ISO/IEC 14543-5-1

Edition 1.0 2010-02

INTERNATIONAL STANDARD

Information technology – Home electronic system (HES) architecture – Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol

ISO/IEC 14543-5-1:2010-02(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
67 000 electrotechnical terminology entries in English and French extracted from the Terms and definitions clause of IEC publications issued between 2002 and 2015. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

ISO/IEC 14543-5-1

Edition 1.0 2010-02

INTERNATIONAL STANDARD

Information technology – Home electronic system (HES) architecture – Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 35.240.67

ISBN 2-8318-1076-0

CONTENTS

FOREWORD 8
INTRODUCTION 9
1 Scope 10
2 Normative references 10
3 Terms, definitions and abbreviations 11
3.1 Terms and definitions 11
3.2 Abbreviations 12
4 Conformance 13
4.1 IGRS network 13
4.2 IGRS devices 13
5 IGRS architecture 14
5.1 Overview 14
5.2 IGRS Core Protocol 15
5.3 IGRS Application Profile 15
5.4 IGRS Application 16
5.5 IGRS and other standards 17
6 IGRS device interaction model 18
6.1 Overview 18
6.2 Device online 19
6.3 Device (group) discovery 19
6.4 Device pipe setup 19
6.5 Device group setup and join 20
6.6 Service discovery 20
6.7 Session setup 21
6.8 Service invocation 21
6.9 Session termination 21
6.10 Device/Service online/offline event subscription 21
6.11 Device/Service online/offline event notification 22
6.12 Device/Service online/offline event unsubscription 22
6.13 Pipe disconnection 22
6.14 Device group dismiss and secession 22
6.15 Device offline 23
7 IGRS message framework 23
7.1 Overview 23
7.2 IGRS request/notification message structure 24
7.3 IGRS response message structure 25
7.4 IGRS message based on secure device pipe 26
7.4.1 Message authentication code generation 26
7.4.2 Message encryption 27
8 IGRS device and service description 27
8.1 IGRS device description 27
8.1.1 Device description template 27
8.1.2 Device identifier 31
8.1.3 Device group identifier 31
8.1.4 Device type identifier 32
8.1.5 Device security mechanism descriptor 33

8.2 IGRS service description 40
8.2.1 Service description template 40
8.2.2 Service identifier 51
8.2.3 Service type identifier 51
8.2.4 Service access control 52
8.2.5 Identity authentication mechanism of service sccess control 53
8.3 IGRS client description 56
8.4 IGRS user description 56
9 IGRS device grouping 56
9.1 Device advertisement 56
9.1.1 Device online advertisement 56
9.1.2 Device offline advertisement 58
9.2 Device pipe management 58
9.2.1 Unsecure device pipe setup 58
9.2.2 Unsecure device pipe maintenance 58
9.2.3 Secure device pipe setup 59
9.2.4 Secure device pipe teardown 79
9.2.5 Device trust relationship 80
9.2.6 Device online detection 80
9.3 Detailed device description document retrieval 81
9.3.1 Retrieve detailed device description document request 81
9.3.2 Retrieve detailed device description document response 82
9.4 Retrieve detailed device description document based on non-secure pipe 84
9.5 Device group setup 84
9.5.1 Global peer-to-peer device group 84
9.5.2 Specified peer-to-peer device group 84
9.5.3 Centralised device group 86
9.6 Device search 90
9.6.1 Device search based on multicast 90
9.6.2 Device search by proxy 94
9.7 Device online/offline event subscription 97
9.7.1 Device online/offline event subscription request 97
9.7.2 Device online/offline event subscription renewal request 99
9.7.3 Device online/offline event subscription response 100
9.7.4 Device online/offline event unsubscription 101
9.7.5 Device online/offline event notification 102
9.8 Device group search 104
9.8.1 Device group search request message 104
9.8.2 Device group search response message 105
10 IGRS resource sharing 107
10.1 Service online advertisement 107
10.1.1 Service online advertisement based on mutilcast 107
10.1.2 Service online registration and offline notification based on device pipe 109
10.2 Service search 111
10.2.1 Service search based on multicast 111
10.2.2 Service search by proxy 114
10.3 Service online/offline event subscription 117
10.3.1 Service online/offline event subscription request 117

10.3.2 Service online/offline event subscription renewal request 119
10.3.3 Service online/offiline event subscription response 120
10.3.4 Service online/offline event unsubscription 121
10.3.5 Service online/offline event notification 122
10.4 Service description document retrieval 124
10.4.1 Retrieve service description document request 124
10.4.2 Retrieve service description document response 126
10.4.3 Other approaches to retrieve service description documents 127
10.5 Session 127
10.5.1 Session setup condition 127
10.5.2 Common session setup and teardown process 127
10.5.3 Session setup when service access control in master/slave device group is not consistent with device pipe security attribute 131
10.6 Service invocation 137
10.6.1 Service invocation request message 137
10.6.2 Service invocation response message 137
10.6.3 Notification message based on session 138
11 Request/response status codes 139
Annex A (normative) IGRS service discovery protocols (ISDP) 142
A.1 General 142
A.2 ISDP message format 142
A.2.1 General 142
A.2.2 ISDP start-lines 142
A.2.3 ISDP message headers 142
A.2.4 ISDP processing rules 143
A.3 ISDP usage in IGRS specification 143
Annex B (normative) Description documents 145
B.1 Specification description 145
B.2 Session description 249
B.3 Service description 250
B.4 Pipe description 253
B.5 Device template 254
B.6 Master slave device group advertisement 256
B.7 Device type list 256
B.8 Peer-to-peer device group advertisement 257
B.9 Device description 257
Bibliography 259

Figure 1 – IGRS specification framework 15
Figure 2 – IGRS application interaction 17
Figure 3 – IGRS device interaction model 18
Figure 4 – Secure device pipe setup 59

Table 1 – IGRS request and notification message 24
Table 2 – IGRS response message 25
Table 3 – Message authentication code 27
Table 4 – Device security mechanism protocol algorithm 40
Table 5 – Service access control policy 53
Table 6 – Device authentication mechanisms and the corresponding encryption algorithm descriptor 56
Table 7 – Device online advertisement 57
Table 8 – Device offline advertisement 58
Table 9 – Pipe setup request based on symmetric-key cryptosystem 60
Table 10 – Pipe setup response based on symmetric-key cryptosystem 61
Table 11 – Pipe setup request based on symmetric-key authentication, encrypted message transmission, and authentication mechanism 61
Table 12 – Pipe setup response based on symmetric-key authentication, encrypted message transmission, and authentication mechanism 62
Table 13 – Pipe setup request based on authentication, encrypted message transmission, and authentication mechanism of public-key cryptosystem 63
Table 14 – Pipe setup response based on authentication, encrypted message transmission, and authentication mechanism of public-key cryptosystem 63
Table 15 – Pipe setup request based on trusted third party authentication, encrypted message transmission, and authentication mechanism 64
Table 16 – Pipe setup response based on trusted third party authentication, encrypted message transmission, and authentication mechanism 64
Table 17 – Authentication request based on identity authentication and message authentication mechanism of symmetric-key cryptosystem 65
Table 18 – Authentication response based on identity authentication and message authentication mechanism of symmetric-key cryptosystem 66
Table 19 – Authentication result request based on identity authentication and message authentication mechanism of symmetric-key cryptosystem 67
Table 20 – Authentication result response based on identity authentication and message authentication mechanism of symmetric-key cryptosystem 67
Table 21 – Authentication request based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem 68
Table 22 – Authentication response based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem 69
Table 23 – Authentication result request based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem 70
Table 24 – Authentication result response based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem 70
Table 25 – Authentication request based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem 71

Table 26 – Authentication response based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem 72
Table 27 – Authentication result request based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem 73
Table 28 – Authentication result response based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem 74
Table 29 – Authentication request based on authentication, encrypted message transmission, and authentication mechanism of trusted third party 75
Table 30 – Authentication response based on authentication, encrypted message transmission, and authentication mechanism of trusted third party 76
Table 31 – Authentication result request based on authentication, encrypted message transmission, and authentication mechanism of trusted third party 77
Table 32 – Authentication result response based on authentication, encrypted message transmission, and authentication mechanism of trusted third party 78
Table 33 – Secure device pipe setup confirmation request 78
Table 34 – Secure device pipe setup confirmation response 79
Table 35 – Secure device pipe teardown notification message 80
Table 36 – Trust relationship formed between devices after pipe setup 80
Table 37 – Device online detection request message 81
Table 38 – Device online detection response message 81
Table 39 – Device description document retrieval request message 82
Table 40 – Device description document retrieval response message 83
Table 41 – Device group advertisement message of specified peer-to-peer device group 85
Table 42 – Device leaves a specified peer-to-peer device group quit group message 86
Table 43 – Device group advertisement message of master-slave device group 87
Table 44 – Request message to join a master-slave device group 88
Table 45 – Response message to join a master-slave device group 88
Table 46 – Device group dissolve notification message sent by master device 89
Table 47 – Withdraw notification message sent by slave device 89
Table 48 – Device search request message 90
Table 49 – Device search response message 92
Table 50 – Device search request that slave device generates to master device 94
Table 51 – Device search response message 96
Table 52 – Device event subscription request message 98
Table 53 – Device online/offline event subscription renewal request message 100
Table 54 – Device online/offline event subscription response message 101
Table 55 – Device online/offline event unsubscription message 102
Table 56 – Device online/offline event notification message 103
Table 57 – Device group search request message 105
Table 58 – Device group search response message 106
Table 59 – Service online advertisement message 108
Table 60 – Service offline advertisement message 109
Table 61 – Service online registration notification message 110
Table 62 – Service offline notification message 111
Table 63 – Multicast-based service search request message 112

Table 64 – UDP unicast-based service search request message 113
Table 65 – Service search request message by proxy 115
Table 66 – Service search response message by proxy 116
Table 67 – Service online/offline event subscription request message 118
Table 68 – Service online/offline event subscription renewal request 120
Table 69 – Service online/offline event subscription response 121
Table 70 – Service online/offline event unsubscription message 122
Table 71 – Service online/offline event notification message 123
Table 72 – Retrieve service description document request message 125
Table 73 – Retrieve service description document response message 126
Table 74 – Common session setup request message 128
Table 75 – Token setup and structure 129
Table 76 – Common session setup response message 130
Table 77 – Common session teardown notification message 131
Table 78 – Retrieve session encryption key generation request 133
Table 79 – Retrieve session encryption key generation response 134
Table 80 – Session encryption key transfer request message 135
Table 81 – Session encryption key transfer response message 136
Table 82 – Service invocation request message 137
Table 83 – Service invocation response message 138
Table 84 – Notification message based on session 139
Table 85 – Response status code category 140
Table 86 – Response status code definition 141

INFORMATION TECHNOLOGY – HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol

FOREWORD

1) ISO (International Organization for Standardization) and IEC (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. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) 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.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14543-5-1 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 14543 series, under the general title *Information technology – Home electronic system (HES) architecture*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

ISO/IEC 14543-5, Intelligent Grouping and Resource Sharing for HES (IGRS), is divided into six parts:

- **IGRS Part 5-1: Core Protocol**
  - Specifies the TCP/IP protocol stack as the basis and the HTTP protocol as the message-exchanging framework among devices.
  - Defines a series of device and service interaction/invocation standards, including device and service discovery protocol, device and service description, service invocation, security mechanisms, etc.
  - Specifies core protocols for a type of home network that supports streaming media and other high-speed data transport within a home.
- **IGRS Parts 5-2#: Application profile** (under consideration)
  - Based on the IGRS Core Protocol.
  - Defines a device and service interaction mechanism, as well as application interfaces used in IGRS Basic Applications.
  - Multiple application profiles are being developed, including:
    - Part 5-21: AV Profile (under consideration)
    - Part 5-22: File Profile (under consideration)
  - Additional application profiles are planned (part numbers to be assigned; these projects are under consideration)
    - Part 5-2w: DVD Profile
    - Part 5-2x:QoS Profile
    - Part 5-2y: DMCP Profile
    - Part 5-2z: Universal Control Profile
- **IGRS Part 5-3: Basic Application** (under consideration)
  - Includes an IGRS basic application list.
  - Defines a basic application framework.
  - Addresses operation specifics (device grouping, service description template, etc.), function definitions, and service invocation interfaces.
- **IGRS Part 5-4: Device Validation** (under preparation)
  - Defines a standard method to validate an IGRS-compliant device.
- **IGRS Part 5-5: Device Types** (under consideration)
  - Defines IGRS Device types used in IGRS applications.
- **IGRS Part 5-6: Service Types** (under consideration)
  - Defines basic service types used in IGRS applications.

INFORMATION TECHNOLOGY – HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol

1 Scope

This part of the ISO/IEC 14543 specifies the services and protocol of the application layer for use by IGRS Devices in the Home Electronic System. An IGRS Device (Intelligent Grouping and Resource Sharing Device) includes the communications protocol specified in the multiple parts of ISO/IEC 14543-5. The objective of this standard is to enable resource sharing and service collaboration among devices. This standard describes:

- the interoperability mechanism;
- the process and messaging format of device discovery and device grouping;
- the process and messaging format of resource sharing among IGRS Devices;
- IGRS Device and service description requirements.

This standard is applicable to resource sharing and service collaboration among computers, consumer electronics, and communication devices in a Local Area Network (LAN) or Personal Area Network (PAN) environment, especially in a wireless dynamic network.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document, including any amendments, applies.

The provisions of the referenced specifications other than ISO/IEC, IEC, ISO and ITU documents, as identified in this clause, are valid within the context of this International Standard. The reference to such a specification within this International Standard does not give it any further status within ISO or IEC. In particular, it does not give the referenced specification the status of an International Standard.

ISO/IEC 9594-8:2005, *Information technology – Open Systems Interconnection – The directory: Public-key and attribute certificate frameworks*

ISO/IEC 10118-3:2004, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*

ISO/IEC 18033-3, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*

ISO/IEC 19790, *Information technology – Security techniques – Security requirements for cryptographic modules*

ISO/IEC 29341-1:2008, *Information technology – UPnP Device Architecture – Part 1: UPnP Device Architecture Version 1.0*

IEEE 1363:2000, *Standard Specifications For Public Key Cryptography*

IETF RFC 1510: *The Kerberos Network Authentication Service (V5)*

IETF RFC 1766: *Tags for the Identification of Languages*

IETF RFC 2234: *Augmented BNF for Syntax Specifications: ABNF*

IETF RFC 2616: *Hypertext Transfer Protocol -- HTTP/1.1*

IETF RFC 2774: *An HTTP Extension Framework*

IETF RFC 3447: *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*

W3C-REC-XML-1998-210:1998, *Extensible Markup Language (XML) 1.0*

W3C SOAP 1.2: *Simple Object Access Protocol Version 1.2*
http://www.w3.org/2002/12/soap-envelope

W3C WSDL 2.0: *Web Service Description Language Version 2.0*
http://www.w3.org/TR/wsdl20/

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document the following terms and definitions apply.

3.1.1
centralised device group
set of IGRS Devices with one IGRS Device acting as the master

NOTE 1 The master is responsible for managing the setup, for dismissing a Device Group, and for processing a join request from other devices.

NOTE 2 The master device and other IGRS Devices in such a Device Group form a centralised or master-slave relationship.

3.1.2
client identifier
unique identifier associated with a Client on an IGRS Device to which that Client belongs

3.1.3
device group
multiple IGRS Devices that are organised into a logical group through the device group management mechanism in the IGRS specification

NOTE Each IGRS Device in a Device Group follows common interaction rules. Two types of Device Groups are defined: peer-to-peer Device Group and centralised (master-slave) Device Group.

3.1.4
device identifier
globally unique device identifier associated with one IGRS Device

3.1.5
device pipe
channel used to transfer device interaction messages

NOTE This channel is set up through the pipe setup mechanism in the IGRS Specification.

3.1.6
IGRS client
application that draws upon the services of one or more connected IGRS Devices

NOTE Multiple client instances can exist on a network at the same time.

