9.6.3.4	WORKING	312
	9.7	Naming Conventions	313

Section	10	Connectionless RPC Protocol Machines	315
	10.1	CL_CLIENT Machine	316
	10.1.1	CL_CLIENT Activities	316
	10.1.2	CL_CLIENT States	318
	10.1.3	CL_CLIENT Events	323
	10.1.4	CL_CLIENT Conditions	326
	10.1.5	CL_CLIENT Actions	331
	10.1.6	CL_CLIENT Data-Items	335
	10.2	CL_SERVER Machine	344
	10.2.1	CL_SERVER Activities	344
	10.2.2	CL_SERVER States	348
	10.2.3	CL_SERVER Events	354
	10.2.4	CL_SERVER Actions	358
	10.2.5	CL_SERVER Conditions	364
	10.2.6	CL_SERVER Data-Items	369
Section	11	Connection-oriented RPC Protocol Machines	381
	11.1	CO_CLIENT Machine	382
	11.1.1	CO_CLIENT Activities	382
	11.1.2	CO_CLIENT States	384
	11.1.3	CO_CLIENT Events	390
	11.1.4	CO_CLIENT Actions	396
	11.1.5	CO_CLIENT Conditions	400
	11.1.6	CO_CLIENT Data-Items	404
	11.2	CO_CLIENT_ALLOC Machine	413
	11.2.1	CO_CLIENT_ALLOC Activities	414
	11.2.2	CO_CLIENT_ALLOC States	415
	11.2.3	CO_CLIENT_ALLOC Events	417
	11.2.4	CO_CLIENT_ALLOC Actions	420
	11.2.5	CO_CLIENT_ALLOC Conditions	421
	11.2.6	CO_CLIENT_ALLOC Data-Items	422
	11.3	CO_CLIENT_GROUP Machine	423
	11.3.1	CO_CLIENT_GROUP States	424
	11.3.2	CO_CLIENT_GROUP Events	425
	11.3.3	CO_CLIENT_GROUP Data-Items	426
	11.4	CO_SERVER Machine	427
	11.4.1	CO_SERVER Activities	428
	11.4.2	CO_SERVER States	430
	11.4.3	CO_SERVER Events	435
	11.4.4	CO_SERVER Actions	441
	11.4.5	CO_SERVER Conditions	446
	11.4.6	CO_SERVER Data-Items	450
	11.5	CO_SERVER_GROUP Machine	459
	11.5.1	CO_SERVER_GROUP States	460
	11.5.2	CO_SERVER_GROUP Events	461
	11.5.3	CO_SERVER_GROUP Actions	463
	11.5.4	CO_SERVER_GROUP Data-Items	463
Section	12	RPC PDU Encodings	465
	12.1	Generic PDU Structure	465
	12.2	Encoding Conventions	466
	12.3	Alignment	466
	12.4	Common Fields	467
	12.4.1	PDU Types	467
	12.4.2	Protocol Version Numbers	467
	12.4.3	Data Representation Format Labels	467
	12.4.4	Reject Status Codes	467
	12.5	Connectionless RPC PDUs	468
	12.5.1	Connectionless PDU Structure	468
	12.5.2	Header Encoding	468
	12.5.2.1	Protocol Version Number	469

