

First edition
1998-09-01

Information technology — Generic coding of moving pictures and associated audio information —

Part 6: Extensions for DSM-CC

*Technologies de l'information — Codage générique des images animées
et des informations sonores associées —*

Partie 6: Extensions pour DSM-CC



Reference number
ISO/IEC 13818-6:1998(E)

CONTENTS

0. INTRODUCTION.....	XIX
0.1 Guiding Factors in the Formulation of DSM-CC	xix
0.2 DSM-CC Client-Network-Server Model.....	xx
0.3 Outline of the DSM-CC Specification.....	xx
0.3.1 User-to-Network.....	xx
0.3.2 User-to-User	xxi
0.4 Supported Network Technologies	xxi
0.5 Supported Connection Types	xxi
0.6 DSM-CC Interfaces.....	xxi
0.7 DSM-CC Interface Protocols	xxiii
0.8 Communications Requirements.....	xxv
0.9 Methods of Specification.....	xxv
0.9.1 Messages	xxv
0.9.2 Message Flow Diagram Scenarios.....	xxvi
0.9.3 Specification and Description Language	xxvi
0.9.4 Interface Definition Language (IDL).....	xxviii
0.9.5 Remote Procedure Call (RPC).....	xxviii
0.9.5.1 Independence of RPC.....	xxix
0.9.5.2 Preferred and Default RPC.....	xxix
0.9.5.3 Local Equivalent Functions.....	xxix
1. GENERAL.....	1
1.1 Scope.....	1
1.2 Profiles and Compliance.....	1
1.2.1 Functional Categories of the DSM-CC protocols.....	1
1.2.2 User-to-Network Session Messages	2
1.2.2.1 U-N Core Session Message Functional Groups.....	2
1.2.2.2 U-N Extended Session Message Functional Groups	2
1.2.3 User-User Interfaces.....	2
1.2.3.1 U-U Core Interfaces	2
1.2.3.2 U-U Extended Interfaces	3
1.3 Definitions.....	3
1.4 Acronyms	4
1.5 Normative References	6
2. DSM-CC MESSAGE HEADER.....	7
2.1 DSM-CC Adaptation Header Format.....	8
2.1.1 DSM-CC Conditional Access Adaptation Format.....	9
2.1.2 DSM-CC User ID Adaptation Format.....	9
3. USER-TO-NETWORK CONFIGURATION MESSAGES.....	10
3.1 Overview and the General Message Format.....	10
3.2 User-to-Network configuration parameters	10
3.2.1 DSM-CC specific configuration parameters.....	10
3.2.2 Network specific configuration parameters.....	11
3.2.3 User defined configuration parameters.....	12

© 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

3.3	User to Network Configuration Messages	12
3.3.1	UNConfigRequest message definition	13
3.3.2	UNConfigConfirm message definition	13
3.3.3	UNConfigIndication message definition	14
3.3.4	UNConfigResponse message definition	14
3.4	User-to-Network Configuration Message Field Data Types.....	15
3.5	User Initiated UNConfigRequest message Sequence	15
3.6	Network Initiated UNConfigIndication message Sequence.....	16
3.7	Broadcasting of UNConfigIndication messages	16
3.8	Mixed User/Network Initiated Configuration Sequences	17
3.9	User-to-Network Configuration Reason Codes	17
3.10	User-to-Network Configuration Response Codes.....	17
4.	USER-TO-NETWORK SESSION MESSAGES	19
4.1	Overview and the General Message Format	19
4.2	Session Messages	19
4.2.1	U-N Functional groups	23
4.2.1.1	U-N Core Group	23
4.2.1.2	Extended Functional groups	24
4.2.2	Use of UserData() structure in session messages.....	24
4.2.3	Use of Resources() structure in session messages	25
4.2.4	Session Set-Up group message definitions	25
4.2.4.1	ClientSessionSetUpRequest	25
4.2.4.2	ClientSessionSetUpConfirm.....	26
4.2.4.3	ServerSessionSetUpIndication	27
4.2.4.4	ServerSessionSetUpResponse	27
4.2.5	Session Release group message definitions	28
4.2.5.1	ClientSessionReleaseRequest	28
4.2.5.2	ClientSessionReleaseConfirm.....	29
4.2.5.3	ClientSessionReleaseIndication.....	29
4.2.5.4	ClientSessionReleaseResponse.....	29
4.2.5.5	ServerSessionReleaseRequest	30
4.2.5.6	ServerSessionReleaseConfirm.....	30
4.2.5.7	ServerSessionReleaseIndication	31
4.2.5.8	ServerSessionReleaseResponse	31
4.2.6	Add Resource group message definitions.....	32
4.2.6.1	ClientAddResourceIndication.....	32
4.2.6.2	ClientAddResourceResponse	32
4.2.6.3	ServerAddResourceRequest	33
4.2.6.4	ServerAddResourceConfirm.....	33
4.2.7	Delete Resource group message definitions	34
4.2.7.1	ClientDeleteResourceIndication	34
4.2.7.2	ClientDeleteResourceResponse.....	35
4.2.7.3	ServerDeleteResourceRequest.....	35
4.2.7.4	ServerDeleteResourceConfirm	36
4.2.8	Continuous Feed Session group message definitions	36
4.2.8.1	ServerContinuousFeedSessionRequest.....	36
4.2.8.2	ServerContinuousFeedSessionConfirm	37
4.2.9	Status group message definitions.....	37
4.2.9.1	ClientStatusRequest.....	37
4.2.9.2	ClientStatusConfirm	38
4.2.9.3	ClientStatusIndication	38
4.2.9.4	ClientStatusResponse	38
4.2.9.5	ServerStatusRequest	39
4.2.9.6	ServerStatusConfirm.....	39
4.2.9.7	ServerStatusIndication.....	40
4.2.9.8	ServerStatusResponse.....	40

4.2.10	Reset group message definitions.....	41
4.2.10.1	ClientResetRequest.....	41
4.2.10.2	ClientResetConfirm.....	41
4.2.10.3	ClientResetIndication.....	42
4.2.10.4	ClientResetResponse.....	42
4.2.10.5	ServerResetRequest.....	42
4.2.10.6	ServerResetConfirm.....	43
4.2.10.7	ServerResetIndication.....	43
4.2.10.8	ServerResetResponse.....	43
4.2.11	Session Proceeding group message definitions.....	44
4.2.11.1	ClientSessionProceedingIndication.....	44
4.2.11.2	ServerSessionProceedingIndication.....	44
4.2.12	Connect group message definitions.....	44
4.2.12.1	ClientConnectRequest.....	44
4.2.12.2	ServerConnectIndication.....	45
4.2.13	Session Transfer group message definitions.....	45
4.2.13.1	ClientSessionTransferIndication.....	45
4.2.13.2	ClientSessionTransferResponse.....	46
4.2.13.3	ServerSessionTransferRequest.....	46
4.2.13.4	ServerSessionTransferConfirm.....	47
4.2.13.5	ServerSessionTransferIndication.....	47
4.2.13.6	ServerSessionTransferResponse.....	48
4.2.14	Session In Progress group message definitions.....	49
4.2.14.1	ClientSessionInProgress.....	49
4.2.14.2	ServerSessionInProgress.....	49
4.3	User-to-Network Session Message Field Data Types.....	49
4.4	Reason Codes.....	52
4.5	Response Codes.....	53
4.6	MPEG-2 DSM-CC statusTypes.....	55
4.7	Resource Descriptors.....	56
4.7.1	DSM-CC User-to-Network Resource Descriptor.....	56
4.7.2	Specifying Ranges and Lists of values in resource descriptors.....	62
4.7.3	Horizontal Association of Resource Descriptors.....	64
4.7.4	Vertical Resource Sharing.....	64
4.7.5	Resource Descriptor Definitions.....	65
4.7.5.1	ContinuousFeedSession resource descriptor definition.....	66
4.7.5.2	AtmConnection resource descriptor definition.....	67
4.7.5.3	MpegProgram resource descriptor definition.....	67
4.7.5.4	Physical Channel resource descriptor definition.....	68
4.7.5.5	TSUpstreamBandwidth resource descriptor definition.....	69
4.7.5.6	TSDownstreamBandwidth resource descriptor definition.....	69
4.7.5.7	AtmSvcConnection resource descriptor definition.....	70
4.7.5.8	ConnectionNotify resource descriptor definition.....	70
4.7.5.9	IP resource descriptor definition.....	70
4.7.5.10	ClientTdmaAssignment resource descriptor definition.....	71
4.7.5.11	PSTNSetup resource descriptor definition.....	71
4.7.5.12	NISDNSetup resource descriptor definition.....	71
4.7.5.13	NISDNConnection resource descriptor definition.....	72
4.7.5.14	Q922Connections resource descriptor definition.....	72
4.7.5.15	SharedResource resource descriptor definition.....	72
4.7.5.16	SharedRequestId resource descriptor definition.....	72
4.7.5.17	HeadEndList resource descriptor definition.....	73
4.7.5.18	AtmVcConnection resource descriptor definition.....	73
4.7.5.19	SdbContinuousFeed resource descriptor definition.....	74
4.7.5.20	SdbAssociations resource descriptor definition.....	74
4.7.5.21	SdbEntitlement resource descriptor definition.....	75
4.8	Client Initiated Command Sequences.....	75