3.1.7
IGRS device
information device that conforms to the IGRS specification

3.1.8
IGRS service
sharable resource encapsulated in an IGRS Device by implementing application interfaces and providing services for other IGRS Devices

NOTE An IGRS Service has an invocation interface that meets the requirements of the IGRS specification. These invocation interfaces are described and announced on the network through the IGRS Service Description Specification.

3.1.9
IGRS specification
ISO/IEC 14543-5 series of standards

3.1.10
IGRS user
owner of an IGRS Device and Client

3.1.11
peer-to-peer device group
set of IGRS Devices where each IGRS Device in this set has a peer-to-peer relationship with each other

3.1.12
service identifier
unique identifier assigned to a service provided by a specific IGRS Device

NOTE Note that the same type of service may be provided by multiple IGRS Devices within the same network. Each instance of a service has a unique service identifier on the IGRS Device providing that service.

3.1.13
service type
category of IGRS Service defined according to the set of resources encapsulated

NOTE The Service Type enables service applications in the same category to have common invocation interfaces.

3.1.14
user identifier
identifier of an IGRS user

3.2 Abbreviations

The following acronyms and abbreviations are used in this standard and commonly used in other industry publications.

3DES	3 Data Encryption Standard
ABNF	Augmented Backus-Naur Form
AES	Advanced Encryption Standard
CA	Certificate Authority

ECC-192	192-bit Elliptic Curve Cryptography
ElGamal	ElGamal Asymmetric Key Encryption Algorithm (Digital Signature Algorithm), see IEEE 1363-2000
FIPS	Federal Information Processing Standards
HTTP	Hypertext Transport Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGRS	Intelligent Grouping and Resource Sharing, see ISO/IEC 14543-5 series
IGRS/1.0	Version 1.0 of the IGRS protocol, see this International Standard
ISDP	IGRS Service Discovery Protocol, see this International Standard
IP	Internet Protocol
PRNG	Pseudo-random Number Generator
RNG	Random Number Generator
RSA-1024	Rivest-Shamir-Adleman-1024
SHA-1	Secure Hash Algorithm 1
SHA-256	Secure Hash Algorithm 256
SSDP	Simple Service Discovery Protocol
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
UPnP	Universal Plug and Play
UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
XML	Extended Markup Language,see W3C-REC-XML-1998-210:1998

4 Conformance

4.1 IGRS network

An IGRS Network is defined by the architecture specified in Clause 5, the IGRS Device interaction model specified in Clause 6, the message formats specified in Clause 7, device grouping procedures specified in Clause 9, and the resource sharing model specified in Clause 10. All mandatory requirements shall be implemented for conformance with this standard.

4.2 IGRS devices

To be considered IGRS compliant, each device shall meet the criteria defined by the IGRS Device interaction model requirements specified in Clause 6, the message format (sent and received) requirements specified in Clause 7, and the device and service description requirements according to the templates specified in Clause 8.

In addition, a basic IGRS-compliant device shall also support the following requirements.

- The device shall be able to send a Device Online Advertisement Message as specified in 9.1.

- The device shall support other IGRS Devices to set up secure pipes based on the null security mechanism using a secure Device Pipe IP address specified in an online advertisement message.
- The device shall be able to respond as specified in 9.3 to Device Description Document Retrieval Request Messages from other IGRS Devices.
- The device shall be able to respond as specified in 9.6 to Device Search Request Messages from other IGRS Devices.
- The service on the device shall be able to send Service Online/Offline Advertisement Messages as specified in 10.1.
- The device shall be able to respond as specified in 10.2 to Device Search Request Messages Based on Multicast from other IGRS Devices.
- The device shall be able to respond as specified in 10.4 to Service Description Document Retrieval Request Messages from other IGRS Devices.
- The device shall be able to respond as specified in 10.5 to Common Session Setup Request Messages from other IGRS Device, and establish session accordingly.
- The device shall be able to respond as specified in 10.6 to Service Invocation Request Messages from other IGRS Devices.

5 IGRS architecture

5.1 Overview

The scope of the ISO/IEC 14543-5 series of standards, called the IGRS specification includes enterprises, public areas, and homes. The IGRS specification includes a universal resource description and a universal service interface. The IGRS specification was developed for the following purposes:

- to support device resource sharing and synergistic applications among diversified information devices, consumer electronic devices, and communication devices in a LAN or PAN network;
- to enable the interoperability among the devices and to make them easy to use, utilising the features and functionality of each device.

The IGRS Specification Architecture is shown in Figure 1. It is based on the TCP/IP protocol suite (specified in IETF RFC 793) and uses the HTTP protocol as the framework. The IGRS Core Protocol supports wired LAN, ISO/IEC 8802-3-family network media, and ISO/IEC 8802-11, as well as Bluetooth. IGRS defines device interaction specifications based on the HTTP/1.1 protocol, including the IGRS Core Protocol comprising the platform of device discovery and resource sharing, and the IGRS Application Profile comprising the collaborative service platform.

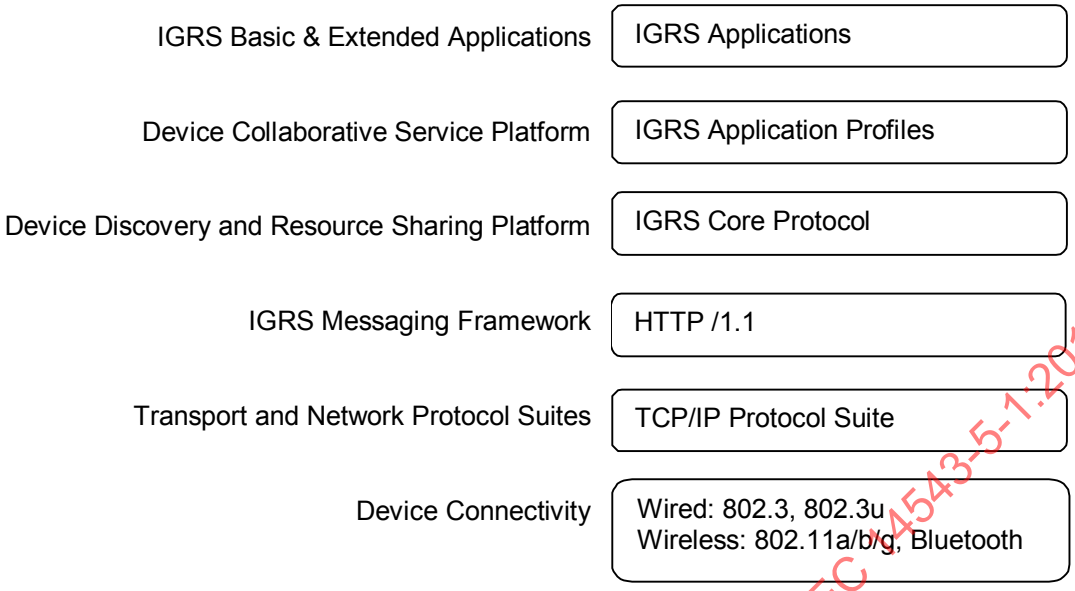

Figure 1 – IGRS specification framework

The techniques specified in the IGRS specification for device connection, data transfer, network protocol, and device message interaction framework are all widely-adopted, mature, open industry technologies. As more and more new network technologies become mature, IGRS may be amended to support these technologies (such as IPv6, etc.). ISO/IEC 14543-5-1 to ISO/IEC 14543-5-6 specify three components: IGRS Core Protocol, IGRS Application Profile, and IGRS Basic Application. The IGRS Core Protocol defines IGRS Device grouping and the interaction mechanism between client and service. Based on the IGRS Core Protocol, the IGRS Application Profiles define service descriptions and interaction logic for various IGRS applications. The various IGRS applications are standardised and shall be implemented as specified in the corresponding IGRS Application Profiles to ensure interoperability. Additional functions may be implemented using protocols not specified in the ISO/IEC 14543-5 series. An application combining a profile specified in ISO/IEC 14543-5-2# with private functions is called an IGRS extended application. The development of an IGRS extended application is specified in ISO/IEC 14543-5-3.

5.2 IGRS Core Protocol

The IGRS Core Protocol, specified in ISO/IEC 14543-5-1, defines the mechanism of IGRS Device discovery and resource sharing, including a device discovery mechanism, Device Pipe setup mechanism, service discovery mechanism, device group management mechanism, session management mechanism, service access mechanism, etc.

The device discovery mechanism defines the process by which an IGRS Device advertises itself and discovers other device information on the network. The Device Pipe setup mechanism defines the process by which two devices that have discovered each other establish a reliable connection pipe. The device group management mechanism defines the rules by which multiple devices form a specific device group on the basis of the connection pipe. The service discovery mechanism defines the process by which one device discovers the service information announced by other devices in the same device group. The session mechanism defines the process of session setup and management for supporting client and service access. The service access mechanism defines the rules that shall be followed in order to complete service invocation between the IGRS Client and the IGRS Service.

5.3 IGRS Application Profile

An IGRS Application Profile, specified in multiple parts of ISO/IEC 14543-5-2x series, is a series of application interaction rules defined for end applications based on the IGRS Core

Protocol. For example, the audio/video framework of home multimedia defines the IGRS Services that every audio/video device shall implement for applications in a home environment, and the interaction rules between the service provider and the client.

5.4 IGRS Application

IGRS applications, specified in ISO/IEC 14543-5-3, are based on the IGRS Core Protocol, specified in ISO/IEC 14543-5-1 and an IGRS Application Profile, specified in ISO/IEC 14543-5-2#. IGRS applications include two categories: IGRS basic applications that are fully specified in ISO/IEC 14543-5-3 and IGRS extended applications (that include vendor specific extensions). An IGRS extended application is based on the IGRS basic protocol and IGRS application profile. An IGRS extended application conforms to the IGRS specification and is intended to diversify and to enhance the current IGRS device functionalities. It is to be noted that IGRS Devices are categorised into different types based on different usage scenarios. IGRS basic applications are standardised applications that relate to specific device types and possess the basic functionalities of an IGRS Device, as defined in ISO/IEC 14543-5-5. In addition to basic applications implemented by IGRS Devices, application developers have the freedom to develop various applications conforming to the IGRS specifications. The developers shall implement the IGRS Core Protocol and may extend an IGRS Application Profile with proprietary functions to fully utilise the functionalities provided by their IGRS Devices. This kind of application extension is called an IGRS extended application.

An IGRS application is implemented through the interaction between one or more services and one or more clients that use these services. A typical IGRS application interaction is shown in Figure 2.

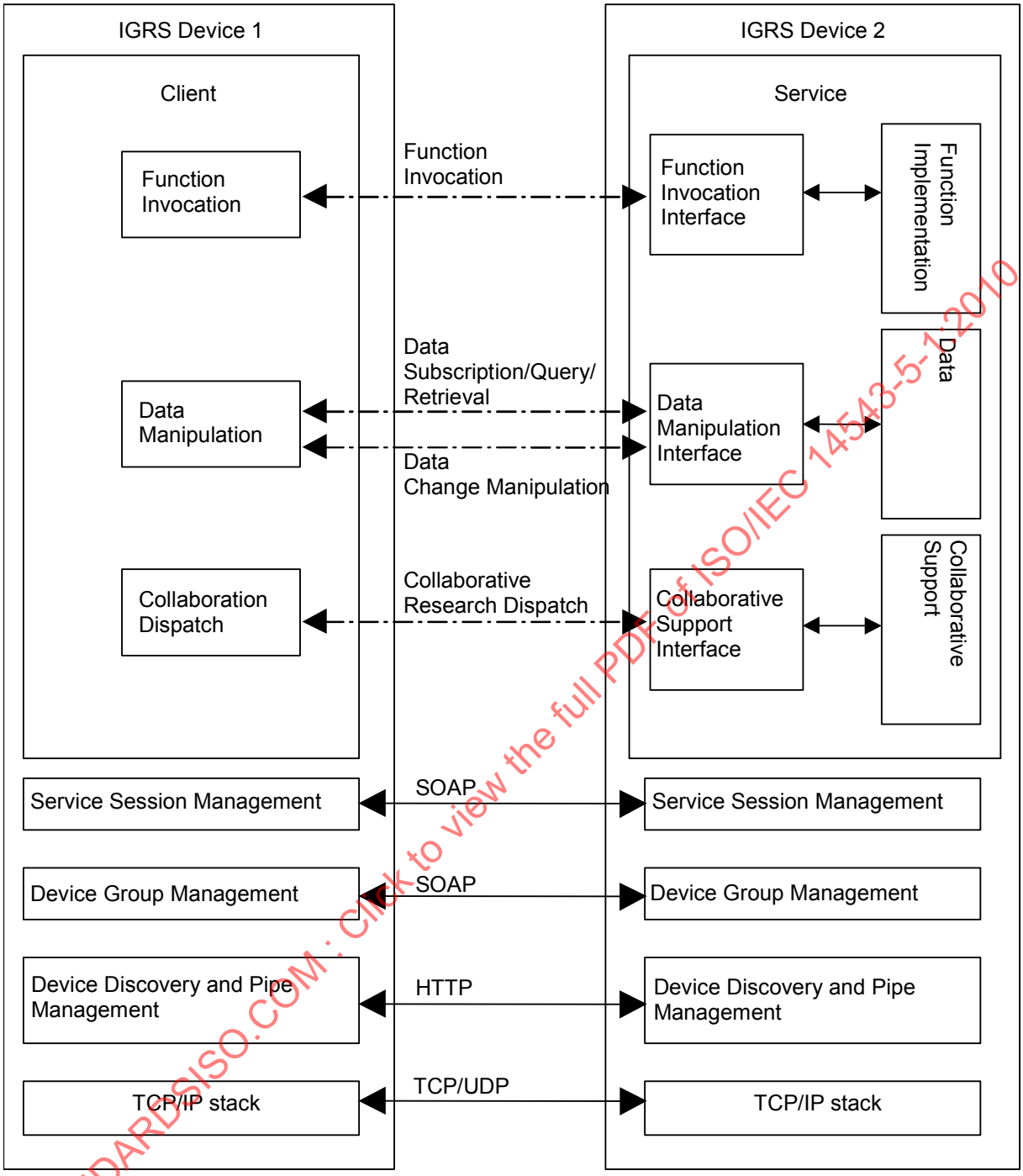

Figure 2 – IGRS application interaction

5.5 IGRS and other standards

The IGRS specification is in part based on existing specifications. The IGRS specification uses the HTTP/1.1 and SOAP/1.2 protocols (specified in http://www.w3.org/2002/12/soap-envelope) as the message framework for device interaction. IGRS supports interoperability with many other specifications due to the wide adoption of the HTTP/1.1 and SOAP/1.2 protocols. Therefore, the IGRS specification provides a foundation for interoperability.

6 IGRS device interaction model

6.1 Overview

The IGRS Device Interaction Model shown in Figure 3 describes the typical process of an IGRS Device coming online, discovering other IGRS Devices, joining an IGRS Device group, discovering the services of other IGRS Devices in the device group, performing service invocation, and finally going offline. It includes 14 steps (see 6.2 through 6.15).