12.5.2.2	PDU Type	469
12.5.2.3	Flags Fields	469
12.5.2.4	Data Representation Format Label	470
12.5.2.5	Serial Number	470
12.5.2.6	Object Identifier	470
12.5.2.7	Interface Identifier	470
12.5.2.8	Activity Identifier	470
12.5.2.9	Server Boot Time	470
12.5.2.10	Interface Version	471
12.5.2.11	Sequence Number	471
12.5.2.12	Operation Number	471
12.5.2.13	Interface Hint	471
12.5.2.14	Activity Hint	471
12.5.2.15	PDU Body Length	471
12.5.2.16	Fragment Number	471
12.5.2.17	Authentication Protocol Identifier	472
12.5.3	Connectionless PDU Definitions	472
12.5.3.1	The ack PDU	472
12.5.3.2	The cancel_ack PDU	472
12.5.3.3	The cancel PDU	473
12.5.3.4	The fack PDU	473
12.5.3.5	The fault PDU	474
12.5.3.6	The nocal PDU	474
12.5.3.7	The ping PDU	475
12.5.3.8	The reject PDU	475
12.5.3.9	The request PDU	475
12.5.3.10	The response PDU	475
12.5.3.11	The working PDU	475
12.6	Connection-oriented RPC PDUs	476
12.6.1	Connection-oriented PDU Structure	476
12.6.2	Fragmentation and Reassembly	476
12.6.3	Connection-oriented PDU Data Types	477
12.6.3.1	Declarations	477
12.6.3.2	Connection-Oriented Protocol Versions	479
12.6.3.3	The frag_length Field	480
12.6.3.4	Context Identifiers	480
12.6.3.5	The call_id Field	480
12.6.3.6	The assoc_group_id Field	480
12.6.3.7	The alloc_hint Field	480
12.6.3.8	Authentication Data	480
12.6.3.9	Optional Connect Reject and Disconnect Data	481
12.6.4	Connection-oriented PDU Definitions	481
12.6.4.1	The alter_context PDU	481
12.6.4.2	The alter_context_resp PDU	483
12.6.4.3	The bind PDU	484
12.6.4.4	The bind_ack PDU	485
12.6.4.5	The bind_nak PDU	486
12.6.4.6	The cancel PDU	487
12.6.4.7	The fault PDU	488
12.6.4.8	The orphaned PDU	490
12.6.4.9	The request PDU	491
12.6.4.10	The response PDU	493
12.6.4.11	The shutdown PDU	494
Section 13	Security	495
13.1	The Generic RPC Security Model	496
13.1.1	Generic Operation	496
13.1.2	Generic Encodings	497
13.1.2.1	Protection Levels	497
13.1.2.2	Authentication Services	497

	13.1.2.3	Authorisation Services	498
	13.1.3	Underlying Security Services Required	498
	13.2	Security Services for Connection-oriented Protocol	499
	13.2.1	Client Association State Machine	499
	13.2.2	Server Association State Machine	499
	13.2.3	Sequence Numbers	499
	13.2.4	The auth_context_id Field	500
	13.2.5	Integrity Protection	500
	13.2.6	Connection-oriented Encodings	501
	13.2.6.1	Common Authentication Verifier Encodings	501
	13.2.6.2	Encoding for Per-PDU Services	501
	13.2.6.3	Credentials Encoding	503
	13.3	Security Services for Connectionless Protocol	505
	13.3.1	Server Receive Processing	505
	13.3.2	Client Receive Processing	505
	13.3.3	Conversation Manager Encodings	505
	13.3.3.1	Challenge Request Data Encoding	505
	13.3.3.2	Response Data Encoding	506
	13.3.4	Authentication Verifier Encodings	506
	13.3.4.1	dce_c_authn_level_none	506
	13.3.4.2	dce_c_authn_level_connect	506
	13.3.4.3	dce_c_authn_level_call	506
	13.3.4.4	dce_c_authn_level_pkt	507
	13.3.4.5	dce_c_authn_level_integrity	507
	13.3.4.6	dce_c_authn_level_privacy	507
Section	14	Transfer Syntax NDR	509
	14.1	Data Representation Format Label	510
	14.2	NDR Primitive Types	511
	14.2.1	Representation Conventions	511
	14.2.2	Alignment of Primitive Types	511
	14.2.3	Booleans	512
	14.2.4	Characters	512
	14.2.5	Integers and Enumerated Types	512
	14.2.5.1	Enumerated Types	513
	14.2.6	Floating-point Numbers	513
	14.2.6.1	IEEE Format	514
	14.2.6.2	VAX Format	515
	14.2.6.3	Cray Format	516
	14.2.6.4	IBM Format	517
	14.2.7	Uninterpreted Octets	518
	14.3	NDR Constructed Types	519
	14.3.1	Representation Conventions	519
	14.3.2	Arrays	519
	14.3.2.1	Uni-dimensional Fixed Arrays	519
	14.3.2.2	Uni-dimensional Conformant Arrays	520
	14.3.2.3	Uni-dimensional Varying Arrays	520
	14.3.2.4	Uni-dimensional Conformant-varying Arrays	521
	14.3.2.5	Ordering of Elements in Multi-dimensional Arrays	521
	14.3.2.6	Multi-dimensional Fixed Arrays	521
	14.3.2.7	Multi-dimensional Conformant Arrays	521
	14.3.2.8	Multi-dimensional Varying Arrays	522
	14.3.2.9	Multi-dimensional Conformant and Varying Arrays	522
	14.3.3	Strings	523
	14.3.3.1	Varying Strings	523
	14.3.3.2	Conformant and Varying Strings	524
	14.3.4	Arrays of Strings	524
	14.3.5	Structures	525
	14.3.5.1	Alignment of Constructed Types	525
	14.3.6	Structures Containing Arrays	526