4.8.1	Client Session Set-Up Command Sequence	76
4.8.1.1	Client Initiates Session Set-Up Request	76
4.8.1.2	Network Rejects Client Session Request	78
4.8.1.3	Server Rejects Server Session Indication	78
4.8.1.4	Client Has Final UserData()	79
4.8.1.5	Client Initiates Early Release.....	79
4.8.1.6	Server Does not respond to serverSessionSetUpIndication.....	80
4.8.1.7	Network Rejects Server's Resource AllocationStep 7 (Network):	80
4.8.1.8	Client Unable to Use Resources	80
4.8.2	Client Session Release Command Sequence	81
4.8.2.1	Client Initiates Release Request	81
4.8.2.2	Network Rejects Client Release Request.....	82
4.8.2.3	Server Rejects Server Release Indication	82
4.8.3	Client Initiated Status Command Sequence.....	82
4.9	Server Initiated Command Sequences	83
4.9.1	Server Continuous Feed Session Set-Up Command Sequence.....	83
4.9.1.1	Server Initiates Continuous Feed Session Set-Up.....	84
4.9.2	Server Add Resource Command Sequence	84
4.9.2.1	Server Initiates Add Resource Request	85
4.9.3	Server Session Delete Resource Command Sequence.....	86
4.9.4	Server Session Release Command Sequence.....	87
4.9.4.1	Server Initiates Release Request.....	87
4.9.4.2	Network Rejects Server Release Request.....	88
4.9.4.3	Client Rejects Client Release Indication.....	88
4.9.5	Server Continuous Feed Session Release Command Sequence.....	88
4.9.5.1	Server Initiates Continuous Feed Session Release Request.....	89
4.9.5.2	Network Rejects Server Release Request	90
4.9.5.3	Client Rejects Client Release Indication.....	90
4.9.6	Server Status Command Sequence	90
4.9.7	Server Session Forward Command Sequence	91
4.9.7.1	Client Initiates Session Set-Up	92
4.9.7.2	Network Rejects Forward.....	93
4.9.8	Server Session Transfer Command Sequence	93
4.9.8.1	Server A Initiates Session Transfer	94
4.9.8.2	Network Rejects Transfer Request	95
4.9.8.3	Server B Rejects the Transfer Request	95
4.9.8.4	Server B Unable to Allocate Resources for Transfer.....	96
4.9.8.5	Client Rejects Transfer	96
4.9.9	Transferred Session Release.....	96
4.9.9.1	SRM is Selecting sessionIds.....	96
4.9.9.2	Server is Selecting sessionId	96
4.10	Network Initiated Command Sequences.....	97
4.10.1	Network Initiated Session Release Command Sequence.....	98
4.10.1.1	Network Initiates Session Release.....	98
4.10.2	Network Initiated Continuous Feed Session Release Command Sequence.....	98
4.10.2.1	Network Initiates Continuous Feed Session Release.....	99
4.10.3	Network Initiated Client Status Command Sequence	100
4.10.3.1	Network Initiates Client Status command sequence	100
4.10.4	Network Initiated Server Status Command Sequence	101
4.10.4.1	Network Initiates Server Status command sequence	101
4.11	Reset Procedures	101
4.11.1	Client Initiated Reset Command Sequence.....	102
4.11.1.1	Client Initiates Reset command sequence.....	102
4.11.2	Server Initiated Reset Command Sequence.....	102
4.11.2.1	Server Initiates Reset command sequence.....	103
4.11.3	Network Initiated Reset Command Sequence.....	103
4.11.3.1	Network Initiates Reset command sequence	103

5.	USER-TO-USER INTERFACES.....	105
5.1	Introduction	105
5.1.1	Contents.....	105
5.1.2	Intended Usage.....	105
5.2	The User-to-User System Environment.....	107
5.2.1	U-U System Hardware User Entities	107
5.2.2	U-U System Logical Entities	107
5.2.3	Application and Service Interfaces	109
5.2.4	Categorization of Client Library Interface Sets	110
5.2.4.1	Consumer Client	110
5.2.4.2	Producer Client.....	111
5.2.4.3	Client Library Profiles	111
5.2.5	Core Interfaces	112
5.2.5.1	Core Client Application Portability Library	113
5.2.5.2	Core Client Service Inter-operability Library	115
5.2.6	Extended Interfaces	116
5.2.6.1	Extended Client Application Portability Library	117
5.2.6.2	Extended Client Service-interoperability Library	118
5.3	Overview of the Interface Definition Language(IDL)	118
5.3.1	Operations	120
5.3.2	Attributes.....	121
5.3.3	Language Mapping	121
5.3.4	Encoding.....	121
5.3.5	Typographical Conventions.....	122
5.3.6	Syntactical Conventions	122
5.4	Common Definitions	122
5.4.1	Basic Types	122
5.4.2	Entity Identification.....	123
5.4.3	Interface Identification	124
5.4.4	Access Roles for Operations.....	126
5.4.4.1	Syntax for Access Control	126
5.4.5	Exceptions	127
5.4.6	Stream and Event Synchronization	130
5.5	Application Portability Interfaces(API).....	131
5.5.1	Core Interfaces	131
5.5.1.1	Base	132
5.5.1.1.1	Summary of Base Primitives	132
5.5.1.1.2	DSM Base close	133
5.5.1.1.3	DSM Base destroy.....	133
5.5.1.2	Access.....	133
5.5.1.2.1	Setting Permissions.....	134
5.5.1.2.2	Access Definitions.....	135
5.5.1.3	Stream.....	136
5.5.1.3.1	Stream Definitions, Exceptions	137
5.5.1.3.2	Normal Play Time Temporal Positioning	137
5.5.1.3.2.1	Application NPT Values	138
5.5.1.3.3	Summary of Stream Primitives	138
5.5.1.3.4	Stream State Machine	139
5.5.1.3.4.1	State Machine	139
5.5.1.3.4.2	Basic State Machine	141
5.5.1.3.4.3	Complete state machine.....	142
5.5.1.3.5	DSM Stream pause	143
5.5.1.3.6	DSM Stream resume.....	144
5.5.1.3.7	DSM Stream status	145
5.5.1.3.8	DSM Stream reset.....	146
5.5.1.3.9	DSM Stream jump.....	146

5.5.1.5.10	DSM Stream play	147
5.5.1.4	File.....	148
5.5.1.4.1	File Definitions, Exceptions	148
5.5.1.4.2	Summary of File Primitives	149
5.5.1.4.3	DSM File read	149
5.5.1.4.4	DSM File write.....	150
5.5.1.5	Directory.....	151
5.5.1.5.1	Directory Definitions, Exceptions	152
5.5.1.5.2	Summary of Directory Primitives	154
5.5.1.5.3	DSM Directory list	155
5.5.1.5.4	DSM Directory resolve.....	156
5.5.1.5.5	DSM Directory bind.....	156
5.5.1.5.6	DSM Directory bind_context	157
5.5.1.5.7	rebind.....	158
5.5.1.5.8	DSM Directory rebind_context	158
5.5.1.5.9	DSM Directory unbind	159
5.5.1.5.10	DSM Directory new_context.....	160
5.5.1.5.11	DSM Directory bind_new_context.....	160
5.5.1.5.12	DSM Directory destroy	161
5.5.1.5.13	DSM Directory open	161
5.5.1.5.14	DSM Directory close.....	162
5.5.1.5.15	DSM Directory get	163
5.5.1.5.16	DSM Directory put.....	164
5.5.1.6	Session.....	165
5.5.1.6.1	Service Transfer	165
5.5.1.6.2	Summary of Session Primitives	166
5.5.1.6.3	DSM Session attach.....	166
5.5.1.6.4	DSM Session detach.....	167
5.5.1.7	ServiceGateway	168
5.5.1.7.1	Summary of ServiceGateway Primitives	168
5.5.1.8	First.....	168
5.5.1.8.1	Summary of First Primitives	168
5.5.1.8.2	DSM First root	169
5.5.1.8.3	DSM First service.....	169
5.5.2	Extended Interfaces	169
5.5.2.1	Download	171
5.5.2.1.1	Download Definitions, Exceptions	171
5.5.2.1.2	Summary of Download Primitives.....	171
5.5.2.1.3	DSM Download info	171
5.5.2.1.4	DSM Download alloc	172
5.5.2.1.5	DSM Download start.....	172
5.5.2.1.6	DSM Download cancel.....	173
5.5.2.2	Event.....	173
5.5.2.2.1	Event Definitions, Exceptions	174
5.5.2.2.2	Summary of Event Primitives	174
5.5.2.2.3	DSM Event subscribe	174
5.5.2.2.4	DSM Event unsubscribe	175
5.5.2.2.5	DSM Event notify.....	176
5.5.2.3	Composite.....	177
5.5.2.3.1	Summary of Composite Primitives	177
5.5.2.3.2	DSM Composite list_subs	177
5.5.2.3.3	DSM Composite bind_subs	178
5.5.2.3.4	DSM Composite unbind_subs	179
5.5.2.4	View	179
5.5.2.4.1	Non-Database View.....	180
5.5.2.4.2	Database View.....	180
5.5.2.4.3	View Procedures.....	180