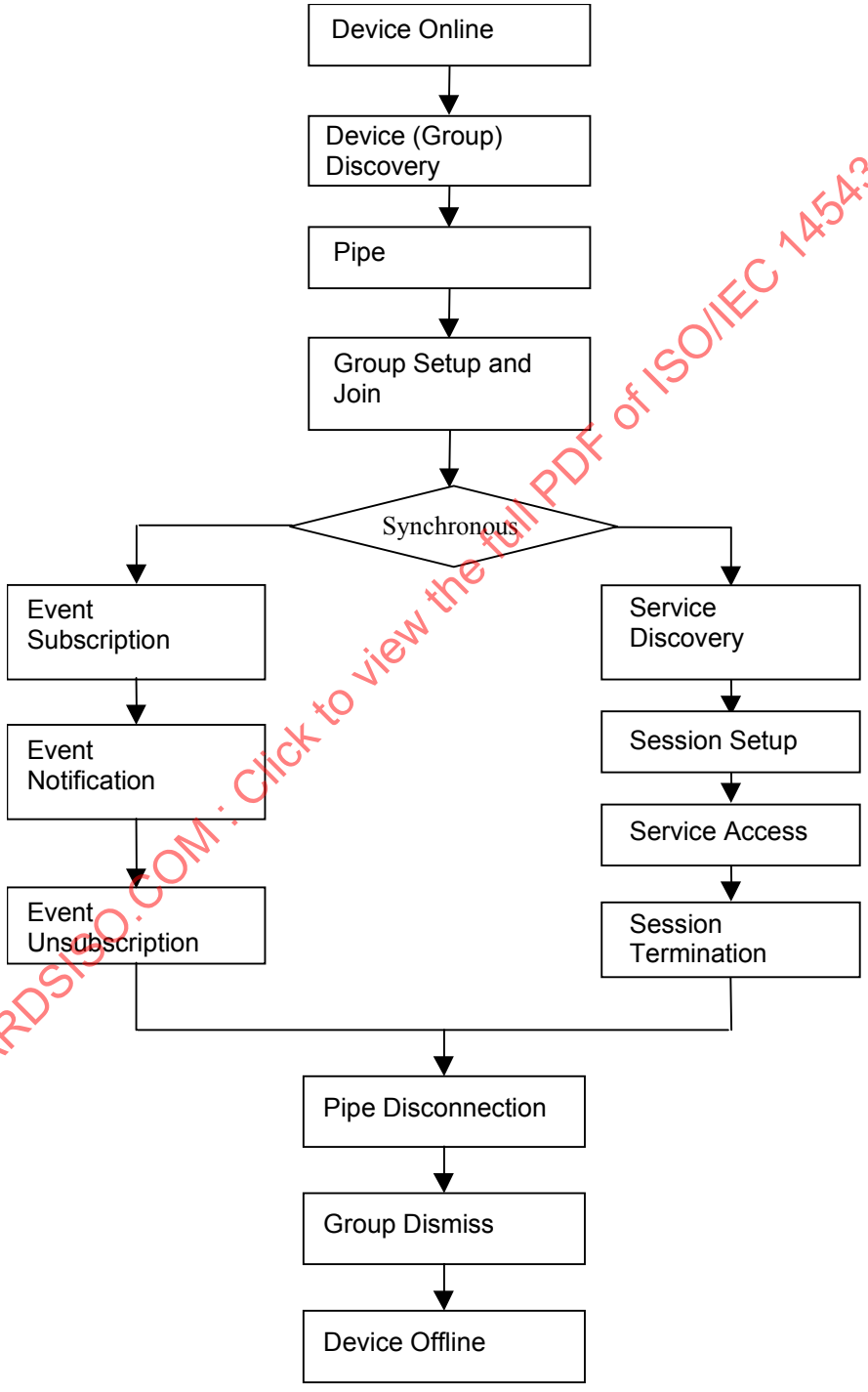

Figure 3 – IGRS device interaction model

6.2 Device online

An IGRS Device shall send a device online advertisement to the specified multicast addresses on all network connections on a regular time interval. The device advertisement shall contain such information as device name, device model, device security attribute requirement, device detailed information access address, etc. (see 9.1.1).

6.3 Device (group) discovery

An IGRS Device may listen to the device online advertisement information on the specified multicast addresses to discover an IGRS Device (group) existing on the network.

An IGRS Device may send a device (group) search requests to the specified multicast address on all networks to which it connects. The IGRS Device (group) that receives this request and matches the search criteria shall send the unicast device search response to the requesting IGRS Device (see 9.6 and 9.8).

6.4 Device pipe setup

Except for an UDP-based multicast search (specified in IETF RFC 768) and unicast response interaction among IGRS Devices, which do not require the setup of a Device Pipe, all the other interactions among IGRS Devices shall be based on the setup of a Device Pipe.

After discovering the target IGRS Device on the network, an IGRS Device may establish a secure pipe with the target IGRS Device through the Device Pipe Management mechanism (see 9.2).

Through the Device Pipe setup mechanism, the IGRS specification encapsulates and simplifies the TCP connection setup and management process during IGRS application interaction. Only one Device Pipe exists between two IGRS Devices at the same time.

Each IGRS Device should control the pipe setup requests from other IGRS Devices by establishing an authorised/accessible IGRS Device list and the maximum number of concurrent pipes.

Device pipes are divided into two categories.

- Secure Device Pipe: the Device Pipe setup is based on the security mechanism commonly supported by the two devices, which is authenticated by this mechanism.
- Unsecure Device Pipe: the Device Pipe is setup without the security mechanism commonly supported by the two devices (not authenticated).

Each IGRS Device online advertisement message shall contain information about the security mechanism and relevant parameters, which are used in the process of Device Pipe setup. An IGRS Device may resolve the target IGRS Device online advertisement message in order to select an appropriate security mechanism to establish a secure Device Pipe with the target IGRS Device.

The IGRS Device pipe mechanism includes Device Pipe setup and Device Pipe management. Two IGRS Devices shall determine the mutual trust relationship through the process of secure Device Pipe setup.

After the Device Pipe setup is completed, two IGRS Devices may detect the online status of each other interactively by exchanging device online-status detection request and response messages.

6.5 Device group setup and join

IGRS Device groups are further divided into three categories: global peer-to-peer device group, specified peer-to-peer device group, and centralised device group.

After coming online, an IGRS Device is placed in a global peer-to-peer device group by default. All IGRS Devices belong to the same global peer-to-peer device group (see 9.5).

An IGRS Device may join or set up a peer-to-peer device group through device configuration. The security interaction rules of all IGRS Devices in the same peer-to-peer device group shall be determined upon negotiation (outside of this IGRS specification) through the secure pipe (see 9.5.2). The master device shall control the joining of other devices through setting the accessible/authorised device list and the maximum number of devices in the device group.

The user may set up a centralised device group and designate a device as the master of this device group. The master device is responsible for periodically sending centralised device group advertisement messages by multicast (see 9.5.3).

After discovering the existence of a centralised device group in the network, an IGRS Device may initiate a joining request to the master device of this centralised device group. The master device shall decide whether to approve the request according to relevant rules. The application shall define specific grouping requirements based on needs, for example, setting up association according to service and device types, etc. IGRS framework supports the establishment of device groups that provide certain sets of services or consist of similar device capabilities on similar physical devices.

An IGRS Device may setup and/or join a specified peer-to-peer device group and a centralised device group.

An IGRS Device may setup and/or join many specified peer-to-peer device groups and centralised device groups at the same time. The creation of each device group: global peer-to-peer device group, specified peer-to-peer device group and centralised device group shall be determined by the requirements indicated in specific guidelines as defined in IGRS basic applications.

6.6 Service discovery

An IGRS Client may listen to the assigned multicast address for the service online advertisement message to discover an existing IGRS Service on the network.

An IGRS Client may initiate the request for a detailed device description document of the target device. The detailed device description document includes the device service list, which describes for all services the names, types, and the URL addresses that point to detailed service description documents (see 10.4).

An IGRS client may send a service search request via a specified multicast address to all IGRS Devices in the network. The search message should contain the type, name, and other attributes of the services. The IGRS Device that receives the service search request and contains the service matching the search criteria shall send an appropriate service search response message to the IGRS Client (see 10.2).

An IGRS Client may send a service search request to a specified IGRS Device through a Device Pipe. The IGRS Device that receives this search request shall return the corresponding search results according to the search conditions (see 10.2).

An IGRS Client in a centralised device group may send a service online/offline event subscription request to the specified IGRS Device through a Device Pipe. The subscription criteria shall be described in the subscription request (see 10.3).

6.7 Session setup

After discovering a service on the target IGRS Device through the service discovery mechanism, an IGRS Client may establish a supporting environment for subsequent service access through the session setup mechanism based on a Device Pipe (see 10.5).

An IGRS Service may attain service access control by gathering the authorised user and device access list of this service. IGRS Services should set up the maximum number of concurrent access.

IGRS Device relationships are determined by whether devices are located in the same device group, whether the IGRS Client device is the trusted device of the IGRS Service device, and whether the IGRS Client device is the specified trusted device of the IGRS Service device.

During the session setup process, the IGRS Device providing an IGRS Service shall acquire information about the IGRS Client device, such as the user ID and client authentication information. Based on this information, the IGRS Client access right shall be authenticated. Meanwhile, concurrent control shall be realised according to the concurrent support capabilities of IGRS Services. Only one session shall exist between one IGRS Client and one IGRS Service at the same time. After a session is successfully established, an IGRS Client may invoke IGRS Services including function invocation, data query, data retrieval, etc. through invoking the interfaces of the IGRS Services.

6.8 Service invocation

After establishing a session with an IGRS Service, an IGRS Client may invoke IGRS Services through pre-defined service invocation mechanisms, in accordance with the interfaces described in the IGRS Service description document (see 10.6).

6.9 Session termination

When an IGRS Client finishes invoking IGRS Services, the IGRS Client shall terminate the session with the IGRS Service (see 10.6). An IGRS Service may initiate the termination of a session with an IGRS Client.

6.10 Device/Service online/offline event subscription

After two IGRS Devices have set up a Device Pipe, the IGRS Client may initiate a device/service online/offline event subscription request with the other party. The IGRS Device that receives the subscription request shall determine whether to accept the subscription; if it accepts the subscription, it then sends the result back to the IGRS Device containing the IGRS Client using an event subscription response message.

An IGRS Client may initiate an event subscription request to the target IGRS Device in the same peer-to-peer device group to subscribe to the service status change on the target device, including service online/offline.

An IGRS Client may initiate an event subscription request to the master device in a centralised device group to subscribe to events in the device group, including device online/offline and service online/offline.

The valid period of event subscription shall be determined by the IGRS Device that accepts the event subscription request. The requesting IGRS Client shall be notified in the subscription response message. An IGRS Client may initiate a subscription renewal request during the valid period. If the IGRS Client does not send the subscription renewal request during the valid period of event subscription, the IGRS Device that accepted the event subscription shall delete all relevant subscription records.

6.11 Device/Service online/offline event notification

When detecting the occurrence of events matching the subscription conditions, the IGRS Device that accepted the event subscription request shall send an event notification message to the IGRS Client through a Device Pipe. The content of the events shall be described in the message.

6.12 Device/Service online/offline event unsubscription

After successfully achieving event subscription with a target IGRS Device, an IGRS Client may send an unsubscription request to the target IGRS Device through a Device Pipe within the valid period of the event subscription.

6.13 Pipe disconnection

After a Device Pipe has been successfully established between two IGRS Devices, the pipe may be disabled under the following conditions.

a) A device goes offline: after the Device Pipe has been established, the pipe between the two devices shall be disabled immediately when the IGRS Device at either end detects the other end has gone offline.

b) Pipe idle times out: after the Device Pipe is established, if no session-based message on the pipe exists and no device interaction occurs in a specific time interval except for a detection message of online/offline device status, the IGRS Device at either end shall disable the pipe.

c) One end initiated disconnection: after the Device Pipe is established, the device at either end may send a pipe disconnection notification to the device at the other end of the pipe before the device has gone offline or is powered off for other management purposes. The device that received the message shall disable the pipe immediately.

After the Device Pipe is disabled, the message based on previous pipe shall require setting up new pipe between the devices.

6.14 Device group dismiss and secession

When all IGRS Devices in the same peer device group quit from this device group, the peer-to-peer device group shall be dismissed (see 9.5.2).

The master device in a centralised device group may dismiss the device group according to management needs. When dismissing a centralised device group, the master device shall multicast a dismissing notification message to the network (see 9.5.3). After joining a centralised device group, an IGRS Device shall continuously monitor the online status of the device group. If one IGRS Device in a centralised device group receives the dismissing notification or detects the master device offline through the device online detection mechanism, it shall mean that the centralised device group has been dismissed. The device shall then quit from this device group.

An IGRS Device in a device group may initiate secession from the device group in the following cases.

a) Device offline: after an IGRS Device joins the device group, if due to network disconnection or device power-off or for other device management reasons, it fails to multicast the device online advertisement to the network on a regular basis, it shall be regarded as device offline. The device offline status shall mean that the device quits from all device groups that it has joined.

b) Initiated quit: after joining a centralised device group, an IGRS Device may send a Device Group secession request to the master device in the group through the Device Pipe in order to quit from the group. The quitting of an IGRS Device from a peer-to-peer device group is only an attribute change of the device itself, and does not require interaction with other peer devices in the group. An IGRS Device shall send a device online advertisement

message when leaving from the peer device group, and delete the related group's information from the online advertisement message (see 9.1.1).

6.15 Device offline

If IGRS Device A receives an offline advertisement message from IGRS Device B or does not receive an online advertisement message from device B again before the valid period of the last online advertisement of device B ends, A should regard B as offline.

The device online advertisement is based on the multicast mechanism. Due to the unreliable nature of a multicast mechanism, the device online status detection mechanism should be used to detect whether a device is online. After the Device Pipe between two IGRS Devices is established, either IGRS Device should send a device online detection message through the pipe to detect whether the IGRS Device on the other side is online. The IGRS Device that receives the message shall send an online detection response message back to the IGRS Device that initiates the detection. If the IGRS Device that initiates the detection fails to receive the response within a specified period, it should regard that device as offline.

If an IGRS Device in a centralised device group receives offline event notification about a particular device from the master device, that device should be regarded as offline.

7 IGRS message framework

7.1 Overview

The messages among IGRS Devices include three types: request messages, response messages, and notification messages.

The IGRS request/response messages shall conform to the request/response model of HTTP/1.1 (see RFC 2616), and also support all IGRS request/response messages through extended HTTP/1.1 command types and self-defined HTTP/1.1 headers (see 7.2, 7.3).

IGRS defines device notification messages that are uni-directional through extended HTTP/1.1 command types and self-defined HTTP/1.1 headers (see 7.2).

When authentication and encryption are needed, the framework of IGRS request/response and notification message differs from the framework without encryption (see 7.4).

When the transmission of IGRS request/response messages and notification messages do not adopt a chunked encoding transfer mechanism (RFC 2616), the message length (including the message body and message header) shall not exceed 20 kbyte. A message exceeding 20 kbyte shall adopt the chunked encoding transfer mechanism (RFC 2616) for transmission.

When an IGRS Device encounters an unknown field in a message, that field shall be ignored.

7.2 IGRS request/notification message structure

The IGRS request and notification message structure is shown in Table 1.

Table 1 – IGRS request and notification message

Message	Field explanation
POST[GET][NOTIFY] /IGRS HTTP/1.1	Extended HTTP command, including POST, NOTIFY and GET; the value of Request-URI part is /IGRS
Host: *Host IP Address:port*	HTTP header: required field
Content-type: text/xml: charset =utf-8	HTTP header: optional field
Content-Length: message body length	HTTP header, can be omitted when the message body is empty or transmitted by chunked encoding transfer mechanism
Man:"http://www.igrs.org/spec1.0" ; ns= 01	IGRS extension header namespace definition: required field. See Clause B.1
01-IGRSVersion: 1.0	IGRS protocol version: required field
01-IGRSMessageType: *IGRS message type*	String: required field
01-SourceDeviceId: *ID of the device sends this message*	URI, defined in 8.1.2: required field
01-TargetDeviceId: *ID of the device receives this message*	URI, defined in 8.1.2: required field
01-SequenceId:*Sequence Number of this request message*	32 bit unsigned Int (0 is reserved): required field
Man:"http://www.w3.org/2002/12/soap-envelope"; ns=02	SOAP extension namespace definition: required field
02-SoapAction: *action identifier*	Refer to detailed message definition: required field
	Shall be blank line
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	
<SOAP-ENV:Body>	
<schema xmlns = "http://www.w3.org/2001/XMLSchema" xmlns:IGRS = "http://www.igrs.org/spec" targetNamespace ="http://www.igrs.org/spec1.0">	Name space definition, see Clauses B.9 and B.1
<element name="DeviceOperation" type="IGRS:DeviceOperationType minOccurs="0" maxOccurs="1"> <element name="Session" type="IGRS:SessionType"minOccurs="0" maxOccurs="1"> <complexType name="DeviceOperation"> <!--*Device interaction*--> </complexType> <complexType name="Session">	IGRS SOAP usage. ClientId, ServiceId and SequenceID are 32-bit unsignedInt (0 is reserved).