	14.3.6.1	Structures Containing a Conformant Array	526
	14.3.6.2	Structures Containing a Conformant and Varying Array	527
	14.3.7	Unions	527
	14.3.8	Pipes	528
	14.3.9	Pointers	528
	14.3.10	Top-level Pointers	529
	14.3.10.1	Top-level Full Pointers	529
	14.3.10.2	Top-level Reference Pointers	529
	14.3.11	Embedded Pointers	530
	14.3.11.1	Embedded Full Pointers	530
	14.3.11.2	Embedded Reference Pointers	531
	14.3.11.3	Algorithm for Deferral of Referents	531
	14.4	NDR Input and Output Streams	532
Annex	A	Universal Unique Identifier	533
	A.1	Format	534
	A.2	Algorithms for Creating a UUID	536
	A.2.1	Clock Sequence	536
	A.2.2	System Reboot	536
	A.2.3	Clock Adjustment	537
	A.2.4	Clock Overrun	537
	A.2.5	UUID Generation	537
	A.3	String Representation of UUIDs	538
	A.4	Comparing UUIDs	539
Annex	B	Protocol Sequence Strings	541
Annex	C	Name Syntax Constants	543
Annex	D	Authentication, Authorisation and Protection-level Arguments	545
	D.1	The authn_svc Argument	545
	D.2	The authz_svc Argument	545
	D.3	The protect_level Argument	546
	D.4	The privs Argument	547
	D.5	The server_princ_name Argument	547
	D.6	The auth_identity Argument	547
	D.7	Key Functions	547
Annex	E	Reject Status Codes and Parameters	549
	E.1	Reject Status Codes	549
	E.2	Possible Failures	551
	E.2.1	comm_status Parameter	551
	E.2.2	fault_status Parameter	551
Annex	F	IDL to C-language Mappings	553
	F.1	Data Type Bindings	553
	F.2	Syntax Mappings	556
Annex	G	Portable Character Set	559
Annex	H	Endpoint Mapper Well-known Ports	561
Annex	I	Protocol Identifiers	563
Annex	J	DCE CDS Attribute Names	565
Annex	K	Architected and Default Values for Protocol Machines	567
Annex	L	Protocol Tower Encoding	569
	L.1	Protocol Tower Contents	570
Annex	M	The dce_error_inq_text Manual Page	571
		dce_error_inq_text()	572
Annex	N	IDL Data Type Declarations	573
	N.1	Basic Type Declarations	573
	N.2	Status Codes	575
	N.3	RPC-specific Data Types	577
Annex	O	Endpoint Mapper Interface Definition	579
Annex	P	Conversation Manager Interface Definition	583
	P.1	Server Interface	583

	P.2	Client Interface	585
Annex	Q	Remote Management Interface	587
Annex	R	Mapping of RPC Protocol to an OSI Infrastructure	589
	R.1	Abstract syntax name	589
	R.2	Transfer syntax name	589
	R.3	Simple RPC application context	590
	R.3.1	Application context name	590
	R.3.2	Component ASEs	590
	R.3.3	Mapping to supporting services	590