5.5.2.4.4	Definition: View Style Attribute.....	181
5.5.2.4.5	View Definitions: Statement, Result.....	182
5.5.2.4.6	Summary of View Primitives.....	182
5.5.2.4.7	DSM View query.....	183
5.5.2.4.8	DSM View read.....	184
5.5.2.4.9	DSM View execute.....	185
5.5.2.5	State.....	185
5.5.2.5.1	Summary of State Primitives	185
5.5.2.5.2	DSM State suspend.....	186
5.5.2.5.3	DSM State resume.....	187
5.5.2.6	Interfaces	187
5.5.2.6.1	TCKind Constants	188
5.5.2.6.2	Exception TCKind Constants	189
5.5.2.6.3	Interfaces Definitions	190
5.5.2.6.4	Summary of Interfaces Primitives.....	190
5.5.2.6.5	DSM Interfaces show	190
5.5.2.6.6	DSM Interfaces define.....	192
5.5.2.6.7	DSM Interfaces check	193
5.5.2.6.8	DSM Interfaces undefine.....	194
5.5.2.7	Security.....	194
5.5.2.7.1	DSM Security authenticate.....	194
5.5.2.8	Config.....	195
5.5.2.8.1	Config Definitions	197
5.5.2.8.2	Summary of Config Primitives	197
5.5.2.8.3	DSM Config inquire.....	197
5.5.2.8.4	DSM Config wait.....	197
5.5.2.9	LifeCycle	198
5.5.2.9.1	DSM LifeCycle create.....	198
5.5.2.10	Kind.....	199
5.5.2.10.1	Summary of Kind Primitives	199
5.5.2.10.2	DSM_Kind_has_a	199
5.5.2.10.3	DSM_Kind_is_a.....	200
5.5.3	C Language Mappings.....	201
5.5.3.1	Scoped Identifiers.....	201
5.5.3.2	C Mapping for Operations.....	201
5.5.3.2.1	C Mapping for Basic Data Types	202
5.5.3.2.2	Constants	202
5.5.3.2.3	Struct Types.....	202
5.5.3.2.4	Sequence Types.....	202
5.5.3.2.5	Strings.....	203
5.5.3.2.6	Any.....	203
5.5.3.2.7	ev.....	203
5.5.3.2.8	Object.....	203
5.5.3.3	API Definitions.....	204
5.5.3.3.1	C Mapping for the Synchronous Interface.....	204
5.5.3.3.1.1	Base	204
5.5.3.3.1.2	Access.....	204
5.5.3.3.1.3	Stream.....	205
5.5.3.3.1.4	File.....	206
5.5.3.3.1.5	Directory.....	206
5.5.3.3.1.6	Session.....	207
5.5.3.3.1.7	First	207
5.5.3.3.1.8	Event	208
5.5.3.3.1.9	Download	208
5.5.3.3.1.10	Composite.....	208
5.5.3.3.1.11	View	209
5.5.3.3.1.12	State.....	209

5.5.3.3.1.13	Interfaces	210
5.5.3.3.1.14	Security.....	210
5.5.3.3.1.15	LifeCycle.....	210
5.5.3.3.1.16	Kind.....	210
5.5.3.3.2	C Mapping for the Synchronous Deferred Interface.....	211
5.5.3.3.2.1	Config.....	211
5.5.3.3.2.2	How to Convert Synchronous to Synchronous Deferred	211
5.6	Service Interoperability Interfaces(SII)	211
5.6.1	ConnBinder and Resource to Connection Association	212
5.6.1.1	Selector.....	214
5.6.2	Remote Procedure Call.....	214
5.6.3	The Object Reference	214
5.6.3.1	Min Protocol Profile.....	215
5.6.3.2	Child Protocol Profile.....	216
5.6.3.3	Options Protocol Profile.....	216
5.6.3.4	Lite Protocol Profiles.....	217
5.6.3.5	BIOP Protocol Profile	217
5.6.3.6	ONC Protocol Profile	218
5.6.4	ServiceContextList	218
5.6.4.1	ServiceContext	219
5.6.5	Core Interfaces	220
5.6.5.1	Base.....	220
5.6.5.2	Access.....	220
5.6.5.3	Stream.....	222
5.6.5.3.1	Transport and Application Level NPT	224
5.6.5.3.2	Consistent Quantization Rules.....	224
5.6.5.4	File.....	224
5.6.5.5	BindingIterator	225
5.6.5.6	NamingContext.....	226
5.6.5.7	Directory.....	227
5.6.6	Extended Interfaces	229
5.6.6.1	SessionUU.....	229
5.6.6.1.1	Partial Path	230
5.6.6.2	ServiceGatewayUU	230
5.6.6.2.1	Summary of ServiceGatewayUU Primitives.....	230
5.6.6.3	SessionSI	231
5.6.6.4	ServiceGatewaySI	231
5.6.6.4.1	Summary of ServiceGatewaySI Primitives.....	232
5.6.6.5	DownloadSI.....	232
5.6.6.6	Event.....	235
5.6.6.7	Composite.....	236
5.6.6.8	View	238
5.6.6.9	State.....	240
5.6.6.10	Interfaces	241
5.7	Application Boot Process	241
5.7.1	Session attach() Pre-conditions	242
5.7.2	Session attach() Procedure	243
5.7.2.1	Resolving Path-specific Parameters.....	243
5.7.2.1.1	Post-condition.....	244
5.7.2.2	Establishing the U-N Session	244
5.7.2.2.1	ClientSessionSetupRequest	244
5.7.2.2.2	ClientSessionSetupConfirm.....	245
5.7.2.2.3	Session Establishment Post-conditions	246
5.7.2.3	Download	246
5.7.3	Session Tear-down	248
5.7.4	Session Transfer Implications.....	248

6.	USER COMPATIBILITY	249
6.1	Compatibility Descriptors.....	249
6.1.1	IEEE OUI Specifier.....	251
7.	USER-TO-NETWORK DOWNLOAD	252
7.1	Overview	252
7.1.1	Download Network Models.....	253
7.1.2	Preconditions and Assumptions.....	254
7.2	Download Message Set	255
7.2.1	Download Control Message Format.....	255
7.2.2	Download Data Message Format.....	255
7.2.2.1	DSM-CC Download Data Header	255
7.3	Message Descriptions.....	256
7.3.1	DownloadInfoRequest.....	257
7.3.2	DownloadInfoResponse and DownloadInfoIndication.....	257
7.3.3	DownloadDataBlock	259
7.3.4	DownloadDataRequest	259
7.3.5	DownloadCancel	260
7.3.6	DownloadServerInitiate.....	263
7.4	Message Sequence for Flow-Controlled Download Scenario.....	263
7.4.1	Getting Download Protocol Parameters	264
7.4.2	Starting Download	265
7.4.3	Acknowledgments	265
7.4.4	Timers and Re-transmission	266
7.4.5	Abort	267
7.4.6	Flow-Controlled Scenario over Reliable Transport.....	267
7.5	Message Sequence for Data Carousel Scenario.....	267
7.5.1	Getting Data Carousel Parameters.....	267
7.5.2	Starting Acquisition and Module Re-Assembly.....	268
7.5.2.1	Pseudo-Code Example of Module Re-assembly.....	268
7.5.3	Timers.....	269
7.5.4	Module Coherency	270
7.5.5	Data Delivery Rate	270
7.6	Message Sequence for Non-Flow-Controlled Download Scenario	270
7.6.1	Getting Download Protocol Parameters	271
7.6.2	Image Assembly and Coherency.....	271
7.6.3	Timers.....	271
7.7	Protocol State Machines for flow-controlled download scenario	271
7.7.1	State Variables common to Client and Download Server.....	272
7.7.1.1	Service Type: reliableService, unreliableService	272
7.7.1.2	Download configured bufferSize: bufferSize	272
7.7.1.3	Download configured maximumBlockSize: blockSize	272
7.7.1.4	Download Identifier: Did.....	272
7.7.1.5	Download negotiated blockSize: Did.blockSize.....	272
7.7.1.6	Download negotiated windowSize: Did.windowSize.	272
7.7.1.7	Download negotiated Acknowledgment Period: Did.ackPeriod.....	272
7.7.1.8	Download negotiated Window Timer: Did.tWindow.....	272
7.7.1.9	Download negotiated Scenario Timer: Did.tScenario	272
7.7.1.10	Download negotiated compatibilities: Did.compatibilities.....	272
7.7.1.11	Download Number of Modules: Did.numModules	272
7.7.1.12	Download Module Identifier: Did.moduleId	272
7.7.1.13	Download Module Version: Did.moduleId.version.....	273
7.7.1.14	Download Module Size: Did.moduleId.moduleSize	273
7.7.1.15	Download Expired downloadId Holding timer: Did.tHold	273
7.7.2	Client-only State Variables.....	273
7.7.2.1	Download Lower Receive Window Edge: Did.NmoduleId, Did.NblockNum.	273