Message	Field explanation
<attribute name="request" type="IGRS:requestType" use="required"/> <!--*Session interaction*--> </complexType> <simpleType name="requestType"> <restriction base="xsd:string"> <xsd:enumeration value="NeedResponse"/> <xsd:enumeration value="NoResponse"/> </restriction> </simpleType> </schema>	
</SOAP-ENV:Body>	
</SOAP-ENV:Envelope>	
NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.	

7.3 IGRS response message structure

The structure of IGRS response messages is shown in Table 2.

Table 2 – IGRS response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Host: *Host IP Address:port*	HTTP message header
Content-type: text/xml; charset = utf-8	HTTP message header
Content-Length: message body length	HTTP header: can be omitted when the message body is empty or transmitted by chunked encoding transfer mechanism
Cache-Control: no-cache=ext;	HTTP message header
Ext:	HTTP message header
MAN:"http://www.igrs.org/spec1.0" ; ns= 01	IGRS namespace definition based on HTTP extension framework [IETF RFC 2774], see Clause B.1
01-IGRSVersion: 1.0	IGRS protocol version
01-IGRSMessageType: *IGRSMsgType*	string, refer to detailed message definition
01-SourceDeviceId: *ID of the device sends this message*	URI, defined in 8.1.2
01-TargetDeviceId: *ID of the device receives this message*	URI, defined in 8.1.2
01-ReturnCode: *Response status code*	Optional: defined in Clause 11
01-AcknowledgedId:*Acknoledged sequence number*	32 bit unsigned Int (0 is reserved), same with the SequenceId in request message
Man:"http://www.w3.org/2002/12/soap-envelope"; ns=02	SOAP HTTP extension identifier and header
02-SoapAction: *action identifier*	Refer to detailed message definition
	Shall be empty

Message	Field explanation
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	
<SOAP-ENV:Body>	
<schema xmlns = “http://www.w3.org/2001/XMLSchema” xmlns:IGRS = “http://www.igrs.org/spec” targetNamespace =“http://www.igrs.org/spec1.0”>	Name space, see Clauses B.9 and B.1
<element name=”DeviceOperation” type=”IGRS:DeviceOperationType minOccurs=”0” maxOccurs=”1”> <element name=”Session” type=”IGRS:SessionType”minOccurs=”0” maxOccurs=”1”> <complexType name=”DeviceOperation”> <!--*Device interaction*--> </complexType> <complexType name=”Session”> <!--*Session interaction*--> </complexType>	IGRS SOAP usage
</SOAP-ENV:Body>	
</SOAP-ENV:Envelope>	
NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.	

7.4 IGRS message based on secure device pipe

7.4.1 Message authentication code generation

For devices implementing message interaction based on one of the following two mechanisms, the message authentication code shall be generated from the content of the IGRS header and the HTTP message body using the sequence and format defined in Table 3 and the corresponding algorithm in the adopted security mechanism, as follows:

a) based on an identity authentication and message authentication mechanism;

b) based on an identity authentication, encrypted message transmission, and message authentication mechanism of the Symmetric Key System (see 8.1.5.2 and 8.1.5.3).

The IGRS HTTP-extended header for labelling message authentication code is as follows:

01-MAC: hexadecimal ASCII character string to which the message authentication code corresponds.

Table 3 – Message authentication code

Message	Field explanation
01-IGRSVersion: IGRS/1.0	Not including CR for newline and blank
01-IGRSMessageType: *real message type*	Not including CR for newline and blank
01-SequenceId: *real sequence ID*	Not including CR for newline and blank; response message is 01-AcknowledgeId: real sequence ID; absent if there is no SequenceId/Acknowledged
01-SourceDeviceId: *real source device ID*	Not including CR for newline and blank
01-TargetDeviceId: *real target device ID*	Not including CR for newline and blank
Message body	HTTP message body content; message body format; absent if there is no message bogy
NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.	

7.4.2 Message encryption

For devices implementing message interation based on the identity authentication, encrypted message transmission, and message authentication mechanism of the Symmetrical Key System (see 8.1.5.2, 8.1.5.3), the encryption object is the original HTTP message body, which is based on secure Device Pipe interaction.

The content-length and content-type of the HTTP message header based on secure Device Pipe interaction shall follow the following rules.

- Content-type: application/octet-stream.
- Content-length: content length of message body after encryption.
- Encryption shall not be needed if there is no message body.

8 IGRS device and service description

8.1 IGRS device description

8.1.1 Device description template

An IGRS Device describes relevant information such as device name, type, and service list with one XML document that conforms to the IGRS Device Description Template specified below.

An IGRS Client may get the device description documents of other IGRS Devices through the request mechanism specified in 9.3.

The IGRS Device description template is defined as follows (see also Clause B.5):

<?xml version="1.0" encoding="utf-8"?>

<schema targetNamespace="http://www.igrs.org/igrs/DeviceTemplate"

xmlns:igrs="http://www.igrs.org/igrs/DeviceTemplate"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"

attributeFormDefault="unqualified">

<element name="root">

```
            <complexType>
                <sequence>
<element name="deviceTemplateVersion" type="igrs:deviceTemplateVersionType"/>
                    <element name="device" type="igrs:deviceTemplateType"/>
                    <element name="protocolVersion"
type="igrs:protocolVersionType"/>
                </sequence>
            </complexType>
        </element>
        <complexType name="deviceTemplateVersionType">
            <sequence>
                <element name="major" type="unsignedInt"/>
                <element name="minor" type="unsignedInt"/>
            </sequence>
        </complexType>
        <complexType name="protocolVersionType">
            <sequence>
                <element name="major" type="unsignedInt"/>
                <element name="minor" type="unsignedInt"/>
            </sequence>
        </complexType>
        <complexType name="deviceTemplateType">
            <sequence>
                <element name="deviceType" type="anyURI"/>
                <element name="deviceName" type="string"/>
                <element name="manufacturer" type="string"/>
                <element name="manufacturerURL" type="anyURI" minOccurs="0"/>
                <element name="modelDescription" type="string" minOccurs="0"/>
```

```
                    <element name="modelName" type="string"/>
                    <element name="modelNumber" type="string" minOccurs="0"/>
                    <element name="modelURL" type="anyURI" minOccurs="0"/>
                    <element name="serialNumber" type="string" minOccurs="0"/>
                    <element name="UDN" type="anyURI"/>
                    <element name="DeviceIconURL" type="anyURI"/>
<element name="joinedGroupList" type="igrs:joinedGroupListType"
minOccurs="0"/>
<element name="deviceSecurityIdList"
type="igrs:deviceSecurityIdListType"/>
                     <element name="serviceList"
type="igrs:serviceListType"minOccurs="0"/>
            </sequence>
        </complexType>
        <complexType name="joinedGroupListType">
            <sequence>
                    <element name="deviceGroup" type="igrs:deviceGroupType"/>
            </sequence>
        </complexType>
        <complexType name="deviceGroupType">
            <sequence>
                    <element name="deviceGroupId" type="anyURI"/>
                    <element name="deviceGroupName" type="string"/>
            </sequence>
        </complexType>
        <complexType name="deviceSecurityIdListType">
            <sequence>
```

```
                <element name="deviceSecurityId" type="anyURI" minOccurs="0"
maxOccurs="5"/>
            </sequence>
        </complexType>
        <complexType name="serviceListType">
            <sequence>
                <element name="service" type="igrs:serviceTemplateType"
maxOccurs="unbounded"/>
            </sequence>
        </complexType>
        <complexType name="serviceTemplateType">
            <sequence>
                <element name="serviceType" type="anyURI"/>
                <element name="serviceId" type="unsignedInt"/>
                <element name="serviceName" type="string"/>
                <element name="serviceSecurityIdList"
type="igrs:serviceSecurityIdListType"/>
<!-- Application developer may add additional service interfaces, but should use different
namespaces to make a distinction -->
            </sequence>
        </complexType>
        <complexType name="serviceSecurityIdListType">
            <sequence>
                <element name="serviceSecurityId" type="anyURI" minOccurs="0"
maxOccurs="3"/>
            </sequence>
        </complexType>
</schema>
```

8.1.2 Device identifier

An IGRS Device identifier is a globally unique ID, called a UUID, so no two devices have the same ID.

The format of an IGRS Device identifier is described as follows by adopting the ABNF (IETF RFC 2234):

<IGRSDeviceIdURN> ::= “urn:” <IGRSNS> “:” <IGRSIDVAL>

<IGRSNS>::= “IGRS:Device:DeviceId”

<IGRSIDVAL>::= 8 <URN chars>“-”4<URN chars>“-”4<URN chars>“-”4<URN chars> “-”12<URN chars>

<URN chars>::= <trans>

<trans>::= <upper> | <lower> | <number>

<upper>::= “A” | “B” | “C” | “D” | “E” | “F”

<lower>::= “a” | “b” | “c” | “d” | “e” | “f”

<number>::=“0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” |

“8” | “9”

<URN chars> is case-insensitive.

8.1.3 Device group identifier

An IGRS Device group identifier is a globally unique ID so no two groups have the same ID.

The format of an IGRS Device group identifier is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceGroupIdURN>::= “urn:”<IGRSNS>“:”<IGRSIDVAL>

<IGRSNS>::=“IGRS:DeviceGroup:DeviceGroupId”

<IGRSIDVAL>::= 8 <URN chars> “-”4<URN chars> “-”4<URN chars> “-”4<URN chars>

“-”12<URN chars>

<URN chars>::=<trans>

<trans>::=<upper> | <lower> | <number>

<upper>::= “A” | “B” | “C” | “D” | “E” | “F”

<lower>::= “a” | “b” | “c” | “d” | “e” | “f”

<number>::=“0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” |

“8” | “9”

<URN chars> is case-insensitive.

8.1.4 Device type identifier

An IGRS Device type identifier is used for identifying an IGRS Device set containing the same functions. The length of the IGRS Device type identifier shall be no more than 127 bytes.

The format of the IGRS Device type identifier is shown as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceTypeURN>::=“urn:”<IGRSNS>“:”<IGRSSingleType>|<IGRSTypeList>

<IGRSNS>::=“IGRS:Device:DeviceType”

<IGRSTypeList>=<IGRSSingleType>*<IGRSTypeVal>

<IGRSTypeVal> =<ConnectionSign><IGRSSingleType>

<IGRSSingleType> = <NAME>

<NAME>::=1*16<URN chars>

<URN chars>::=<trans>

<trans>::=<upper>|<lower>|<number>|<other>

<upper>::=“A” | “B” | “C” | “D” | “E” | “F” | “G” | “H” |

“I” | “J” | “K” | “L” | “M” | “N” | “O” | “P” |

“Q” | “R” | “S” | “T” | “U” | “V” | “W” | “X” |

“Y” | “Z”

<lower>::=“a” | “b” | “c” | “d” | “e” | “f” | “g” | “h” |

“i” | “j” | “k” | “l” | “m” | “n” | “o” | “p” |

“q” | “r” | “s” | “t” | “u” | “v” | “w” | “x” |

“y” | “z”

<number>::=“0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” |

“8” | “9”

<other>::=“-” | “.” | “_”

<ConnectionSign>::=“;”

<URN chars> is case-insensitive.

8.1.5 Device security mechanism descriptor

8.1.5.1 Null security mechanism among devices

When two IGRS Devices that set up the Device Pipe adopt this security mechanism, no security assurance measures shall be taken. That is, when the pipe is established, ID authentication for the device shall not be performed. Messages in the channel are transmitted without using IGRS specifications for confidentiality and/or integrity. This mechanism shall be the required default security mechanism for IGRS Devices and may be used in an application scenario where data security is not needed, for example, when communicating on a wired medium within a house.

IGRS based on this device security mechanism is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN> :: =“urn:”<IGRSNS>

<IGRSNS> :: =“IGRS:DeviceSecurity:NULL”

8.1.5.2 Identity authentication and message authentication mechanism based on symmetric-key cryptosystem

Two IGRS Devices may use a pre-shared key to authenticate each other and to establish a secure Device Pipe. The authentication process is a bi-directional challenge/response process. The message transmitted on the Device Pipe shall not be encrypted, but shall be checked for integrity and authenticity. This mechanism is optional for IGRS Devices and may be applied in small networks and devices with weak cipher computing capabilities. This security option might be useful in a scenario where bogus or unqualified devices shall be excluded from a device group. This security method protects the devices that are communicating, not the data transmitted.

The challenge value in the process of the ID authentication in this mechanism shall be generated based on the secure random number generator (RNG) algorithm or pseudo-random number generator (PRNG) algorithm and shall be regenerated each time when performing ID authentication. The design and implementation of the RNG or PRNG algorithm conforming to the general detection standard is outside the scope of this standard. Please see ISO/IEC 18031:2005 for the required elements and characteristis of RNG and PRNG.

The response value in the process of ID authentication in this mechanism is generated through the secure hash algorithm based on the other party’s challenge value and the preset shared key. The generation method is defined as follows.

- The secure hash algorithm function is Hash (x).
- The response value is Hash (Challenge || Key), where Challenge is the challenge value and Key is the shared key by both parties.

The secure Hash algorithms supported by this mechanism are SHA-1 and SHA-256 specified in ISO/IEC 10118-3:2004.

The mechanism used for ensuring the integrity and authentication of the message that is transmitted in the pipe is the following.

a) Calculate one digest value for the target message stream transmitted by the pipe using the secure Hash function.

b) Calculate the digest value again after concatenating with the preset-shared key.

c) Attach the digest value to 01-MAC.

d) Send the digest value and 01-MAC to the receiver.

This digest value shall be used as the basis for determining the integrity and authentication of the message. The algorithm format is: Hash (Hash (message) || Key), where, the Message is the message to be transmitted by the pipe and the Key is the shared key for both parties.

After receiving the message, the receiver shall perform the same calculation process, and shall compare the calculation results with the added digest value. If they are equal, the message is transmitted by the transmitter without modification; otherwise, the message shall be regarded as untrusted.

The precondition for implementing this mechanism is that both protocol participants should share one key and establish an appropriate system outside of IGRS to complete the secure distribution and management of the shared key.

IGRS based on this device security mechanism, is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::=“urn:” <IGRSNS> “:” <IGRSVAL1>“:”<IGRSVAL2>

<IGRSNS>::=“IGRS:DeviceSecurity:PreSharedKey_MAC”

<IGRSVAL1>::=“SHA-1”|”SHA-256”

<IGRSVAL2>::=“SHA-1”|”SHA-256”

IGRSVAL1 is the description of the identity authentication algorithm; IGRSVAL2 is the description of the message authentication algorithm.

8.1.5.3 Identity authentication, encrypted message transmission and message authentication mechanism based on symmetric-key cryptosystem

When two IGRS Devices adopt this security mechanism to establish the Device Pipe, ID authentication shall be performed for the two devices based on a pre-shared key. The authentication process is a bi-directional challenge/response process. This security mechanism provides increased confidentiality, integrity, and authenticity for messages transmitted on the device channel. This mechanism is optional for IGRS Devices and may be applied in small networks and devices with weak cipher computing capabilities. This security option might be useful in a scenario where bogus or unqualified devices must be excluded from a device group. This security method may be useful for a wireless medium because it protects the devices that are communicating and the data transmitted,

The challenge value, Key, or Nonce in this mechanism shall be generated using the secure RNG or PRNG algorithm. The secure hash algorithm shall be used to implement the challenge/response authentication protocol and message integrity/authenticity protocol. Meanwhile, the method of ensuring the integrity and authenticity of the messages transmitted in the pipe is also the same as the second security mechanism (specified in 8.1.5.2).

In addition, this security mechanism uses the symmetric-key cipher algorithm to ensure the confidentiality of messages. The PipeKey that implements the message confidentiality is generated through a Hash algorithm in the process of protocol authentication based on the two challenge values generated by the bi-directional challenge/response protocol and the pre-shared keys. The generation method is:

a) if the length of the bit string output by the Hash (Hash (Challenge 1|| Challenge 2)|| Key) is greater than or equal to the length of key of the symmetric-key cipher algorithm, the bit string with the same length as the key length drawn from left to right shall be used as the PipeKey;

b) if the length of the bit string output by the Hash (Hash (Challenge 1|| Challenge 2)|| Key) is less than the length of key of the symmetric-key cipher algorithm, the Hash

(Hash(Hash(Challenge1 || Challenge 2)|| Key) || Hash (Hash (Hash (Challenge 1 || Challenge 2) || Key)|| Key)) is generated;

c) if the length is greater than or equal to the length of key of the symmetric-key cipher algorithm, the bit strings with the same length as the key length drawn from left to right shall be used as the PipeKey.