List of Figures

2-1	Information Required to Complete an RPC	20
2-2	Server Binding Relationships	24
2-3	Decisions in Looking Up an Endpoint	27
2-4	Decisions for Selecting a Manager	28
6-1	Execution Phases of an RPC Thread	272
6-2	Concurrent Call Threads Executing in Shared Execution Context	273
10-1	CL_CLIENT Statechart	316
10-2	CL_SERVER Statechart	344
11-1	CO_CLIENT Statechart	382
11-2	CO_CLIENT_ALLOC Statechart	413
11-3	CO_CLIENT_GROUP Statechart	423
11-4	CO_SERVER Statechart	427
11-5	CO_SERVER_GROUP Statechart	459
14-1	NDR Format Label	510
14-2	The Boolean Data Type	512
14-3	Character Data Type	512
14-4	NDR Integer Formats	513
14-5	IEEE Single-precision Floating-point Format	514
14-6	IEEE Double-precision Floating-point Format	515
14-7	VAX Single-precision (F) Floating-point Format	515
14-8	VAX Double-precision (G) Floating-point Format	516
14-9	Cray Floating-point Formats	517
14-10	IBM Floating-point Formats	517
14-11	Uninterpreted Octet Representation	518
14-12	Uni-dimensional Fixed Array Representation	520
14-13	Uni-dimensional Conformant Array Representation	520
14-14	Uni-dimensional Varying Array Representation	520
14-15	Uni-dimensional Conformant and Varying Array Representation	521
14-16	Multi-dimensional Fixed Array Representation	521
14-17	Multi-dimensional Conformant Array Representation	522
14-18	Multi-dimensional Varying Array Representation	522
14-19	Multi-dimensional Conformant and Varying Array Representation	523
14-20	Varying String Representation	523
14-21	Conformant and Varying String Representation	524
14-22	Multi-dimensional Conformant and Varying Array of Strings	525
14-23	Structure Representation	525
14-24	Representation of a Structure Containing a Conformant Array	526
14-25	Representation of a Structure Containing a Conformant and Varying Array	527
14-26	Union Representation	527
14-27	Pipe Representation	528
14-28	Top-level Full Pointer Representation	529
14-29	Top-level Reference Pointer Representation	530
14-30	Embedded Full Pointer Representations	531
14-31	Embedded Reference Pointer Representation	531
14-32	NDR Input Stream	532
14-33	NDR Output Stream	532

List of Tables

3-1	Client and Server Binding Handles	48
3-2	Rules for Returning an Object's Type	169
4-1	Integer Base Types	225
4-2	IDL Directional Attributes	238
4-3	Alphabetic Listing of Productions	250
4-4	Constructed Identifier Classes	253
5-1	Transmitted Type Routines	262
5-2	Transferred Type Routines	263
5-3	Floating Point Error Handling	266
6-1	Execution Semantics	270
6-2	Protocol Tower Structure	278
6-3	The server_name Object Attributes	279
6-4	RPC-specific Protocol Tower Layers	280
6-5	Example Protocol Tower	280
6-6	Service Group Object Attributes	281
6-7	Configuration Profile Object Attributes	282
7-1	Invoke Parameters	286
7-2	Result Parameters	287
7-3	Cancel Parameters	288
7-4	Error Parameters	289
7-5	Reject Parameters	290
8-1	Events Related to Other Elements	297
8-2	Compound Events	297
8-3	Conditions Related to Other Elements	298
8-4	Compound Conditions	298
8-5	Actions Related to Other Elements	298
8-6	Compound Actions	299
12-1	RPC Protocol Data Units	465
12-2	The First Set of PDU Flags	469
12-3	Second Set of PDU Flags	469
12-4	Authentication Protocol Identifiers	472
14-1	NDR Format Label Values	510
14-2	NDR Floating Point Types	513
A-1	UUID Format	534
A-2	UUID version Field	534
A-3	UUID variant Field	534
A-4	The 4 Msb of clock_seq_hi_and_reserved	537
A-5	Field Order and Type	539
B-1	RPC Protocol Sequence Strings	541
C-1	RPC Name Syntax Defined Constants	543
D-1	Casts for Authorisation Information	547
D-2	RPC Key Acquisition for Authentication Services	547
E-1	Reject Status Codes	549
E-2	Failures Returned in a comm_status Parameter	551
E-3	Failures Returned in a fault_status Parameter	551
F-1	IDL/NDR/C Type Mappings	553
F-2	Recommended Boolean Constant Values	554
G-1	Portable Character Set NDR Encodings	559
H-1	Endpoint Mapper Well-known Ports	561
I-1	NDR Transfer Syntax Identifier	563
I-2	Registered Single Octet Protocol Identifiers	563
J-1	DCE CDS Attribute Names	565
K-1	Default Protocol Machine Values	567
L-1	Floors 1 to 3 Inclusive	570
L-2	Floors 4 and 5 for TCP/IP Protocols	570
L-3	Floors 4, 5 and 6 for DECnet Protocol	570