7.7.2.2	Number received blocks: Did.Nblock	273
7.7.2.3	Acknowledgment threshold: Did.AckThreshold.....	273
7.7.3	Server-only State Variables	273
7.7.3.1	Lower Transmit Window Edge: Did.LmoduleId, Did.LblockNum	273
7.7.3.2	Upper Transmit Window Edge: Did.UmoduleId, Did.UblockNum	273
7.7.3.3	Data Sending Rate Timer: Did.tSend	273
7.7.4	Client Conditions	273
7.7.4.1	Invalid ServerId.....	273
7.7.4.2	Number of re-transmission exceeded.....	274
7.7.4.3	Unacceptable blockSize.....	274
7.7.4.4	Unacceptable WindowSize.....	274
7.7.4.5	Unacceptable Acknowledgment Period	274
7.7.4.6	Unacceptable Window Timer	274
7.7.4.7	Unacceptable Scenario Timer.....	274
7.7.4.8	Unacceptable Compatibilities	274
7.7.4.9	Unacceptable Module Table.....	274
7.7.4.10	Acknowledgment period full	274
7.7.4.11	Download complete.....	274
7.7.5	Download Server Conditions.....	274
7.7.5.1	Unacceptable maximumBlockSize	274
7.7.5.2	Unacceptable bufferSize.....	275
7.7.5.3	Unacceptable Compatibilities	275
7.7.6	Client Procedures	275
7.7.6.1	Initial Setup of State Variables	275
7.7.6.2	Sending DownloadDataRequest Messages	275
7.7.6.3	Sending DownloadCancel Messages	275
7.7.6.4	Increment Lower Receive Window Edge	275
7.7.6.5	Increment block counter	276
7.7.6.6	Transition to DCExpire State	276
7.7.7	Download Server Procedures	276
7.7.7.1	Initial Setup of State Variables	276
7.7.7.2	Increment Lower Transmit Window Edge.....	277
7.7.7.3	Set Upper Transmit Window Edge.....	277
7.7.7.4	Sending DownloadDataBlock Messages	277
7.7.7.5	Sending DownloadCancel Messages	277
7.7.7.6	Transition to DSExpire State.....	277
7.7.8	State Machine SDL.....	277
7.8	Partial Protocol State Machines for non-flow-controlled download scenario	277
8.	STREAM DESCRIPTORS.....	279
8.1	Normal Play Time	279
8.1.1	NPT Reference Descriptor	279
8.1.2	Reconstruction of NPT	280
8.1.3	NPT Conversion to Seconds and Microseconds.....	281
8.1.4	NPT Uncertainty.....	281
8.1.4.1	Frequency of NPT Reference Descriptor.....	281
8.1.5	NPT Endpoint Descriptor	282
8.2	Stream Mode Descriptor	282
8.3	Stream Event Descriptor.....	283
9.	TRANSPORT	284
9.1	DSM-CC Requirements on Lower-Level Network Transport Protocol.....	284
9.1.1	U-N Message Categories	284
9.1.2	U-U Interface Categories.....	284
9.2	Encapsulation within MPEG-2 Transport Streams	285
9.2.1	Role of MPEG-2 Transport Stream in the Protocol Stack.....	285
9.2.2	DSM-CC Sections	285

9.2.2.1	Semantic definition of fields in DSMCC_section.....	286
9.2.3	DSM-CC Stream Types.....	288
9.2.4	DSM-CC Multi-protocol Encapsulation.....	288
9.2.5	U-N Message Categories.....	289
9.2.6	U-U Service Inter-operability Interface using Remote Procedure Call.....	289
9.2.7	DSM-CC Stream Descriptors.....	289
9.2.7.1	Semantic definition of fields in DSM-CC Descriptor List.....	289
9.3	Encapsulation within MPEG-2 Program Streams.....	289
9.3.1	DSM-CC Stream Descriptors.....	289
9.3.1.1	Semantic definition of fields in DSM-CC_program_stream_Descriptor List.....	290
9.3.2	U-N Messages and U-U SSI.....	290
10.	U-N SWITCHED DIGITAL BROADCAST -- CHANNEL CHANGE PROTOCOL.....	291
10.1	Overview.....	291
10.1.1	Preconditions and Assumptions.....	291
10.1.2	General Message Format.....	291
10.2	Switched Digital Broadcast Channel Change Protocol Messages.....	291
10.2.1	Use of Private Data in SDB CCP messages.....	292
10.2.2	Use of BroadcastProgramId in SDB CCP messages.....	292
10.2.3	SDB CCP message definitions.....	292
10.2.3.1	SDBProgramSelectRequest message definition.....	293
10.2.3.2	SDBProgramSelectConfirm message definition.....	293
10.2.3.3	SDBProgramSelectIndication message definition.....	294
10.2.3.4	SDBProgramSelectResponse message definition.....	294
10.3	SDB Channel Change Protocol Command Scenarios.....	294
10.3.1	Client Initiated Program Select Command Sequence.....	294
10.3.2	SDB Server Initiated Program Select Command Sequence.....	296
10.4	SDB Reason and Response Codes.....	297
10.4.1	SDB Reason Codes.....	297
10.4.2	SDB Response Codes.....	298
10.5	SDB State Machine.....	298
10.5.1	SDB State Machine for the Client Side.....	298
10.5.2	State machine for the SDB Server Side.....	300
11.	U-U OBJECT CAROUSEL.....	303
11.1	Introduction.....	303
11.2	Concepts.....	304
11.2.1	Supported U-U Objects and Interfaces.....	304
11.2.2	Service Domain and Service Gateway.....	304
11.2.3	Object References.....	305
11.2.4	Transport of BIOP Messages.....	305
11.2.5	Module Delivery Parameters.....	306
11.2.6	Taps.....	306
11.3	Broadcast Inter ORB Protocol.....	307
11.3.1	Inter-operable Object Reference (IOR).....	307
11.3.1.1	Profile Body Definition.....	307
11.3.1.1.1	Object Location Component.....	307
11.3.1.1.2	ConnBinder Component.....	307
11.3.2	Message Set Definition.....	308
11.3.2.1	Generic Object Message Format.....	308
11.3.2.2	Directory Message Format.....	310
11.3.2.3	File Message Format.....	311
11.3.2.4	Stream Message Format.....	311
11.3.2.5	Service Gateway Message Format.....	313
11.3.3	Transport Definitions.....	313
11.3.3.1	BIOP Messages.....	313
11.3.3.2	Module Delivery Parameters.....	313

11.3.3.3	IOR of Service Gateway.....	314
11.4	MPEG-2 Descriptors	315
11.4.1	Carousel identifier descriptor	316
11.4.2	Association tag descriptor	316
11.4.3	Deferred association tags descriptor.....	318
12.	USER-TO-NETWORK PASS-THRU MESSAGES.....	319
12.1	Overview and the General Message Format	319
12.2	Pass-Thru Messages	319
12.2.1	Use of PassThruData() structure in Pass-Thru messages.....	320
12.2.2	Pass-Thru message definitions.....	321
12.2.2.1	PassThruRequest	321
12.2.2.2	PassThruIndication	321
12.2.2.3	PassThruReceiptRequest	321
12.2.2.4	PassThruReceiptConfirm.....	322
12.2.2.5	PassThruReceiptIndication.....	322
12.2.2.6	PassThruReceiptResponse.....	323
12.3	User-to-Network Pass-Thru Message Field Data Types.....	323
12.4	Pass-Thru Message Scenario	324
12.4.1	Pass-Thru Message scenario.....	324
12.4.1.1	The Sending User sends a PassThruRequest	324
12.5	Pass-Thru Receipt Message Scenario	324
12.5.1	Pass-Thru Receipt Message scenario.....	325
12.5.1.1	The Sending User sends a PassThruReceiptRequest	325
12.6	Pass-Thru Response Codes.....	326
12.7	Pass-Thru Type Codes.....	326
12.8	State Machine	326
ANNEX A (NORMATIVE) USER-NETWORK PROTOCOL STATE MACHINES.....		327
A.1	Introduction	327
A.2	U-N Session.....	327
A.3	U-N Download – Flow Controlled Scenario	364
A.4	U-N Switched Digital Broadcast Channel Change Protocol.....	377
A.5	U-N Pass-Thru.....	390
ANNEX B (INFORMATIVE) APPLICATION EXAMPLES		399
B.1	Introduction	399
B.2	Video Stream Play.....	399
B.3	Building a Directory Hierarchy	401
B.4	Movie Information Database	402
B.5	View as a Personalized Directory.....	408
ANNEX C (INFORMATIVE) ONC RPC XDR MAPPINGS.....		410
C.1	Overview	410
C.2	General RPC Message Formats	410
C.3	CORBA IDL C to XDR Mapping	412
C.3.1	Mapping for Integer Data Types	412
C.3.2	Mapping for void.....	412
C.3.3	Mapping for Constants	413
C.3.4	Mapping for octet	413
C.3.5	Mapping for Fixed-length Constructed Types	413
C.3.5.1	Mapping for struct	413
C.3.6	Mapping for sequences.....	413
C.3.6.1	Example: Mapping for opaque	413
C.3.6.2	Example: Mapping for PathSpec	414
C.3.7	Mapping for string.....	415
C.4	DSM-CC ONC Protocol Profile for the Interoperable Object Reference.....	415