The rest shall follow the same rules. The PipeKey, which can be shared by both parties, shall then be generated.

The PipeKey is only valid during the life cycle of the current pipe.

When implementing a symmetric-key cipher block algorithm (see ISO/IEC 18033-3), CBC (Cipher Block Chaining) mode for encryption/decryption is used. When the message is divided into blocks, the length of the last block may be insufficient and will need padding. The padding method is the following: assume the encrypted message length is l (unit is byte) and the block size is n bytes. 1 byte will be needed to represent the length of the padding (unit is byte) and u bytes for padding (every bit is 0)., u is the least non-negative integer (0 = < u < 16) that satisfies $(l+1+u) \bmod n = 0$. The last byte in the block will be used to represent the padding length (u +1). When using the CBC mode to encrypt/decrypt messages, the first message block will perform an XOR operation with an Initialisation Vector (IV), then apply block encryption algorithm to encrypt the message. The result will become the second "IV." This operation will continue for the remaining blocks. Here, IV is randomly selected by the encryption side before the encryption algorithm begins. The length of the IV is the block length used in the block encryption algorithm. IV will be placed in front of the encryption result of the first block to send to the receiving end. The mathematical formulas used by CBC encryption/decryption mode are: encryption: $C_1 = E_K(P_1 \oplus IV)$, $C_2 = E_K(P_2 \oplus C_1)$, …, $C_N = E_K(P_N \oplus C_{N-1})$; decryption: $P_1 = IV \oplus D_K(C_1)$, $P_2 = C_1 \oplus D_K(C_2)$, …, $P_N = C_{N-1} \oplus D_K(C_N)$. Here N means the number of blocks in the message.

The symmetric-key cipher algorithms that this security mechanism currently supports are

- AES-128-128 (128-bit block-length and 128-bit key-length), specified in ISO/IEC 18033-3,
- 3DES-64-112 (64-bit block-length and 112-bit key-length), specified in ISO/IEC 19790.

The implementation of this security mechanism to enable the secure distribution and management of a shared key is outside the scope of this standard.

IGRS based on this device security mechanism, is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::="urn:"<IGRSNS>":"<IGRSVAL1>":"<IGRSVAL2>":"

<IGRSVAL3>

<IGRSNS>::="IGRS:DeviceSecurity:PreSharedKey_Cipher_MAC"

<IGRSVAL1>::="SHA-1"|"SHA-256"

<IGRSVAL2>::="AES-128-128"|"3DES-64-112"

<IGRSVAL3>::="SHA-1"|"SHA-256"

IGRSVAL1 is the description of the identity authentication algorithm; IGRSVAL2 is the description of the message encryption algorithm; IGRSVAL3 is the description of the message authentication algorithm.

8.1.5.4 Identity authentication, encrypted message transmission and message authentication mechanism based on public-key cryptosystem

When two IGRS Devices adopt this security mechanism to establish a Device Pipe, the two devices shall be authenticated based on the PKI X.509 v3 (ISO/IEC 9594-8:2005). The authentication process is a bi-directional challenge-response process. This security mechanism provides increased confidentiality, integrity, and authenticity for messages transmitted on the device channel. This mechanism is optional for IGRS Devices and may be applied in medium or large networks and devices with strong cipher computing capabilities. This security option might be useful in a scenario where bogus or unqualified devices shall be excluded from a device group. This security method may be useful for a wireless medium because it protects the devices that are communicating and the data transmitted.

NOTE A public key is an aymmetric key system where one key is published and the other is secret. This could support mass-marketed devices all containing the same public key, while the secret key is embedded in or downloaded into a controller communicating with the mass-marketed devices.

The challenge value, Key, or Nonce in this mechanism shall be generated using the secure RNG or PRNG algorithm. The secure hash algorithm shall be used to enable the message integrity/authenticity protocol.

For integrity and authenticity when processing a message, the secure hash algorithm shall be used to calculate the message to produce a digest. Then a private key shall be used corresponding to the PKI X.509 certificate to perform signature operation on the digest value. The signature result shall be the criterion for message integrity and authenticity. The signature results together with the message shall be sent to the message receiver.

The PipeKey generating method for message confidentiality in this mechanism is different from the third security mechanism. The following two design methods may be considered.

a) Simple method: The Pipe Key is generated from negotiations during the bi-directional challenge/response authentication process. The detailed procedure is the following.
   - The response party in the first round selects one random number (reserved for the PipeKey).
   - Encrypts it with the other party's public key.
   - Sends the encryption results, response value, and the challenge value for the second round together to the other party.
   - The response party in the second round uses its private key to decrypt the encrypted value and to perform a secure Hash algorithm for the concatenation of the decryption result and the challenge value. Meanwhile, the response party uses the private key to generate a signature for the challenge value and sends the signature result together with the result of the Hash algorithm to the other party.
   - The receiver party determines that the other party has obtained accurate (used as the PipeKey) random number through verifying the signature and the comparison with the result of the Hash algorithm.
   - The random number shall be used as the shared key PipeKey for both parties to enable message confidentiality.
b) DH method: Adopt the Diffie-Hellman key switching protocol to negotiate the key PipeKey shared by both parties. In this case, the Diffie-Hellman public key cipher model and relevant algorithm shall be constructed.

With the third security mechanism, the PipeKey is only valid during the current life cycle of the pipe and the PipeKey updating mechanism is not considered either.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES. The public key algorithms, which this security mechanism currently supports, are RSA-1024 [IETF RFC 3447] and ECC-192 (IEEE 1363-2000).

The enabling of this security mechanism needs the support of the PKI X.509 v3 certificate and CA (Certificate Authority) system. Due to the large scalability of the system, when the IGRS protocol network is enabled within a small scope, the developers of protocols can enable one simplified and sufficient X.509 CA system. When the IGRS protocol network is implemented in a large-scale network, a complicated X.509 CA system is needed, or an external existing X.509 CA system is another option.

It is required that the IGRS protocol use the public key certificate function of this security mechanism according to the X.509 CA system interface standard.

IGRS based on this device security mechanism is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::=“urn:”<IGRSNS>“:”<IGRSVAL1>“:”<IGRSVAL2>“:”

<IGRSVAL3>“:”<IGRSVAL4>

<IGRSNS>::=“IGRS:DeviceSecurity:PKICertificate_Cipher_MAC”

<IGRSVAL1>::=“RSA-1024”|”ECC-192”|”ElGamal”

<IGRSVAL2>::=“Simple”|”DH”

<IGRSVAL3>::=“AES-128-128”|”3DES-64-112”

<IGRSVAL4>::=“SHA-1”|”SHA-256”

IGRSVAL1 is the description of the public key algorithm; IGRSVAL2 is the description of the negotiation key algorithm; IGRSVAL3 is the description of encrypted message transmission algorithm, and IGRSVAL4 is the description of message authentication algorithm.

8.1.5.5 Identity authentication, encrypted message transmission and message signature mechanism based on trusted third party

This security mechanism provides the ID authentication service for both parties that set up the pipe based on the trusted third party. The three parties complete the authentication process according to the Kerberos v5 protocol [IETF RFC 1510]. The key generation and distribution function for message confidentiality, integrity and authenticity shall be provided in the protocol ticket generated by the trusted third party. Generally, this mechanism should allow handling with the confidentiality, integrity, and authenticity of messages transmitted in the pipe, but also can support the two parties of a pipe to negotiate and select according to the security requirement stipulated by the device advertisement. This security mechanism is optional for IGRS Devices and may be applied in large enterprise networks and systems dedicated to support security. This security option might be useful in a scenario where bogus or unqualified devices must be excluded from a device group. This security method may be useful for a wireless medium because it protects the devices that are communicating and the data transmitted.

NOTE The involvement of a trusted third party to supply keys is commonly used by web browsers in Internet commerce to block secure communications with bogus web sites masquerading as known vendors. A trusted key is important if home control is offered as a managed service from a web-based vendor.

There are two methods for implementing the Kerberos protocol specified in IETF RFC 1510:

One is based on the shared key or symmetric key encryption mechanism. In this case, within the scope of the IGRS protocol network, each IGRS Device simply shares one key with the trusted third party (authentication service provider) in advance, without pre-setting the shared key between the two interworking devices.

The other implementation method is based on the public key system or PKI X.509 CA certificate system. Although this method could be designed as the fourth security mechanism, computation limitations should be considered. Since some low-power computing platforms may have difficulty implementing the public key encryption algorithm, the mixed mode is generally considered. In the mixed mode the platform with the more powerful computing capability adopts the PKI X.509 certificate system, while the platform with the less powerful computing capability adopts the symmetric-key cipher system. The trusted third party that offers the authentication services should be able to support both cipher systems. The security mechanism between the device and the trusted third party should adopt the IGRS Device security mechanism defined in this document.

The trusted third party-based authentication services and the Kerberos protocol are highly efficient only when an IGRS network has a large scale (especially the certificate system based on PKI X.509). The concept of the centralised device group also exists in the IGRS protocol network, and the master device may assume the role of the trusted third party. Therefore, for this security mechanism, the enabling method using the symmetric-key cipher system is recommended.

From the perspective of the development trend in security technology, authentication services based on the trusted third party are becoming main stream. Meanwhile, the technical complexity is great. Therefore, this security mechanism shall be considered as the basis or direction for future extensions of the security technology of the IGRS protocol. At present, it is recommended that a simplified and effective method be chosen to enable this security mechanism, i.e., only support the minimum set of the Kerberos v5 mechanism attributes.

This security mechanism also needs the support of the secure RNG or PRNG algorithm, secure hash algorithm, and symmetric-key cipher algorithm.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES. The public key algorithms, which this security mechanism currently supports, is RSA-1024.

IGRS based on this device security mechanism, is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::=“urn:”<IGRSNS>“:”<IGRSVAL1>“:”<IGRSVAL2>“:”

<IGRSVAL3>“:”<IGRSVAL4>“:”<IGRSVAL5>“:”<IGRSVAL6>

<IGRSNS>::=“IGRS:DeviceSecurity:3rdPartyAuthenService”

<IGRSVAL1>::=“3rdAuthProtocol_Kerberosv5:AES-128-128”|”3rdAuthProtocol_Kerberosv5:RSA-1024”

<IGRSVAL2>::=“AuthenServiceProvider”

<IGRSVAL3>::=<IGRSDeviceIdURN> (see 8.1.2)

<IGRSVAL4>::= Authentication service identifier (see 8.2.2)

<IGRSVAL5>::=“AES-128-128”|”3DES-64-112”

<IGRSVAL6>::=“SHA-1”|”SHA-256”

IGRSVAL1 is the description of the Kerberos5-based authentication algorithm of the trusted third party mechanism. IGRSVAL2, 3 and 4 compose the description of the third party authentication service provider, where IGRSVAL3 is the identifier for device that provides

authentication service and IGRSVAL4 is the service identifier of such service on a device that is identified by IGRSVAL3. IGRSVAL5 is the description of the encrypted message transmission algorithm for both end devices of a secure pipe, and IGRSVAL6 is the description of message signature algorithm for both end devices of a secure pipe.

8.1.5.6 Encryption algorithm descriptors supported by security mechanism

The five security mechanisms described in 8.1.5.1 through 8.1.5.5 are categorized below by the following levels of complexity from the lowest (level 0) to the highest complexity (level 4). Level 0 is the default and means no security is used.

0. NULL
1. PreSharedKey_MAC
2. PreSharedKey_Cipher_MAC
3. PKICertificate_Cipher_MAC
4. 3rdPartyAuthenService

Every device shall implement the null security mechanism. The other four mechanisms are optional. However, if a device implements a security level above 0, it shall implement all lower-complexity levels. For an example, a device that implements PKICertificate_Cipher_MAC must also implement PreSharedKey_Cipher_MAC, PreSharedKey_MAC, and NULL mechanisms. When a communications session via a device pipe is initiated, the following procedure shall be executed to ensure that the same and most secure mechanism is used by the devices sharing the pipe:

a) Using the NULL level of security, each device shall announce to the other devices or to the master, if present, the highest level of security it can support;

b) the device that initiates the secure pipe setup or the master shall determine the highest level supported by the devices sharing the pipe based on the security algorithm parameters it receives from the other devices;

c) the initiating device or the master shall send to each device a secure pipe setup request message to use this level for subsequent communications via the pipe;

d) a secure pipe setup response message shall be returned from the recipient device to the initiating device or to the master verifying the successful device pipe setup at the specified security level.

Table 4 describes five security mechanisms defined by IGRS and the corresponding relationships between the security protocol and the security algorithm used in each mechanism.

Table 4 – Device security mechanism protocol algorithm

Device security mechanism	Security algorithm type involved	Algorithm related to device security mechanism
PreSharedKey_MAC	AuthenticationAlgorithm	AuthenticationAlgorithm:SHA-1
		AuthenticationAlgorithm:SHA-256
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256
PreSharedKey_Cipher_MAC	AuthenticationAlgorithm	AuthenticationAlgorithm:SHA-1
		AuthenticationAlgorithm:SHA-256
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256
PKICertificate_Cipher_MAC	PublicKeyAlgorithm	PublicKeyAlgorithm:RSA-1024
		PublicKeyAlgorithm:ECC-192
		PublicKeyAlgorithm:ElGamal
	KeyNegotiationAlgorithm	KeyNegotiationAlgorithm:Simple
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256
3rdPartyAuthenService	3rdAuthProtocol_Kerberosv5	3rdAuthProtocol_Kerberosv5:AES-128-128
		3rdAuthProtocol_Kerberosv5:RSA-1024
	AuthenServiceProvider	AuthenServiceProvider: DeviceID||ServiceID
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
	MACAlgorithm	MACAlgorithm:SHA-1
		MACAlgorithm:SHA-256

The level of complexity design principle for implementing security mechanisms shall also apply to each of the security algorithm options in the third column of Table 4. These options are listed in order from lowest to highest complexity. All lower complexity security algorithms must be implemented whenever a higher complexity security algorithm is chosen. For example, if the AuthenticationAlgorithm of the PreSharedKey_MAC is required, then a device that implements SHA-256 shall also implement SHA-1.

8.2 IGRS service description

8.2.1 Service description template

The IGRS Service description is based on WSDL2.0 (specified in http://www.w3.org/TR/wsdl20/). In addition, the IGRS Service description adds specific requirements that include:

- extension of WSDL portType, add support to IGRS Service information description;
- transmission Binding based on Device Pipe.