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 11578 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open systems interconnection, data management and open distributed processing*.

Annexes A to R form an integral part of this International Standard.

Information technology - Open Systems Interconnection -

Remote Procedure Call (RPC)

Section 1. General

Using RPC to develop distributed applications is growing in popularity and importance. This International Standard for OSI RPC will meet the growing demand for standardization in this area to ensure future interoperability of distributed applications making use of RPC.

This International Standard facilitates the specification and development of distributed applications based on extending the well-known remote procedure call paradigm to operate between application processes on two separate real open systems in the OSI environment.

The RPC issues addressed by this International Standard reflect the fact that RPC is related both to application programming environments and to communication systems. The modeling of RPC divides naturally into two complementary areas:

- a) describing the RPC paradigm and how it relates to the application programming environments in which it is used, and
- b) defining the communication service used to support RPC in an OSI environment.

This International Standard specifies both portability and interoperability for the Remote Procedure Call (RPC) mechanism. The specification contains material directed at two audiences:

- It provides a portability guide for application programmers.
- It provides both portability and interoperability specifications for those who are implementing or porting RPC or who are testing an RPC implementation.

This International Standard may be thought of as an implementation specification, covering both portability and interoperability, that contains within it an application portability guide. The application portability guide consists of Part 2, RPC Application Programmer's Interface and Part 3, Interface Definition Language and Stubs.

Although the portability specification is part of the broader implementation specification, it has been designed to stand alone so that it may be used by application programmers without reference to the other parts of the implementation specification.

Note: In order to make the portability specification independent, some material is repeated, especially between Section 2 and Section 6.

1.1 Scope

This International Standard

- a) defines the RPC Interaction Model for applications; it describes the styles of interactions supported by the OSI RPC Standard as perceived by application programmers;
- b) defines the RPC Communication Model which describes the RPC components and characteristics within the framework of OSI Application Layer Structure;
- c) defines the requirements met by the standard; and
- d) defines the requirements to be met by other standards for use by RPC and defines the necessary components for the use of these standards.
- e) specifies conformance requirements;
- f) specifies the infrastructure of support environments for RPC such as naming, security and management;
- g) specifies programming language bindings or specific programming language interfaces;

This International Standard

- a) does not specify individual implementations or products;
- b) does not specify local customizations and configurations;
- c) does not specify the interface to local resources or access facilities that are provided within the local system; and
- d) does not provide formal description tools to describe the algorithmic aspects of a distributed application or application semantics.

1.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. 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 7498-1:1994, *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*

ISO 7498-2:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.*

ISO 7498-3:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and addressing.*

ISO 8649:1996, *Information technology - Open Systems Interconnection - Service definition for the Association Control Service Element.*

ISO 8822:1994, *Information technology - Open Systems Interconnection - Presentation service definition.*

ISO/IEC 8824:1990, *Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*

ISO/IEC 9545:1994, *Information technology - Open Systems Interconnection - Application Layer Structure.*

ISO/IEC 10646-1:1993, *Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane.*

ISO/IEC 10731:1994, *Information technology - Open Systems Interconnection - Basic Reference Model - Conventions for the definition of OSI services.*

1.3 Informative References

X/Open P315: April 1996, *X/Open DCE: Authentication and Security Services*.

X/Open C312: December 1994, *X/Open DCE: Directory Services*.

IEEE P1003.4a:1995: *POSIX Threads*.

D. Harel, Science of Computer Programming, 8(3), 231-274, June 1987, *Statecharts: a visual formalism for complex systems*.