C.5	Exceptions	416
C.6	Request and Reply Header Structures	417
C.7	DSM-CC RPC Program Numbers	418
C.7.1	RPC Program Dispatch Tables Mapping.....	418
ANNEX D (INFORMATIVE) USING DSM-CC U-N SESSION MESSAGES WITH ATM.....		421
D.1	Methods of using DSM-CC over ATM	421
D.1.1	Session Method	421
D.1.2	Network Method with AddResource messages between the Server and the SRM	421
D.1.3	Network Method with NO AddResource messages between the Server and the SRM.....	421
D.1.4	Integrated Method	422
D.2	Association of DSM-CC connection resources to ATM SVCs	423
D.2.1	DSM-CC resourceId Mapping into Q.2931.....	423
D.3	Session Method Command Sequences	424
D.3.1	Session Set-Up	424
D.3.1.1	Client Session Set-Up.....	425
D.3.2	Add Resource Request	428
D.3.2.1	Add Resource Request by the Server	429
D.3.3	Resource Deletion	430
D.3.3.1	Resource Deletion by the Server	431
D.3.4	Session Tear-Down	433
D.3.4.1	Session Tear-Down by Server	434
D.3.4.2	Session Tear-Down by Client.....	436
D.4	Network Method with DSM-CC AddResource messages between the Server and SRM	437
D.4.1	Session Set-Up	437
D.4.1.1	Client Session Set-Up, Server ATM Connection Set-Up	437
D.4.2	Add Resource Request	440
D.4.2.1	Add Resource Request by Server and ATM SVC Connection Set-Up by Server	440
D.4.3	Resource Deletion	442
D.4.3.1	Resource Deletion Request by Server and ATM SVC Connection Release by Server....	442
D.4.4	Session Tear-Down	443
D.4.4.1	Session Tear-Down Request by Server and ATM SVC Connection Release by Client ..	443
D.4.4.2	Session Tear-Down Request by Client and ATM SVC Connection Release by Server ..	444
D.4.4.3	Session Tear-Down Request by Server and ATM SVC Connection Release by Server..	445
D.5	Network Method with NO DSM-CC AddResource messages between the Server and SRM	446
D.5.1	Session Set-Up	446
D.5.1.1	Client Session Set-Up.....	447
D.5.2	Add Resource Request	447
D.5.2.1	Add Resource Request by the Server	447
D.5.3	Connection Clearing	449
D.5.3.1	Connection Clearing by the Server.....	449
D.5.3.2	Connection Clearing by the Client.....	451
D.5.4	Session Tear-Down	451
D.5.4.1	Session Tear-Down by Server	451
D.5.4.2	Session Tear-Down by Client.....	452
D.6	Integrated Method Command Sequences	453
D.6.1	Session Set-Up	454
D.6.1.1	Client Session Set-Up.....	454
D.6.1.2	Server Session Set-Up	455
D.6.2	Integrated Method for Adding Resources.....	455
D.6.3	Connection Clearing	456
D.6.4	Session Tear-Down	456
D.6.4.1	Server Session Tear-Down	457
D.6.4.2	Client Session Tear Down	457
D.7	References	457
ANNEX E (INFORMATIVE) UNO INTER-OPERABLE RPC PROTOCOL STACK.....		459

E.1	Abstract	459
E.2	Motivation	459
E.3	Solution Space	459
E.4	Inter-operation Framework	460
E.5	Protocol Selection	461
E.6	Common Data Representation	461
E.6.1	Encapsulation	462
E.6.2	Alignment	462
E.6.3	Primitive Data Types	462
E.6.4	Compound Types	463
E.6.5	TypeCode	463
E.7	UNO Session Protocol	464
E.7.1	Message Set	464
E.7.1.1	Request Message	464
E.7.1.2	Reply	465
E.7.1.3	CancelRequest	466
E.7.1.4	LocateRequest	466
E.7.1.5	LocateReply	466
E.7.1.6	CloseConnection	466
E.7.1.7	MessageError	466
E.7.2	Session Semantics	466
E.8	Transport and Network Semantics	467
ANNEX F (INFORMATIVE) USE OF U-U OBJECT CAROUSEL		468
F.1	Introduction	468
F.2	Purpose of U-U Object Carousels	468
F.3	IDL structures	468
F.3.1	Inter-operable object Reference	468
F.3.2	Generic object Message	470
F.3.3	Directory Message	470
F.4	Support for New Object Representations	471
F.5	How to resolve an object from its IOR	472
F.6	Service Gateway and Download support	474
F.7	U-U Object Carousels on top of MPEG-2 TS Broadcast Networks	475
ANNEX G (INFORMATIVE) SHARED RESOURCES AND THE ASSOCIATION TAG		477
G.1	Introduction	477
G.2	Use of the Association Tag	477
G.3	Use of the SharedResource Descriptor	478
G.4	Use of the SharedRequestId Descriptor	478
G.5	Common Examples of Use	479
G.5.1	Download Phase, Multiple ATM SVCs	479
G.5.1.1	End-to-End ATM	479
G.5.1.2	Non-ATM HFC Client View	480
G.5.2	Video Play Phase, Multiple ATM SVCs	481
G.5.2.1	End-to-End ATM	481
G.5.2.2	Non-ATM HFC Client View	482
G.5.3	Single Asymmetric ATM SVC	483
G.5.3.1	End-to-End ATM	483
G.5.3.2	Non-ATM HFC Client View	484
G.5.4	Single Asymmetric ATM PVC	486
G.5.5	Download Phase, Multiple ATM PVCs	486
G.5.6	Video Play Phase, Multiple ATM PVCs	486
G.5.7	Use of sharedResourceRequest Descriptors	486
ANNEX H (INFORMATIVE) SWITCHED DIGITAL BROADCAST SERVICE		487
H.1	Introduction	487

H.2	Switched Digital Broadcast Service	487
H.3	Functional Flows	488
H.3.1	Broadcast Program Configuration	488
H.3.2	Client Service Profile Transfer to the SDB Server	489
H.3.3	Broadcast Program Guide Transfer to Client	490
H.3.4	Switched Digital Broadcast Service Session Establishment	490
H.3.5	Client Initiated Channel Changes	492
H.3.6	Network Initiated Channel Changes	493
H.3.7	Digital Broadcast Session Release.....	495
ANNEX I (INFORMATIVE) EXAMPLE U-N LIFE CYCLE WALK THROUGH.....		497
I.1	Introduction	497
I.2	General Flow	497
I.3	U-N-Configuration	498
I.3.1	Pre Conditions	498
I.3.2	Procedure.....	498
I.3.3	Post Conditions	498
I.4	U-N Session Setup.....	499
I.4.1	Pre Conditions	499
I.4.2	Procedure.....	499
I.4.3	Post Conditions	502
I.5	U-N Download	503
I.5.1	Pre Conditions	503
I.5.2	Procedure.....	504
I.5.3	Post Conditions	504
ANNEX J (INFORMATIVE) EXAMPLE OF AN OSI NSAP ADDRESS FORMAT		505
J.1	Purpose.....	505
J.2	Introduction	505
J.3	E.164 NSAP	505
ANNEX K (INFORMATIVE) STREAM PLAYLIST.....		507
K.1	Overview	507
K.2	DSM QStream next	509
ANNEX L (INFORMATIVE) SERVICE TRANSFER MESSAGE FLOWS		510
L.1	Introduction	510
L.1.1	Use of service transfer in the normal course of service	510
L.1.2	Use of Service Transfer in emergency cases	511
L.2	Basic application level Service Transfer	511
L.2.1	Service Transfer: sourceServer to destinationServer with sourceServer Session Release ...	511
L.2.2	Service Transfer: sourceServer to destinationServer, Service maintained on sourceServer	512
L.3	Enhanced application level Service Transfer.....	513
L.3.1	Release the Session with the sourceServer	513
L.3.2	Maintain minimum resources with the sourceServer	514
L.3.3	Maintain the service with the sourceServer	515
L.3.4	Fall back to Server A after Session release with the sourceServer	515
L.3.5	Resumption of the full context on Server A after reduced Session.....	515
L.3.6	Emergency Service Transfer.....	516
ANNEX M (INFORMATIVE) T.120 INTER-WORKING.....		519
M.1	Introduction	519
M.2	Reference Model for side-by-side integrated DSM-CC/T.120	519
M.3	Features, Functions and Services of the DSM-CC and the T.120 specifications.....	521
M.3.1	Features, Functions and Services of DSM-CC	521
M.3.2	Features, Functions and Services of T.120	521

This is a preview of "ISO/IEC 13818-6:1998". [Click here to purchase the full version from the ANSI store.](#)

M.3.3	Inter-working of DSM-CC and T.120 Features, Functions and Services	521
M.4	DSM-CC and T.120 Components Harmonized	522
M.5	Specifics for inter-operation between DSM-CC and T.120.....	523
M.5.1	Terminal 1 creates a conference	524
M.5.2	Terminal 2 queries a conference.....	524
M.5.3	Terminal 2 joins the conference	524
M.6	T.120 service within DSM-CC	536
M.6.1	An Example of Extending DSM-UU to provide custom interfaces	536
ANNEX N (INFORMATIVE) THE RELATION OF DSM-CC TO MHEG-5		539
N.1	Overview	539
N.2	Name Space	539
N.2.1	MHEG Object References	540
N.2.2	Content References	540
N.3	Stream Events and Normal Play Time.....	540
N.4	Example of DSM-CC file structure for an application	541
N.5	Example of Mapping High-Level API Actions on DSM-CC U-U Primitives	542

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 13818-6 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 13818 consists of the following parts, under the general title *Information technology — Generic coding of moving pictures and associated audio information*:

- *Part 1: Systems*
- *Part 2: Video*
- *Part 3: Audio*
- *Part 4: Compliance testing*
- *Part 5: Software simulation*
- *Part 6: Extensions for DSM-CC*
- *Part 7: Advanced Audio Coding (AAC)*
- *Part 9: Extension for real time interface for systems decoders*
- *Part 10: Conformance extensions for DSM-CC*

Annex A forms an integral part of this part of ISO/IEC 13818. Annexes B to N are for information only.