This standard (see Clause 10) extends the WSDL service description template specified in W3C WSDL 2.0. The IGRS Service description template is shown as follows (see also Clause B.3):

```
<wsdl:definitions xmlns:igrs="http://www.igrs.org/igrs/ServiceDescription"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://www.igrs.org/igrs/ServiceDescription" name="igrs">

        <wsdl:types>

<schema targetNamespace=http://www.igrs.org/igrs/ServiceDescription

xmlns="http://www.w3.org/2001/XMLSchema"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

attributeFormDefault="unqualified" elementFormDefault="qualified">

                        <!-- IGRS Service Attribute Types -->

                        <attributeGroup name="occurs">

                                <attribute name="minOccurs" type="xsd:nonNegativeInteger"
use="optional" default="1"/>

                                <attribute name="maxOccurs">

                                        <simpleType>

                                                <union
memberTypes="xsd:nonNegativeInteger">

                                                        <simpleType>

                                                                <restriction base="NMTOKEN">

                                                                        <enumeration
value="unbounded"/>

                                                                </restriction>

                                                        </simpleType>

                                                </union>

                                        </simpleType>

                                </attribute>

                        </attributeGroup>
```

```
<element name="serviceAttribute" type="igrs:IGRSDataType"/>

<element name="serviceAttributeValues"
type="igrs:IGRSDataValueType"/>

<complexType name="IGRSDataType">

<sequence>

<element name="ServiceId"type="xsd:unsignedInt"/>

<element name="ServiceName"type="xsd:string"/>

<element name="ServiceType"type="igrs:ServiceType"/>

<element
name="ServiceSecurityId"type="igrs:ServiceSecurityId"/>

<!--other elements for serviceAttribute -->

</sequence>

<attribute name="name" type="xsd:NCName"/>

<attribute name="type" type="xsd:QName"/>

<attributeGroup ref="igrs:occurs"/>

<attribute name="mutability" use="optional" default="dynamic">

<simpleType>

<restriction base="xsd:string">

<enumeration value="static"/>

<enumeration value="dynamic"/>

</restriction>

</simpleType>

</attribute>

<attribute name="notifiable" use="optional" type="xsd:boolean"
default="false"/>

</complexType>

<complexType name="IGRSDataValueType">

<sequence>
```

```
                    <any namespace="##any" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <!-- findServiceAttribute Message Types -->
            <element name="findServiceAttributeByNames"
type="igrs:NCNamesType"/>
            <element name="subscribeServiceAttributeByNames"
 type="igrs:NCNamesType"/>
            <element name="findServiceAttribute"
type="igrs:findServiceAttributeType"/>
            <complexType name="findServiceAttributeType">
                <sequence>
                    <element name="findExpression"
type="igrs:findExpressionType"/>
                </sequence>
            </complexType>
            <complexType name="findExpressionType">
                <sequence>
                    <element ref="igrs:findServiceAttributeByNames"/>
                </sequence>
            </complexType>
            <complexType name="NCNamesType">
                <sequence>
                    <element name="name" type="xsd:NCName"
maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <element name="findServiceAttributeResponse"
```

```
type="igrs:findServiceAttributeResponseType"/>
                    <complexType name="findServiceAttributeResponseType">
                        <sequence>
                            <element name="result" type="igrs:resultType"/>
                        </sequence>
                    </complexType>
                    <complexType name="resultType">
                        <sequence>
                            <element name="serviceAttributeValues"
type="igrs:serviceAttributeValuesType" minOccurs="0"/>
                            <element name="returnCode" type="xsd:unsignedInt"/>
                        </sequence>
                    </complexType>
                    <complexType name="serviceAttributeValuesType">
                        <sequence maxOccurs="unbounded">
                            <element name="name" type="xsd:NCName"/>
                            <element name="value"
type="igrs:IGRSDataValueType"/>
                        </sequence>
                    </complexType>
                    <!-- Subscription Message Types -->
                    <element name="subscribe" type="igrs:subscribeType"/>
                    <complexType name="subscribeType">
                        <sequence>
                            <element name="subscribeExpression"
type="igrs:subscribeExpressionType"/>
                            <element name="location" type="xsd:anyURI"/>
                            <element name="expirationTime" type="xsd:dateTime"/>
```

```
                </sequence>
            </complexType>
            <complexType name="subscribeExpressionType">
                <sequence>
                    <element
ref="igrs:subscribeServiceAttributeByNames"/>
                </sequence>
            </complexType>
            <element name="renewSubscription">
                <complexType>
                    <sequence>
                        <element name="subscriptionId"
type="xsd:unsignedInt"/>
                        <element name="location" type="xsd:anyURI"/>
                        <element name="expirationTime"
type="xsd:dateTime"/>
                    </sequence>
                </complexType>
            </element>
            <element name="unsubscribe">
                <complexType>
                    <sequence>
                        <element name="subscriptionId"
type="xsd:unsignedInt"/>
                    </sequence>
                </complexType>
            </element>
            <element name="subscribeResponse"
type="igrs:subscribeResponseType"/>
            <complexType name="subscribeResponseType">
```

```
                    <sequence>
                        <element name="subscriptionId"
type="xsd:unsignedInt"/>
                        <element name="terminationTime" type="xsd:dateTime"
minOccurs="0"/>
                        <element name="returnCode" type="xsd:unsignedInt"/>
                    </sequence>
                </complexType>
                <!-- Notification Message Types -->
                <element name="sendNotification" type="igrs:sendNotificationType"/>
                <complexType name="sendNotificationType">
                    <sequence>
                        <element name="serviceAttribute"
type="igrs:serviceAttributeTypes"/>
                    </sequence>
                </complexType>
                <complexType name="serviceAttributeTypes">
                    <sequence maxOccurs="unbounded">
                        <element name="name" type="xsd:NCName"/>
                        <element name="value"
type="igrs:IGRSDataValueType"/>
                        <element name="subscriptionId"
type="xsd:unsignedInt"/>
                    </sequence>
                </complexType>
                <import namespace="http://schemas.xmlsoap.org/wsdl/"/>
                <element name="IGRSportType" type="igrs:portTypeType"/>
                <complexType name="portTypeType">
                    <complexContent>
```

```
                                    <extension base="wsdl:tPortType">
                                        <sequence>
                                            <any namespace="##other"
minOccurs="0"
 maxOccurs="unbounded"/>
                                        </sequence>
                                        <attribute name="extends" use="optional">
                                            <simpleType>
                                                <list itemType="QName"/>
                                            </simpleType>
                                        </attribute>
                                        <anyAttribute namespace="##other"/>
                                    </extension>
                                </complexContent>
                            </complexType>
                    </schema>
            </wsdl:types>
            <message name="findServiceAttributeInputMessage">
                    <part name="parameters" element="igrs:findServiceAttribute"/>
            </message>
            <message name="findServiceAttributeOutputMessage">
                    <part name="parameters" element="igrs:findServiceAttributeResponse"/>
            </message>
            <message name="subscribeInputMessage">
                    <part name="parameters" element="igrs:subscribe"/>
            </message>
            <message name="subscribeOutputMessage">
                    <part name="parameters" element="igrs:subscribeResponse"/>
```

```
        </message>
        <message name="sendNotificationInputMessage">
                <part name="parameters" element="igrs:sendNotification"/>
        </message>
        <IGRSportType name="subscribeServiceAttributePortType">
                <operation name="subscribe">
                        <input message="igrs:subscribeInputMessage"/>
                        <output message="igrs:subscribeOutputMessage"/>
                </operation>
                <igrs:serviceAttribute name="notifiableServiceAttributeName"
type="igrs:IGRSDataType"
minOccurs="0" maxOccurs="unbounded" mutability="dynamic" notifiable="true"/>
                <igrs:staticServiceAttributeValues>
                        <igrs:subscribeExtensibility
inputElement="igrs:subscribeServiceAttributeByNames"/>
                </igrs:staticServiceAttributeValues>
        </IGRSportType>
        <IGRSportType name="notifyServiceAttributePortType">
                <operation name="sendNotification">
                        <input message="igrs:sendNotificationInputMessage"/>
                </operation>
        </IGRSportType>
        <IGRSportType name="findServiceAttributePortType">
                <operation name="findServiceAttribute">
                        <input message="igrs:findServiceAttributeInputMessage"/>
                        <output message="igrs:findServiceAttributeOutputMessage"/>
                </operation>
```

```
                <igrs:serviceAttribute name="serviceAttributeName"
type="xsd:NCName" minOccurs="0"
maxOccurs="unbounded" mutability="dynamic"/>
                <igrs:staticServiceAttributeValues>
                        <igrs:findServiceAttributeExtensibility
inputElement="igrs:findServiceAttributeByNames"/>
                </igrs:staticServiceAttributeValues>
        </IGRSportType>
</wsdl:definitions>
```

This International Standard defines a transport protocol based on a Device Pipe to extend the WSDL2.0 transport protocol binding. This transport binding, based on a Device Pipe, is an extension to SOAP binding, and is identified by transport="http://www.igrs.org/igrs/igrspipe", see Clause B.4.

The following are the definitions of a transport-binding template in the service description of Clause 10, see also Clause B.3:

```
<?xml version="1.0" encoding="utf-8"?>
<schema targetNamespace="http://www.igrs.org/igrs/ServiceDescription"
xmlns:igrs="http://www.igrs.org/igrs/ServiceDescription"
xmlns="http://www.w3.org/2001/XMLSchema">
        <element name="binding" type="igrs:bindingType"/>
        <complexType name="bindingType">
                <attribute name="transport" type="igrs:transportMethod" use="optional"/>
                <attribute name="style" type="igrs:styleChoice" use="optional"/>
        </complexType>
        <simpleType name="transportMethod">
                <restriction base="string">
                        <enumeration value=http://www.igrs.org/igrs/igrspipe/>
                </restriction>
        </simpleType>
        <simpleType name="styleChoice">
```

```
        <restriction base="string">
            <enumeration value="rpc"/>
            <enumeration value="document"/>
        </restriction>
</simpleType>
<element name="operation" type="igrs:operationType"/>
<complexType name="operationType">
        <attribute name="soapAction" type="anyURI" use="optional"/>
        <attribute name="style" type="igrs:styleChoice" use="optional"/>
</complexType>
<element name="body" type="igrs:bodyType"/>
<complexType name="bodyType">
        <attribute name="encodingStyle" type="anyURI" use="optional"/>
        <attribute name="parts" type="NMTOKENS" use="optional"/>
        <attribute name="use" type="igrs:useChoice" use="optional"/>
        <attribute name="namespace" type="anyURI" use="optional"/>
</complexType>
<simpleType name="useChoice">
        <restriction base="string">
            <enumeration value="literal"/>
            <enumeration value="encoded"/>
        </restriction>
</simpleType>
<element name="fault" type="igrs:faultType"/>
<complexType name="faultType">
        <complexContent>
            <restriction base="igrs:bodyType">
```

```
                    <attribute name="parts" type="NMTOKENS" use="prohibited"/>
                </restriction>
            </complexContent>
        </complexType>
        <element name="address" type="igrs:addressType"/>
        <complexType name="addressType">
            <attribute name="location" type="anyURI" use="required"/>
        </complexType>
</schema>
```

8.2.2 Service identifier

The IGRS Service identifier shall be used for identifying one service registered on a device and shall be allocated by the device where the IGRS Service is located. This identifier shall be unique on this device. The service identifier and the device identifier shall be jointly used for addressing of services in the IGRS Device interworking protocol.

The IGRS Service identifier is one 32-bit unsigned integer. The 0 is reserved but not used for IGRS Service identification.

8.2.3 Service type identifier

The IGRS Service type shall be used to define the globally unique identifier for different kinds of standard services, and to identify the methods and event mechanisms of some services. The length of the IGRS Service type identifier shall not exceed 127 bytes.

The format of the IGRS Service type identifier is described as follows by adopting the ABNF (RFC 2234):

```
<IGRSServiceTypeURN> ::= "urn:" <IGRSNS> ":" <IGRSIDVAL>

<IGRSNS>::="IGRS:Service:ServiceType"

<IGRSIDVAL>::=1*<URN chars>

<URN chars>::=<trans>

        <trans>::=<upper> | <lower> | <number> | <other>

<upper>::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" |"I" |

"J" | "K" | "L" | "M" |"N" | "O" | "P" | "Q" | "R" |

"S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

<lower>::= "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" |"i" |
```

“j” | “k” | “l” | “m” | “n” | “o” | “p” | “q” | “r”|

“s” | “t” | “u” | “v” | “w” | “x” | “y” | “z”

<number>::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” |”8” |

“9”

<other>::= “-” | “.” | “_”

<URN chars> is case-insensitive.

IGRS allows developers to define proprietary service categories. These proprietary services shall be described according to the following URN-based method based on ABNF (RFC 2234):

<IGRSSelfDefineServiceTypeURN> ::= “urn:” <IGRSNS> “:” <IGRSIDVAL>

<IGRSNS>::= “IGRS:service:servicetype-p”

<IGRSIDVAL>::=1*<URN chars>

<URN chars>::= <trans>

<trans>::= <upper> | <lower> | <number> | <other>

<upper>::= “A” | “B” | “C” | “D” | “E” | “F” | “G” | “H” | “I” |

“J” | “K” | “L” | “M”

| “N” | “O” | “P” | “Q” | “R” | “S” | “T” | “U” | “V”|

“W” | “X” | “Y” | “Z”

<lower>::= “a” | “b” | “c” | “d” | “e” | “f” | “g” | “h” | “i” |

“j” | “k” | “l” | “m” | “n” | “o” | “p” |”q” | “r” |

“s” | “t” | “u” | “v” | “w” | “x” | “y” | “z”

<number>::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” |

“8” | “9”

<other>::= “-” | “.” | “_”

<URN chars> is case-insensitive.

8.2.4 Service access control

The IGRS Service should achieve access control by managing the user identity and the relationship between a user-located device and a service-located device. For any service, the IGRS users should be divided into three categories: AnyUser, TrustedUser (user that has passed identity authentication), and DesignatedTrustedUser (designated user that has passed identity authentication). The relationship between a user-located device and a service-located device shall be categorised into five types:

a) AnyDevice;

b) GroupDevice (device that is in the same device group with the service-located device);

c) TrustedDevice (device that has passed identity authentication);

d) TrustedGroupDevice (device that has passed identity authentication and is in the same device group with the service-located device);

e) DesignatedTrustedDevice (designated device that has passed identity authentication).

An IGRS Service should ensure the service access control logic through the setup of a service access control policy described in Table 5.

Table 5 – Service access control policy

	AnyDevice	GroupDevice	TrustedDevice	TrustedGroup-Device	Designated TrustedDevice
AnyUser	Access allowed or denied	Access allowed or denied	Access allowed or denied	Access allowed or denied	Access allowed or denied
TrustedUser	Access allowed or denied	Access allowed or denied	Access allowed or denied	Access allowed or denied	Access allowed or denied
Designated TrustedUser	Identifier list of users with access permission	Identifier list of users with access permission	Identifier list of users with access permission	Identifier list of users with access permission	Identifier list of users with access permission

Whether the relationship between a user-located device and a service-located device is trusted depends on the device identity authentication through the process of Device Pipe setup. Devices that pass such authentication are trusted devices.

Users that pass identity authentication through session setup process are trusted users to a service.

8.2.5 Identity authentication mechanism of service sccess control

8.2.5.1 Use of identity authentication mechanism

The IGRS Protocol has designed four user identity authentication mechanisms for services (see 8.2.5.2 to 8.2.5.5). An IGRS Service shall advertise the requirements for user identity authentication mechanism in the “ServiceSecurityId” of the service description document (see 8.2.1).

8.2.5.2 No identity authentication

With this mechanism, services do not implement user identity authentication through the process of session setup. This mechanism is optional for an IGRS Service.

Based on this service security mechanism, IGRS is described as follows by adopting the ABNF (RFC 2234):

<IGRSServiceSecurityURN>::=“urn:” <IGRSNS>

<IGRSNS>::=“IGRS:ServiceSecurity:NULL”

8.2.5.3 Identity authentication mechanism based on symmetric-key cryptosystem

With this mechanism, services implement user identity authentication based on a pre-shared key through the process of session setup. This mechanism is optional for an IGRS Service. In other words, services implement authentication by judging whether the other party is aware of the pre-shared key. The protocol authentication process is completed based on the Challenge/Response interaction process. This mechanism is optional for an IGRS Service.

In order to prevent authentication protocol messages from fabrications or repeated attacks, the Challenge shall be generated on the basis of a secure random number generator (RNG) algorithm or a pseudo random number generator (PRNG) algorithm in the protocol implementation process. Also, the Challenge shall be regenerated in each execution of the protocol.

The response value in the authentication protocol is generated by the other party's Challenge and pre-shared key with a secure Hash algorithm. The generation method is defined as: given that secure Hash algorithm function is Hash (X), the Response value is Hash (Challenge || Key), where Challenge is challenge value and key is the shared key for both parties of protocol. The secure Hash algorithm supported by this mechanism are SHA-1 and SHA-256.

The precondition of this mechanism is that both parties to the authentication protocol need to pre-share a same key. The implementer of this mechanism should ensure the secure distribution and administration of the pre-shared key.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES.

This service security mechanism is described as follows by adopting the ABNF (RFC 2234):

<IGRSServiceSecurityURN>::="urn:" <IGRSNS> ":" <IGRSVAL1>":"<IGRSVAL2>

<IGRSNS>::="IGRS:ServiceSecurity:PreSharedKey"

<IGRSVAL1>::="SHA-1"|"SHA-256"

<IGRSVAL2>::="AES-128-128"|"3DES-64-112"

IGRSVAL1 is the description of authentication algorithm and IGRSVAL2 is the description of message encryption algorithm.

8.2.5.4 Identity authentication mechanism based on public-key cryptosystem

With this mechanism, when service is invoked, the service provider implements the identity authentication on the service applicant based on the PKI X.509 v3 certificate. The authentication protocol process is implemented based on Challenge/Response process. This mechanism is optional for an IGRS Service.

This security mechanism also needs a security random number generator or a pseudo random number generator to generate a challenge value, key, or Nonce.

This authentication mechanism implementation needs the support from the PKI X.509 v3 certification and CA (Certificate Authority) system. It is suggested that the IGRS protocol developer implement a public key certificate function for this security mechanism according to X.509 CA system interface standard.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES. The public key algorithms, which this security mechanism currently supports, are RSA-1024, ECC-192, and ElGamal.

This service security mechanism is described as follows by adopting the ABNF (RFC 2234):

<IGRSServiceSecurityURN>::="urn:"<IGRSNS>":"<IGRSVAL1>":"<IGRSVAL2>

<IGRSNS>::="IGRS:ServiceSecurity:PKICertificate"

<IGRSVAL1>::=“RSA-1024”|”ECC-192”|”ElGamal”

<IGRSVAL2>::=“AES-128-128”|”3DES-64-112”

IGRSVAL1 is the description of public key algorithm, and IGRSVAL2 is the description of encrypted message transmission algorithm.

8.2.5.5 Identity authentication mechanism based on trusted third party

With this mechanism, the service implements user identity authentication based on the trusted third party. The three parties (the trusted third party, service, and user) implement authentication a protocol process based on the Kerberos v5 protocol. This mechanism is optional for an IGRS Service.

The implementation of the Kerberos protocol is mainly based on a pre-shared key encryption mechanism. An IGRS Service or user needs to pre-share a key only with the trusted third party. The distribution and administration of the key is implemented by the trusted third party.

The symmetric-key cipher algorithms, which this security mechanism currently supports, are AES and 3DES. The public key algorithms, which this security mechanism currently supports, are RSA-1024 and ECC-192.

This service security mechanism is described as follows by adopting the ABNF (RFC 2234):

<IGRSDeviceSecurityURN>::=“urn:”<IGRSNS>“:”<IGRSVAL1>“:”<IGRSVAL2>“:”

<IGRSVAL3>“:”<IGRSVAL4>“:”<IGRSVAL5>“:”<IGRSVAL6>

<IGRSNS>::=“IGRS:ServiceSecurity:3rdPartyAuthenService”

<IGRSVAL1>::=“3rdAuthProtocol_Kerberosv5:AES-128-128”|”
3rdAuthProtocol_Kerberosv5:RSA-1024”

<IGRSVAL2>::=“AuthenServiceProvider”

<IGRSVAL3>::=<IGRSDeviceIdURN> (see 8.1.2)

<IGRSVAL4>::= Authentication service identifier (see 8.2.2)

<IGRSVAL5>::=“AES-128-128”|”3DES-64-112”

IGRSVAL1 is the description of the Kerberos5-based authentication algorithm of the trusted third party mechanism. IGRSVAL2, 3 and 4 compose the description of the third party authentication service provider, where IGRSVAL3 is the identifier for the device that provides authentication service and IGRSVAL4 is the service identifier of such service on a device that is identified by IGRSVAL3. IGRSVAL5 is the description of the message (between client and service) transmission encryption algorithm.

8.2.5.6 Encryption algorithm descriptor based on authentication mechanism

Table 6 defines the above mentioned authentication mechanisms and the corresponding encryption algorithm descriptor.

Table 6 – Device authentication mechanisms and the corresponding encryption algorithm descriptor

Authentication mechanism	Algorithm/Protocol descriptor	Algorithm name descriptor
PreSharedKey	AuthenAlgorithm	AuthenAlgorithm:SHA-1
		AuthenAlgorithm:SHA-256
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
PKICertificate	PublicKeyAlgorithm	PublicKeyAlgorithm:RSA-1024
		PublicKeyAlgorithm:ECC-192
		PublicKeyAlgorithm:ElGamal
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112
3rdPartyAuthenService	Kerberosv5	Kerberosv5:AES-128-128
	AuthenServiceProvider	AuthenServiceProvider:IGRS DeviceID|| IGRS ServiceID
	EncryptAlgorithm	EncryptAlgorithm:AES-128-128
		EncryptAlgorithm:3DES-64-112

The “level of complexity” design principle for implementing security mechanisms shall also apply to each of the security algorithm options in the third column of Table 6. These options are listed in order from lowest to highest complexity. All lower complexity security algorithms shall be implemented whenever a higher complexity security algorithm is chosen. For example, if the AuthenticationAlgorithm of the PreSharedKey_MAC is required, then a device that implements SHA-256 shall also implement SHA-1.

8.3 IGRS client description

The IGRS Client identifier shall be used to identify an IGRS Client. The IGRS Client identifier type is unsigned integer and 0 is reserved.

8.4 IGRS user description

The IGRS user identifier shall be used to identify an IGRS user. The IGRS user identifier type is string and the maximum length is 127 bytes.

9 IGRS device grouping

9.1 Device advertisement

9.1.1 Device online advertisement

The IGRS Device advertisement mechanism shall adopt the ISDP protocol mechanism that is derived from Simple Service Discovery Protocol (SSDP) defined in ISO/IEC 29341-1 and specified in Annex A (ISDP). In the active state, each IGRS Device shall send the device online advertisement message to the multicast address 239.255.255.250:3880 in multicast. The device online advertisement message belongs to the IGRS notification-type message. The form of this message is shown in Table 7.

Table 7 – Device online advertisement

Message	Field explanation
NOTIFY * HTTP/1.1	HTTP extended command line
Host: 239.255.255.255.250:3880	Required
Cache-control:max-age=*Max advertising valid time*	Required field, in seconds, min time is 3 s, when device receives this message and the time has expired, then the advertising device may be offline
Location: url of device description document	Required field, support unsecure pipes among IGRS Devices to get device description documents. The value of this field is the URL of the documents. When device does not support device description documents access through non-secure pipes, then this field shall be: http://www.igrs.org/device, see Clause B.7.
NT: uuid: Advertising device ID	Required, defined in 8.1.2
NTS:isdp:alive	Required
SERVER: OS/version IGRS/1.0 product/version	Required
USN: uuid: Advertising device ID	Required field, see 8.1.2
Man:"http://www.igrs.org/spec1.0"; ns= 01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS protocol version number
01-IGRSMessageType: DeviceOnlineAdvertisement	Required field
01-SourceDeviceId: Advertising device ID	Required field, see 8.1.2
01-DeviceType: Device Type	Required field, device type identifier, see 8.1.4
01-DeviceName: Device Name	Required field, device name, string
01-ConfigId:Indicator of device configuration change	Required field, type is ASCII code of 32bit unsignedInt (0 reserved), this field will indicate when service and other configuration changes in this device. 01-ConfigId shall be increased by one (1) whenever there is a configuration change. This field will return to 1 when upper limit is reached.
01-BootId:Indicator of device reboot	Required field, type is ASCII code of 32bit unsignedInt (0 reserved), this field should differentiate new advertising message and the advertising message before reboot when device are rebooted. 01-BootId value shall be increased by one (1) whenever there is a reboot. This field will return to 1 when upper limit is reached.
01-DeviceGroupIdList:Device Group ID list	Required field, device group ID, URI, defined in 8.1.3, spaced by ";" this field is empty when device doesn't join non-global group.
01-DeviceSecurityIdList: Security mechanism ID of device.	Required field, URI, defined in 8.1.5, the security mechanisms are spaced by ";".
01-SecureListenerList: IP address and Port list used to set up security Device Pipe.	Optional field, type is string, format is "IP address:port", muliti IP address:port are spaced by ";" listening port is fixed at 3880
01-ListenerList: IP address and Port list used to set up non-security Device Pipe.	Required field, type is string, format is "IP address:port", muliti IP address:port are spaced by ";" one of the port number in the list shall be 3 880

When the configuration information on the IGRS Device changes, the online advertisement message shall be re-sent to the network and the value of the 01-ConfigId shall be increased by 1.

When the IGRS Device restarts, the online advertisement message shall be re-sent to the network and the value of the 01-BootId shall be increased by 1.

Each IGRS Device may detect and listen to other IGRS Device online advertisement message at address: 239.255.255.250:3880.

If an IGRS Device fails to re-send the online advertisement message within the specified maximum time interval, the other IGRS Device shall regard the device as offline from the network. The IGRS Device shall re-send online advertisement message within the maximum time interval.

9.1.2 Device offline advertisement

When an IGRS Device needs to exit the network due to power-off or for a management reason, it should send the device offline advertisement message to the address of 239.255.255.250:3880 in multicast to notify other IGRS Devices. The format of the device offline advertisement message is shown as the Table 8.

Table 8 – Device offline advertisement

Message	Field Explanation
NOTIFY * HTTP/1.1	Extended command line
Host:239.255.255.250:3880	Required field
NT:uuid:*Advertising device ID*	Required field, type is URI, value refer to 8.1.2
NTS:isdp:byebye	Required field
USN:uuid:*Advertising device ID*	Required field, type is same as NT
Man:"http://www.igrs.org/spec1.0"; ns= 01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field
01-IGRSMessageType: DeviceOfflineAdvertisement	Required field
01-SourceDeviceId: *Advertising device ID*	Required field, type is URI, defined in 8.1.2

9.2 Device pipe management

9.2.1 Unsecure device pipe setup

If two devices have no security mechanism commonly supported and cannot implement interactive authentication, they shall set up a Device Pipe without a security guarantee using an unsecure Device Pipe.

An unsecure Device Pipe setup process is a TCP connection setup process between two devices. If the TCP connection is successfully set up between two devices, an unsecure Device Pipe shall be regarded as successfully set up between two devices; otherwise, the setup shall be regarded as a failure. The IGRS Device shall use 01-ListenerList field in the Device Online Advertisement message to describe the TCP connection address and the ports to construct unsecure Device Pipe.

9.2.2 Unsecure device pipe maintenance

Unsecure Device Pipe teardown may be caused by the following.

- If any device is offline, the unsecure Device Pipe related to this device shall be torn down.
- If any device initiates a tear-down process of a TCP-related connection, the relevant unsecure Device Pipe shall be torn down.

9.2.3 Secure device pipe setup

9.2.3.1 Overview

An IGRS secure Device Pipe setup process is shown in Figure 4. It includes three phases as follows:

a) pipe setup initialisation;

b) identity authentication and pipe key negotiation phase, and

c) pipe setup confirmation.

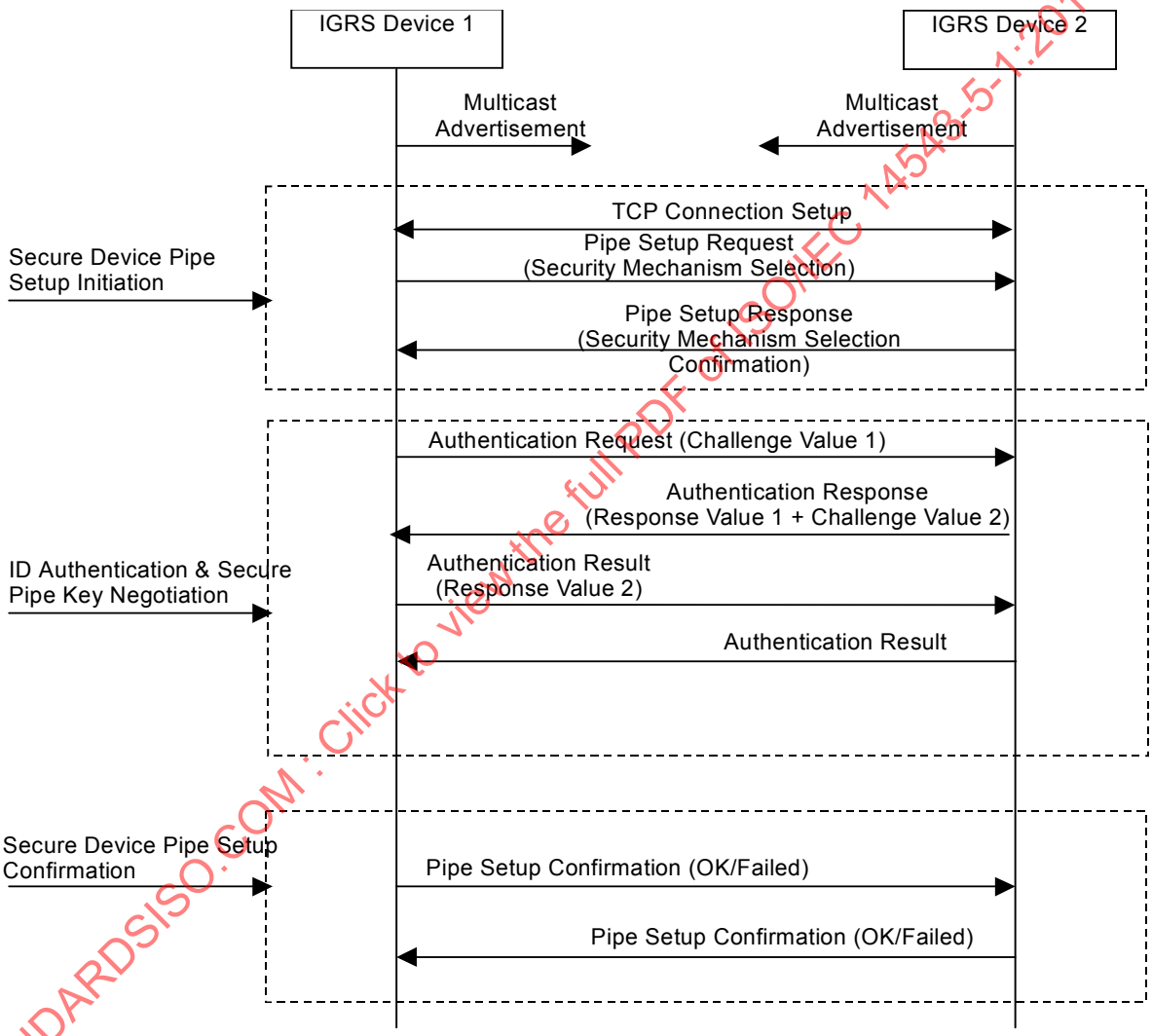

Figure 4 – Secure device pipe setup

Subclauses 9.2.3.2 to 9.2.3.4 describe the device interaction logic in the three phases of a secure Device Pipe setup, among which IGRS Device 1 is used for the IGRS Device that initialises a pipe setup request and IGRS Device 2 is the target IGRS Device in the pipe setup request.

9.2.3.2 Secure device pipe setup initialisation

9.2.3.2.1 Secure device pipe setup initialisation – General

IGRS Device 1 implements the pipe interaction-security mechanism negotiation process with IGRS Device 2 in this phase.

Pipe setup initiation includes three steps are as follow.

Step 1: IGRS Device 1 initialises a TCP connection setup request to the device connection address available for IGRS Device 2. If the setup is successful, it shall continue to Step 2; otherwise, the pipe setup is a failure.

Step 2: Device 1 implements the intersection operation between the security mechanism and a relevant encryption algorithm list (note that list sequence reflects priority) supported in the advertisement message of Device 2, and the security mechanism and relevant encryption algorithm list supported by Device 1. If the intersection is not empty, Device 1 shall further select an optimal security mechanism and corresponding encryption algorithm based on the security attributes of both parties, and then send it to Device 2 through pipe setup request. If intersection is empty, Device 1 cannot send a pipe setup request to Device 2, and pipe setup has failed.

Step 3: After Device 2 receives the pipe setup request from Device 1, it confirms whether the security mechanism and corresponding encryption algorithm selected by Device 1 are satisfied with the current security requirement and status of Device 2. If Device 2 is satisfied, it shall send a pipe setup response to Device 1, including the security mechanism and corresponding encryption algorithm. If it is not satisfied, it shall continue to Phase 3, where it sends a pipe setup failure confirmation to Device 1.