0. Introduction

The Digital Storage Media Command and Control (DSM-CC) specification is an integral part of ISO/IEC 13818 (MPEG-2). It consists of a modular set of protocols that may be combined or used individually to provide a wide range of functionality which may be used to support emerging multimedia technologies.

The concepts and protocols of DSM-CC provide the general capability to browse, select, download, and control a variety of bit stream types. DSM-CC also provides a mechanism to manage network and application resources through the concept of a "session". A Session is an associated collection of resources required to deliver a Service. Examples of resources are MPEG-2 Transport Stream packet identifiers and network bandwidth. The Session complements a "Service Domain", which is a collection of interfaces to browse and select services, and control the delivery of bit streams.

One of the strengths of DSM-CC is in its abstraction from underlying networks; a suite of uniform interfaces are visible to the application, shielding it from the details of inter-working among heterogeneous networks – e.g., Hybrid Fiber Coax (HFC), Asynchronous Transfer Mode (ATM), Asymmetric Digital Subscriber Loop (ADSL), Internet Protocol (IP), and combinations of these technologies as part of an end-to-end multimedia system. In other words, a server may simultaneously and uniformly interact through a single network interface with clients connected to different network types, without requiring a separate network interface to each client.

The session signaling layer provides a uniform, flexible, and extensible method for managing heterogeneous resource types. In addition to the network and service types described in this specification, DSM-CC may be extended to support other networks and services through the definition of new resource types.

In DSM-CC, a bit stream is sourced by a Server and delivered to a Client. Both the Client and the Server are logical embodiments and do not imply a singular device in an actual implementation.

Application/service examples are interactive multimedia retrieval (including video-on-demand), Internet access, digital video broadcasting, data downloading, and audio/video/graphics conferencing.

0.1 Guiding Factors in the Formulation of DSM-CC

The DSM-CC specification was influenced by the following factors:

- A wide range of network topologies may be used to deliver DSM-CC.
- Resources are finite and need to be managed.
- Latencies need to be minimized to provide (interactive) services.
- DSM-CC applications need to be supported by an underlying protocol that facilitates communications between a Server-side application and a corresponding Client-side application.

0.2 DSM-CC Client-Network-Server Model

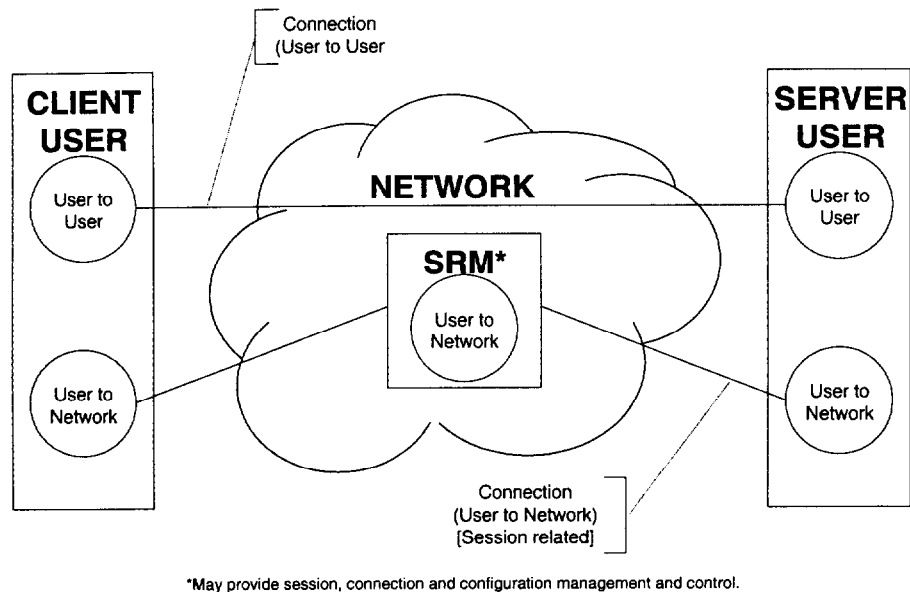


Figure 0-1 DSM-CC basic Client-Network-Server model

Figure 0-1 depicts the basic model used in DSM-CC. A Session and Resource Manager (SRM) provides logically centralized management of the DSM-CC Sessions and Resources. DSM-CC User-to-Network (U-N) messages flow between the Client and SRM and the Server and SRM. Both the Client and the Server are called Users of DSM-CC. The U-N session protocol establishes a Session and groups all the resources required for delivering a service. The service interactions are carried between the Client and the Server participating in the Session using DSM-CC User-to-User (U-U) messages. The SRM also does U-N configuration management and control of both Clients and Servers to allow their participation in the DSM-CC environment.

DSM-CC supports network topologies which consist of multiple Clients and multiple Servers. Any Client-Server pair can communicate together by establishing a Session. Each Client can have multiple simultaneous Sessions with any specific Server or any combination of Servers. For this phase of DSM-CC, a Session is typically limited to one Client, one Server, and one SRM. The exception is the case of Continuous Feed Sessions (CFS). A CFS may be used by, e.g., a stream broadcasting application, where broadcast "feeds" are established with the network with no particular Client specified. Clients may "attach" to a CFS by setting up a Session with the network to connect to the CFS and, optionally, to establish Client-unique resources (such as a return control channel that may be needed by an interactive application which shares a downstream feed, e.g., game show voting). Alternatively, Clients may "attach" to a CFS or another broadcast "feed" by using the U-N Switched Digital Broadcast Channel Change Protocol (SDB-CCP), when no Client-unique Resources are needed by the application (such as with traditional "pay-per-view").

0.3 Outline of the DSM-CC Specification

DSM-CC consists of a set of User-to-Network and User-to-User protocols. These protocols are described in the clauses listed below.

0.3.1 User-to-Network

- DSM-CC Message Header, clause 2
- U-N Configuration messages, clause 3
- U-N Session messages and flow diagrams for Session and Resource management, clause 4
- U-N Download messages, clause 7
- U-N Switched Digital Broadcast Channel Change Protocol, clause 10
- U-N Pass Thru messages, clause 12
- The transport of DSM-CC U-N messages using MPEG-2 Systems (ISO/IEC 13818-1), clause 9
- The transport of generic IP messages using DSM-CC Sections and MPEG-2 Systems, clause 9

0.3.2 User-to-User

- U-U Remote Procedure Call (RPC), clause 5
- U-U Session interface, clause 5
- U-U Download interface, clause 5
- U-U Object Carousel interface, clause 11
- U-U Local Object interface, clause 5
- U-U Stream Descriptors, clause 8

0.4 Supported Network Technologies

DSM-CC does not specify the underlying physical, data link, transport, or RPC layers of the overall protocol stack. However, DSM-CC does specify requirements for these layers in clause 9.

0.5 Supported Connection Types

DSM-CC supports the following connection types:

- Point-to-point
- Point-to-multi-point (broadcast)

User-to-User application and service exchanges are carried over point-to-point type connections.

The point-to-multi-point type connections are used to feed a single stream to multiple Clients. In this case, no single Client has control (e.g., for the purpose of pause, fast forward) of the received bit stream. However, in the case where the network (as opposed to the Client) does stream switching such as with Switched Digital Broadcast (SDB) applications, a means is provided for Clients to switch between streams using the SDB channel change protocol (SDB-CCP). The latter is useful for applications such as the so-called "enhanced pay-per-view" or "near video on demand".

0.6 DSM-CC Interfaces

The DSM-CC model (Figure 0-1) consists of three Subsystems:

- Client
- Server
- Session & Resource Manager (SRM)

Each subsystem is a logical embodiment within a DSM-CC System. It does not map directly to physical equipment. The SRM represents the DSM-CC functionality within a DSM-CC network (the Network).

In order to define interfaces, a DSM-CC System Reference Model is used to subdivide the DSM-CC environment into a hierarchy of several levels (see Figure 0-2):

- System
- Subsystem
- Entity
- Sub-entity

A Subsystem may contain more than one Entity. The types of Entities are:

- Client User-to-User Entity
- Client User-to-Network Entity
- Server User-to-User Entity
- Server User-to-Network Entity
- SRM User-to-Network Entity

DSM-CC signaling is always exchanged between specific Subsystem Entities.

From the normative perspective, the System Reference Model does not show any more detail below the Entities. However, for informative reasons, the Entities have in some cases been further divided into Sub-entities. For example, the Client and Server U-N Entities include a Session Manager, Resource Manager, and a Configuration Manager. On the Server side, the U-U Entity includes a Service Gateway and an Object Access Manager.

DSM-CC recognizes that Subsystems will require other Entities which are not specified by this part of ISO/IEC 13818. Examples are a Connection Management Entity and Application Entities.

Figure 0-2 shows the different entities and sub-entities of the DSM-CC system.

To show the boundaries between Entities, Figure 0-2 is divided into four layers:

- The Application Entity layer, which is outside the scope of this part of ISO/IEC 13818
- The User-to-User Entity, which is an Application/Service Control/Management Layer
- The User-to-Network Entity, which is a Session and Resource Control/Management Layer
- The Connection Control Entity, which is (typically part of) a Transport Layer, and is outside the scope of this part of ISO/IEC 13818.

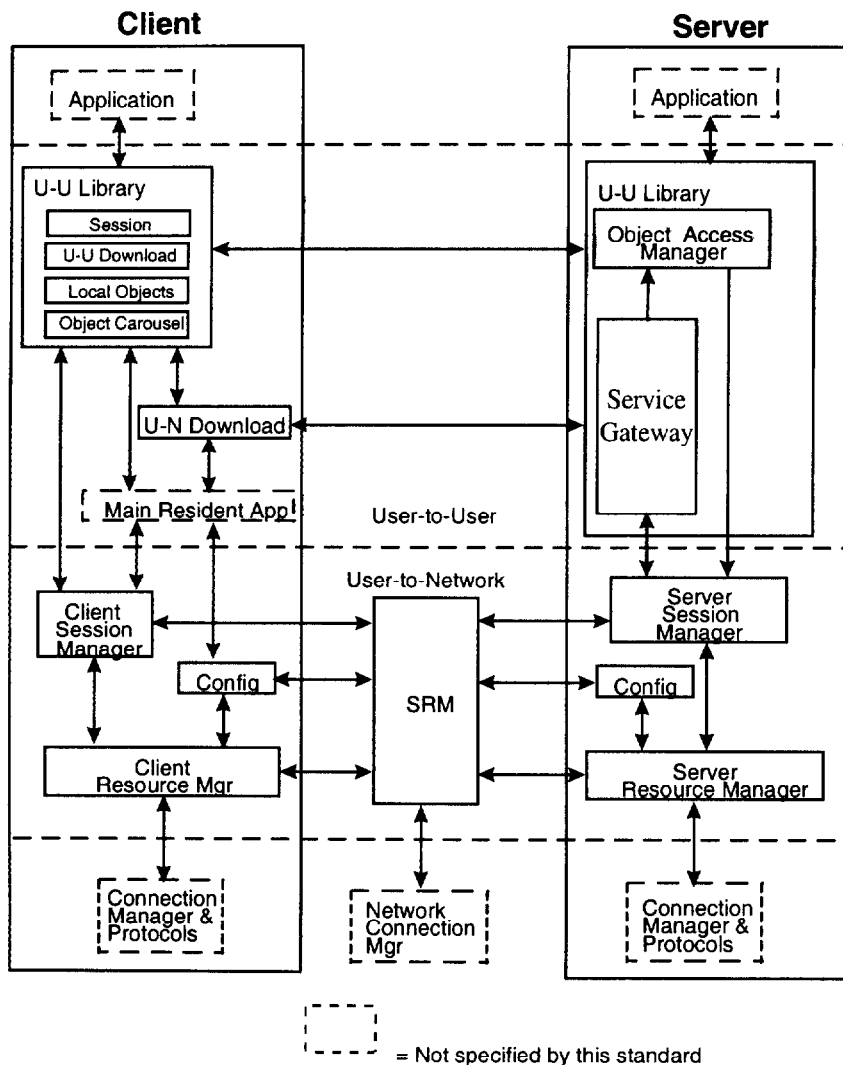


Figure 0-2 DSM-CC System Reference Model

Each of the lines with arrows in Figure 0-2 represents a logical interface. DSM-CC defines three types of interfaces:

- Inter-Entity

This is a preview of "ISO/IEC 13818-6:1998". [Click here to purchase the full version from the ANSI store.](#)

- Intra-Entity
- Intra-Subsystem

The Inter-Entity interfaces are between peer Entities in different Subsystems. The interfaces between the Sub-entities within a common Entity are called Intra-Entity interfaces. The interfaces between Entities within a common Subsystem are called Intra-Subsystem interfaces.

The DSM-CC System Reference Model specifies three communication paths over which DSM-CC messages are exchanged. The communication between U-U Entities are represented as the DSM-CC U-U Protocol. The communication between U-N Entities are represented as the DSM-CC U-N Protocol.

- Client U-U Entity to Server U-U Entity (U-U)
- Client U-N Entity to SRM U-N Entity (U-N)
- Server U-N Entity to SRM U-N Entity (U-N)

Table 0-1 summarizes Inter-Entity interfaces and Intra-Subsystem interfaces within the scope of DSM-CC.

Table 0-1 DSM-CC Interface Scope Summary

Peer 1	Peer 2	Protocol	Inter-Entity	Intra-Subsystem
Client U-U Library	Server Service Gateway	U-U	X	
Client U-U Library	Server Object Access	U-U	X	
Client Session Gateway	SRM	U-N	X	
Client Resource Manager	SRM	U-N	X	
Server Session Manager	SRM	U-N	X	
Server Resource Manager	SRM	U-N	X	
Client Configuration	SRM	U-N Config	X	
Server Configuration	SRM	U-N Config	X	
Server DSM Source (e.g. MPEG-2 Transport / Video / Audio)	Client DSM Consumer	(MPEG)	X ¹	
Download Server (Source)	U-N Download Client (Consumer)	Download	X ¹	
Object Carousel Server	Object Carousel Client	Object Carousel / Download	X ¹	
SDB Server	(SDB) Client	SDB-CCP	X ¹	
Client Application	Client U-U Library	U-U		X

Note 1: Interface not shown on Figure 0-2.

0.7 DSM-CC Interface Protocols

Figure 0-3 depicts DSM-CC protocols used at DSM-CC interface points. The top section of the figure contains some applications which may use DSM-CC. The middle section of the figure contains all of the DSM-CC specified protocols. The specific Transport Layers, the bottom section, are not specified by this part of ISO/IEC 13818.

Note that Figure 0-3 applies to the case where the full suite of DSM-CC protocols (except for the extended protocol groups) are employed. DSM-CC allows each protocol to be implemented without the others (see subclause 1.2 Profiles and Compliance). If the U-U Library is not used, then the implementation will not have an Application Portability Interface specified by DSM-CC.

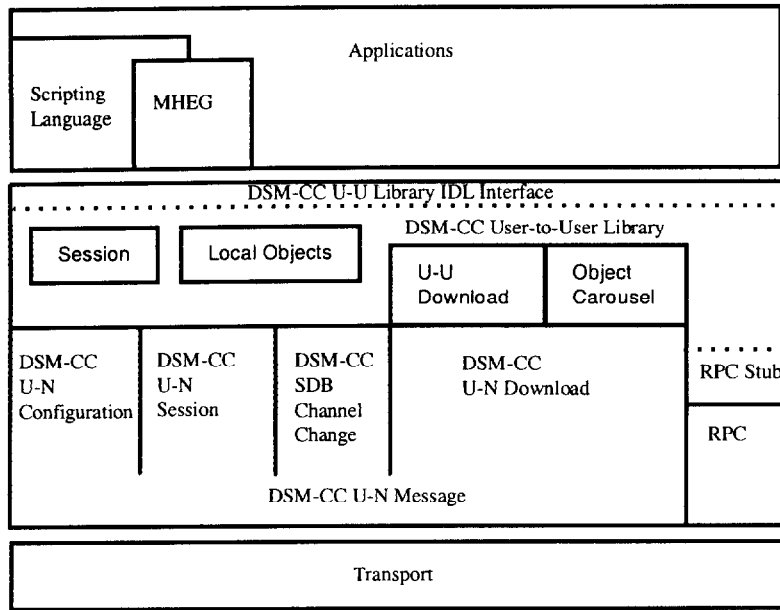


Figure 0-3 DSM-CC Interface Protocols

DSM-CC provides access to Stream and Data objects for applications (e.g., MHEG applications and scripting language applications). The primary application interface layer is the DSM-CC U-U Library Interface Definition Language (IDL), or Application Portability Interface. The U-U Library may in turn make use of the U-N Session Management, U-N Download, and U-U Object Carousel layer to establish and manage Sessions and Resources required for the management and delivery of the Stream and Data objects.

Table 0-2 lists the DSM-CC protocols. The protocols which use the DSM-CC message format are U-N Configuration, U-N Session, U-N SDB-CCP, U-N Download, and U-U Object Carousel (because it, in turn, uses U-N Download). In some cases, the use of a message passing interface is needed because the Client device may not have higher layer protocols (e.g., RPC) resident.

The U-U Library uses the services of the U-N protocols, but also adds its own on-the-wire protocol, the U-U RPC Stub Library, which are based on existing Remote Procedure Call (RPC) interfaces. The protocols which use RPC do so because it provides sophisticated object based services.

The third category is IDL, which is used in communicating within the Subsystem to applications.

Table 0-2 The DSM-CC Protocols used on the Interfaces

DSM-CC Protocols	Peer 1	Peer 2	U-N Message Format	IDL/RPC
U-N Configuration	Client / Server	SRM	x	
U-N Session	Client / Server	SRM	x	
U-N Download	Client	Download server	x	
U-N Switched Digital Broadcast Channel Change	Client	SDB server	x	
U-N Pass Thru	Client	Server	x	
U-U RPC	Client	RPC server		RPC
U-U Session	Client Application	Client U-U Library		IDL
U-U Download	Client Application	Client U-U Library		IDL
U-U Object Carousel	Object Carousel Client	Object Carousel	x	
U-U Local Objects	Client Application	U-U Library		IDL

The transport layer in Figure 0-3 may consist of any protocol which meets the transport requirements described in clause 9. Examples are, TCP or UDP over IP, AAL-5 over ATM, or DSM-CC/private_sections over MPEG-2 Transport Stream.

0.8 Communications Requirements

The DSM-CC U-N Configuration, U-N Session, U-N SDB-CCP and U-N Download messages all use the DSM-CC Message Format and are implemented using a simple message passing method; therefore, all have similar Transport Layer requirements. The U-U Object Carousel uses the U-N Download protocol and its associated transport requirements. The U-U RPC Stub Library uses RPC and its associated transport requirements.

The requirements for the underlying Transport services for all DSM-CC protocols are provided in detail in clause 9, Transport.

0.9 Methods of Specification

0.9.1 Messages

U-N messages are described in tables which list the bit or byte level assignment for all of the fields in each message. The syntactical structure of the messages are defined by Syntax Tables like the example below. Field names are shown in bold and always have an associated number of bytes indicated. All numeric values are unsigned big-endian (most significant byte first, most significant bit first) unless otherwise specified. The method of syntax description supports loops and 'procedures' using a pseudo-C syntax. In the example below, a for() loop, in normal font, indicates that the field **uuDataByte** repeats uuDataCount times. Also, the structure has been named UserData(), which now can in turn be referenced in other larger structures.

Syntax	Num. of Bytes
UserData(){	
uuDataLength	2
for(i=0;i<uuDataCount;i++) {	
uuDataByte	1
}	
privateDataLength	2
for(i=0;i<privateDataLength;i++) {	
privateDataByte	1
}	
}	

Figure 0-4 Example of U-N message syntax

The messages for U-N Configuration and U-N Session flow between Client and Network (SRM), and Server and Network (SRM). For consistency, the suffix of each of these messages use the following terminology:

Request - A message sent from a User (Client or Server) to the Network to begin a scenario.

Confirm - A message being sent from the Network to a User (Client or Server) in response to a Request message.

Indication - A message which is sent from the Network to a User.

Response - A message from a User to the Network in response to an Indication message.

Clause 9, Transport, defines the communications requirements (reliability, addressing etc.) for the delivery of these messages.

A standard programming API for the use of these messages is outside the scope of this part of ISO/IEC 13818.

0.9.2 Message Flow Diagram Scenarios

Flow diagrams have been provided to help explain the use of the DSM-CC message protocols. These diagrams show the sequence and direction of flow for the messages of a specific scenario. In these diagrams, the time axis runs vertically, with messages lower on the diagrams representing later transmission. The selected scenarios are the most typical ones and do not represent the exhaustive list of examples of scenarios. The Specification and Description Language (SDL) representations provide a more exhaustive representation, including exception cases.

0.9.3 Specification and Description Language

The SDL-language is officially defined in ITU-T recommendation Z.100. For the translation of the DSM-CC specification into SDL, SDL-88 (Z.100 blue book) is used. There are several advantages to using SDL:

- Contrary to the textual part, usage of SDL in the specification makes it unambiguous due to the fact that SDL is a formal language.
- One representation of SDL is the graphical one. This makes the language more comprehensible.
- The SDL specification can be analyzed for completeness and correctness.
- It is easy to generate executable code in order to simulate and validate the specification.
- The specification can also be used for conformance test purposes.

For simulation purposes, Message Sequence Charts (MSC), as defined in ITU-T recommendation Z.120, are used.

A model described in SDL consists of three different types of levels.

1. System level
2. Block level
3. Process level

The highest level of the SDL model is the system level. The system is surrounded by the environment represented by a rectangle in the graphical representation. On the system level, the model of the system is described in a very rough shape

divided into one or more blocks. The blocks can contain either new blocks or processes. At some block level, the content is one or several processes in each block. The process level could then describe logical parts of the system related to each other with the signals exchanged between them.

A static process is created at start-up time for the system. A dynamic process is created during runtime by another process. The number of dynamic processes which may be created is set by a constant value. A process can be stopped by the process itself at any point in time.

A process is a state machine and the only way to move from one state to another state is via a transition. One or several possible transitions can be connected to a state. A transition is always initiated by either an input signal or an enabling condition. An input signal can be generated by an output signal from an outside process, from within the same process, or by an expired timer. Here, the environment is also regarded as a process. The input signal is put in an input queue which is a common queue for the process.

When an input signal is consumed, a transition is started and the actual code defined between the state and the next state is executed. In the graphical representation, the code consists of one or several graphical symbols with some additional plain text; variables may be assigned new values in a task, questions may be answered in a decision, an output signal may be sent to another process, etc.

Figure 0-5 shows some common SDL symbols. Complete specification of SDL is outside of the scope of this specification, but may be found in ITU-T Z.100 and Z.120.

The intent is to have the message flow diagrams and prose be consistent with the SDL tables. Since the SDL is more exhaustive, if there is any form of contradiction between the prose and SDL, the SDL shall take precedence.

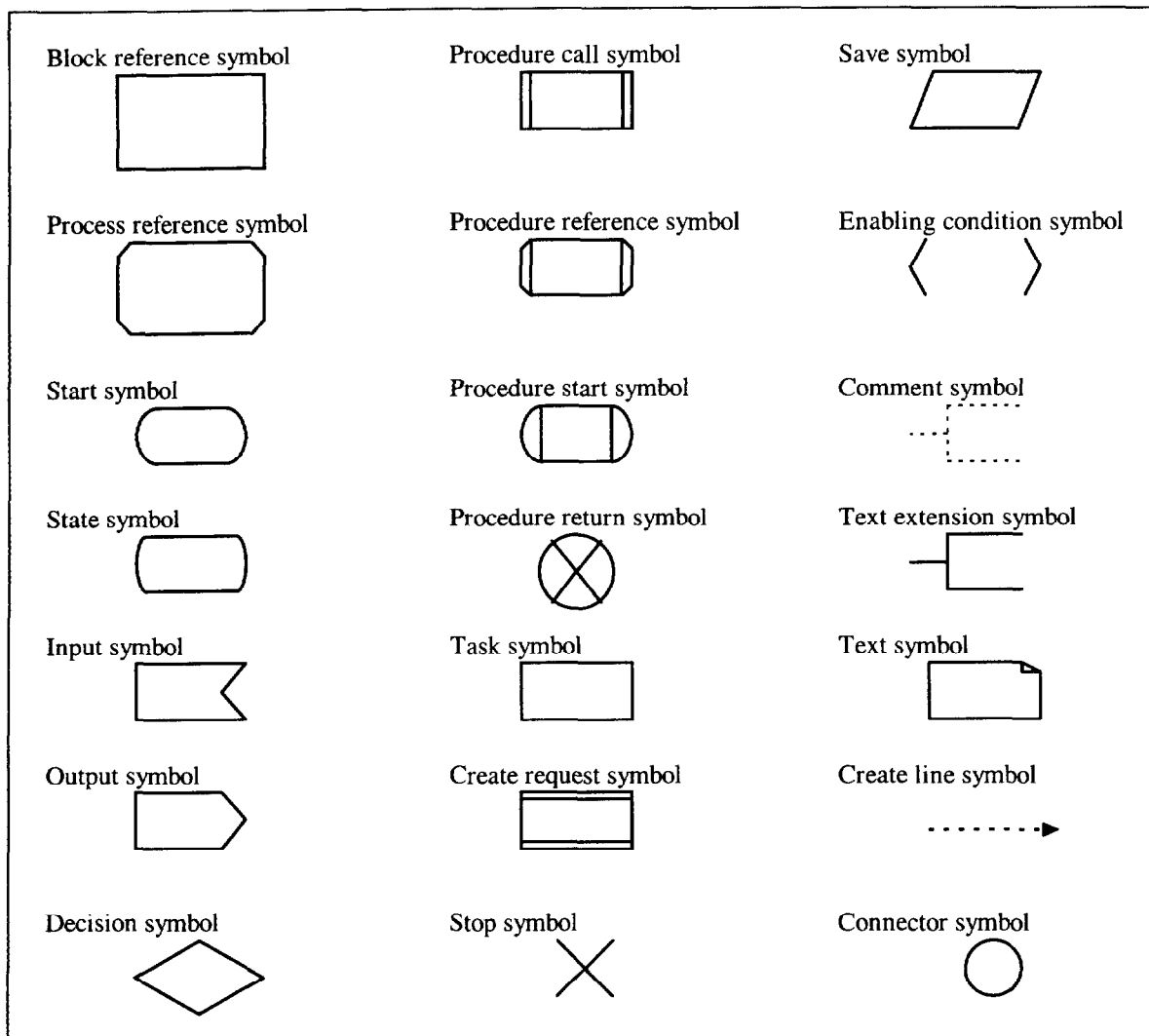


Figure 0-5 SDL Symbols

0.9.4 Interface Definition Language (IDL)

The U-U API primitives that use RPC are defined in terms of OMG Interface Definition Language (IDL), defined by ISO/IEC 14750. The IDL provides a grammar for defining the function call-like API specification for each primitive. Primitives written in the IDL are compiled by an IDL compiler to produce client and server stubs (executable code that implements packet formation, dispatch, receipt, and interpretation) and a header file used during compilation of the client and server applications.

0.9.5 Remote Procedure Call (RPC)

U-U functionality exploits a Remote Procedure Call (RPC) protocol. A RPC allows implementation of a client-server model in which applications on a client are written to call functions that are similar to those that might be used if all actions were to be executed locally. For those U-U API primitives that use the RPC, the RPC and data encoding defines the actual bits that are exchanged as primitives are executed.

The downstream reply from the Server can be delivered via encapsulation within a MPEG-2 Transport Stream. Although this part of ISO/IEC 13818 specifies how to encapsulate common protocols (e.g., IP) over MPEG-2 Transport, there is no requirement that control messages or RPC messages be delivered within MPEG-2 Transport Streams.