The pipe setup request and response message formats correspond to the device security mechanism adopted. Detail descriptions are defined in 9.2.3.2.2 to 9.2.3.2.5.

9.2.3.2.2 Device security mechanism based on symmetric-key authentication and message authentication mechanism

When the device security interaction mechanism adopts a symmetric-key cryptosystem, the pipe setup request and response message formats are described as follow (see Table 9).

Table 9 – Pipe setup request based on symmetric-key cryptosystem

Message	Field Explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host:Target host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeRequest	Required field
01-TargetDeviceId:target device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId:source device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId:Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-DeviceSecurityId:"urn:IGRS:DeviceSecurity:PreSharedKey_MAC"	Required field, type is URI

A pipe setup response shall be returned to the requesting device within 30 s after the request is sent. The message format is shown as follows (see Table 10).

Table 10 – Pipe setup response based on symmetric-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field
01-IGRSMessageType:CreatePipeResponse	Required field
01-TargetDeviceId:Target device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId:source device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode:Response status code of pipe creation	Required field, status code refer to Clause 11
01-AcknowledgeId:Response Message Sequence ID	Required field, same with the sequence ID in request message

9.2.3.2.3 Device security mechanism based on symmetric-key authentication, encrypted message transmission, and authentication mechanism

When the security interaction mechanism used between devices adopts symmetric-key cryptosystem, the pipe setup request and response message formats are as follow (see Table 11).

Table 11 – Pipe setup request based on symmetric-key authentication, encrypted message transmission, and authentication mechanism

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-DeviceSecurityId:"urn:IGRS:DeviceSecurity:PreSharedKey_Cipher_MAC"	Required field, type is URI

A pipe setup response shall be returned to the requesting device within 30 s after the request is sent. The message format is as follows (see Table 12).

Table 12 – Pipe setup response based on symmetric-key authentication, encrypted message transmission, and authentication mechanism

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code of pipe creation	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

9.2.3.2.4 Security mechanism between devices is authentication, encrypted message transmission, and authentication mechanism based on public-key cryptosystem

In this mechanism, Device 1 shall enclose an associated PKI X.509 v3 certificate for Device 2.

Step 3: After Device 2 receives the security mechanism and corresponding encryption algorithm confirmation request selected by Device 1,

a) it confirms whether the security mechanism and corresponding encryption algorithm selected by Device 1 is accommodated by the current security requirement and status of Device 2,

b) it authenticates the creditability (or validity) of the PKI X.509 certificate of Device 1.

If all meet the requirement, Device 2 shall send the pipe setup response to Device 1 and enclose the PKI X.509 certificate of Device 2. After Device 1 receives the security mechanism selection confirmation and the PKI X.509 certificate of Device 2, it shall verify the validity of this certificate; if it is valid, it shall continue to Phase 2 of pipe setup. If the conditions above cannot be met, it shall continue to Phase 3, sending pipe setup failed confirmation to the partner.

The pipe setup request and response message formats in this security mechanism are as follow (see Table 13).

Table 13 – Pipe setup request based on authentication, encrypted message transmission, and authentication mechanism of public-key cryptosystem

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-DeviceSecurityId:"urn:IGRS:DeviceSecurity:PKICertificate_Cipher_MAC"	Required field, type is URI
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Message body	Required field, certificate based on x.509

A pipe setup response shall be returned to requesting device within 30 s after the request is sent, and the message format is as follows (see Table 14).

Table 14 – Pipe setup response based on authentication, encrypted message transmission, and authentication mechanism of public-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code of pipe creation	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Message body	Required field, certificate based on x.509

9.2.3.2.5 Device security mechanism based on trusted third party's authentication, encrypted message transmission, and authentication mechanism

In this mechanism, Device 1, based on the Kerberos protocol, first requests that a third party provide a security identity authentication service to generate the security interaction Ticket between Device 1 and Device 2. Then it checks the validity of the related Ticket retrieved; if it is valid, it shall continue to Phase 2, otherwise it shall proceed to Phase 3, sending a pipe setup failed confirmation to Device 2.

The pipe setup request and response message formats in this security mechanism are as follows (see Table 15).

Table 15 – Pipe setup request based on trusted third party authentication, encrypted message transmission, and authentication mechanism

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-DeviceSecurityId:"urn:IGRS:DeviceSecurity:3rdPartyAuthenService"	Required field, type is URI

A pipe setup response shall be returned to requesting device within 30 s after the request is sent. The message format is as follows (see Table 16).

Table 16 – Pipe setup response based on trusted third party authentication, encrypted message transmission, and authentication mechanism

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code of pipe creation	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

If the security mechanism negotiation of both parties is successful, it shall continue to Phase 2 of pipe setup, implementing protocol interaction based on the security mechanism and encryption algorithm selected.

9.2.3.3 Secure identity authentication and pipeKey negotiation phase

9.2.3.3.1 Security mechanism between devices is identity authentication and message authentication mechanism based on symmetric-key cryptosystem

The interaction steps in this phase depend on the security mechanism negotiation result of IGRS Device 1 and IGRS Device 2 in Phase 1 (see 9.2.3.2).

Step 4: Device 1 generates a random number as Challenge 1 using a random number generator and encapsulates it as an authentication request to send to Device 2.

The authentication request message format of step 4 is as follows (see Table 17).

Table 17 – Authentication request based on identity authentication and message authentication mechanism of symmetric-key cryptosystem

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended command line
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field
01-IGRSMessageType:AuthenticateRequest	Required field
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Challenge value 1 in step4	Required field

Step 5: After Device 2 receives an authentication request, it shall extract Challenge 1 and calculate response 1: Hash (Challenge1||PreSharedKey), among which PreSharedKey is pre-share key for both parties. Meanwhile it generates a random number as Challenge 2, and encapsulates the message {Response 1, Challenge 2} as the authentication response of the first cycle to send to Device 1; the response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 5 is given in Table 18.

Table 18 – Authentication response based on identity authentication and message authentication mechanism of symmetric-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field
01-IGRSMessageType:AuthenticateResponse	Required field
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
Content-type:multipart/byteranges; boundary=RESPONSE1_AND_CHALLANGE2	Required field
Content-length: Message body length	Required field
	Shall be empty
--RESPONSE1_AND_CHALLANGE2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Response value 1 in step 5	Required field
--RESPONSE1_AND_CHALLANGE2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Challenge value 2 in step 5	Required field
--RESPONSE1_AND_CHALLANGE2--	Required field

Step 6: After Device 1 receives the authentication response of the first cycle, it shall extract Response 1 and Challenge 2, and calculate $Hash(Challenge1 || PreSharedKey)$ based on the Challenge 1 and PreSharedKey of Device 1. Then it shall compare the calculation result with Response 1, if they are equal, it will calculate $Hash(Challenge2 || PreSharedKey)$ as Response 2, and encapsulate Response 2 as the authentication result to send to Device 2. If they are not equal, it will continue to Phase 3, sending pipe setup Failed confirmation to Device 2.

The authentication result message format of step 6 is as follows (see Table 19).

Table 19 – Authentication result request based on identity authentication and message authentication mechanism of symmetric-key cryptosystem

Message	Field Explanation
M-POST /IGRS HTTP/1.1	extended command line
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field
01-IGRSMessageType:AuthenticateResultRequest	Required field
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Message body is response value 2 in step 6	Required field

Step 7: After Device 2 receives an authentication response result from Device 1, it shall extract Response 2 from it and calculate $Hash(Challenge2 \,||\, PreSharedKey)$ based on the Challenge 2 and PreSharedKey of Device 2. Then Device 2 shall compare the calculation result with Response 2. If they are equal, it will send authentication OK to Device 1. If they are not equal, it shall continue to Phase 3, sending pipe setup failed confirmation to Device 1. The response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 7 is as follows (see Table 20).

Table 20 – Authentication result response based on identity authentication and message authentication mechanism of symmetric-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field
01-IGRSMessageType:AuthenticateResultResponse	Required field
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

Because this security mechanism does not have message confidentiality processing, the PipeKey calculation is not required here; therefore, if the bi-directional identification authentication is successful, it shall continue to Phase 3, sending a pipe setup OK confirmation to both parties and controlling the security attribute synchronisation relationship.

9.2.3.3.2 Security mechanism between devices is identity authentication and encrypted message transmission, and authentication mechanism based on symmetric-key cryptosystem

The interaction steps in this phase depend on the security mechanism negotiation result of IGRS Device 1 and IGRS Device 2 in Phase 1 (see 9.2.3.2).

Step 4: Device 1 generates a random number as Challenge 1 using a random number generator and encapsulates it as an authentication request to send to Device 2. The message format is as follows (see Table 21).

Table 21 – Authentication request based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Message body is challenge value 1 in step 4	Required field

Step 5: After Device 2 receives an authentication request, it shall extract Challenge 1 and calculate response 1: Hash(Challenge 1 || PreSharedKey), among which PreSharedKey is the pre-share key for both parties. Meanwhile it generates a random number as Challenge 2, and encapsulates the message {Response 1, Challenge 2} as the authentication response of the first cycle to send to Device 1; the response message shall be returned to the requesting device within 30 s after the request is sent.

The message format of step 5 is (Table 22):

Table 22 – Authentication response based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem

Message	Field Explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see ClauseB.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
Content-type:multipart/byteranges; boundary=RESPONSE1_AND_CHALLANGE2	Required field
Content-length: Message body length	Required field
	Shall be empty
--RESPONSE1_AND_CHALLANGE2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Response value 1 in step 5	Required field
——RESPONSE1_AND_CHALLANGE2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Challenge value 2 in step 5	Required field
-- RESPONSE1_AND_CHALLANGE2--	Required field

Step 6: After Device 1 receives the authentication response of the first cycle, it shall extract Response 1 and Challenge 2, and calculate Hash (Challenge 1 || PreSharedKey) based on the Challenge 1 and PreSharedKey of Device 1. Then it shall compare the calculation result and Response 1; if they are equal, it will calculate Hash (Challenge 2 || PreSharedKey) as Response 2, and encapsulate Response 2 as authentication result to send to Device 2. If they are not equal, it shall continue to Phase 3, sending a pipe setup failed confirmation to Device 2.

The authentication result message format of step 6 is as follows (see Table 23).

Table 23 – Authentication result request based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResultRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID (unsignedInt)	Required field, type is unsignedInt (0 is reserved)
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Message body is response value 2 in step 6	Required field

Step 7: After Device 2 receives an authentication response result from Device 1, it shall extract Response 2 from it and calculate Hash (Challenge 2 || PreSharedKey) based on the Challenge 2 and PreSharedKey of Device 2. Then it shall compare the calculation result with Response 2, if they are equal, it shall send authentication OK to Device 1. If they are not equal, it shall continue to Phase 3, sending pipe setup failed confirmation to Device 1. The response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 7 is as follows (see Table 24).

Table 24 – Authentication result response based on identity authentication and encrypted message transmission and authentication mechanism of symmetric-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResultResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

Because the security mechanism URN:IGRS:Security:PreSharedKey_Cipher_MAC requires that message confidentiality processing be implemented, PipeKey calculation is needed here.

First, when Device 2 sends identification authentication OK, it shall calculate the PipeKey immediately according to the methods described in this security mechanism. After Device 1 receives a bi-directional identification authentication OK message from Device 2, it shall calculate PipeKey immediately the same way.

Phase 3 then continues with each device sending a pipe setup OK confirmation to each other, and controlling the security attribute synchronisation relationship.

9.2.3.3.3 Security mechanism between devices is authentication and encrypted message transmission and authentication mechanism based on public-key cryptosystem

The interaction steps in this phase depend on the security mechanism negotiation result of IGRS Device 1 and IGRS Device 2 in Phase 1 (see 9.2.3.2).

Step 4: Device 1 generates a random number as Challenge 1 using a random number generator and encapsulates it as an authentication request to send to Device 2. The message format is as follows (see Table 25).

Table 25 – Authentication request based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
Content-type:application/octet-stream	Required field
Content-length: Message body length	Required field
	Shall be empty
Message body is challenge value 1 in step 4	Required field

Step 5: After Device 2 receives an authentication request, it shall extract Challenge 1 and the public key certificate of Device 1, and implement a digital signature to Challenge 1 with the private key of Device 2, assigning the signature value as Response 1. The signature algorithm is the public key algorithm for authentication. Meanwhile Device 2 generates a random number as Challenge 2 and another Random_bit-string. It encrypts for Random_bit_string with the public key of Device 1, representing the result as Cipher_Random_bit_string. It encapsulates the message {Response 1, Challenge 2, Cipher_Random_bit_string} as a confirmation response of the first cycle to send to Device 1. The response message shall be returned to the requesting device within 30 s after the request is sent. The message format is as follows (see Table 26).

Table 26 – Authentication response based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP Command Line
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
Content-type:multipart/byteranges; boundary=RESPONSE1_CHALLANGE2_CIPHERRANDOMSTRING	Required field
Content-length: Message body length	Required field
	Shall be empty
--RESPONSE1_CHALLANGE2_CIPHERRANDOMSTRING	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Response value 1 in step 5	Required field
--RESPONSE1_CHALLANGE2_CIPHERRANDOMSTRING	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Challenge value 2 in step 5	Required field
--RESPONSE1_CHALLANGE2_CIPHERRANDOMSTRING	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Cipher_Random_bit_string in step 5	Required field
--RESPONSE1_CHALLANGE2_CIPHERRANDOMSTRING--	Required field

Step 6: After Device 1 receives the authentication response of the first cycle, it shall extract Response 1, Challenge 2, and Cipher_Random_bit_string, and then implement decryption to Cipher_Random_bit_string with the private key of Device 1. The decryption result is recorded as Random_bit_string1. Device 1 implements signature authentication with the public key, Challenge 1 and Response 1 of Device 2; if it passes the authentication, Device 1 shall implement a digital signature to Challenge 2 with the private key of Device 1 and regard the signature result as Response 2; meanwhile Random_bit_string1 is used to encrypt for Challenge2 with symmetric encryption algorithm, recording the encryption result as Cipher_Challenge2, and then it encapsulates Response 2 and Cipher_Challenge2 as authentication result to send to Device 2. If the signature is invalid, it shall continue to Phase 3, sending pipe setup failed confirmation to Device 2.

The authentication result message format of step 6 is as follows (see Table 27).

Table 27 – Authentication result request based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResultRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
Content-type:multipart/byteranges; boundary=RESPONSE2_CIPHERCHALLANGE2	Required field
Content-length: Message body length	Required field
	Shall be empty
--RESPONSE2_CIPHERCHALLANGE2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Response value 2 in step 5	Required field
--RESPONSE2_CIPHERCHALLANGE2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Cipher_Challenge2 in step 5	Required field
--RESPONSE2_CIPHERCHALLANGE2--	Required field

Step 7: After Device 2 receives authentication response result from Device 1, it shall extract Response 2 and Cipher_Challenge2 from it and decrypt for Cipher_Challenge2 with symmetric encryption algorithm, using Random_bit_string as the key; meanwhile it implements signature validity check with the public key of Device 2, Challenge 2 and Response 2; if the decryption result is Challenge 2 and signature passes validity check, it shall send bi-directional identification authentication OK and both party's correctly sharing Random_bit_string message to device 1. If they are not equal, it shall continue to Phase 3, sending pipe setup failed confirmation to device 1. The response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 7 is as follows (see Table 28).

Table 28 – Authentication result response based on authentication and encrypted message transmission and authentication mechanism of public-key cryptosystem

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResultResponse	Required field, content shall be x with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

After that, Device 1 shall define Random_bit_string1 as PipeKey; Device 2 shall define Random_bit_string as PipeKey and then continue to Phase 3, sending a pipe setup OK confirmation to both parties to control the security attribute synchronisation relationship.

9.2.3.3.4 Security mechanism between devices is authentication, encrypted message transmission and authentication mechanism based on trusted third party

Here the symmetric encryption system implements the Kerberos protocol, and assumes that both Device 1 and Device 2 share their respective pre-shared keys with a trusted third party that provides authentication service.

The interaction steps in this phase depend on the security mechanism negotiation result of IGRS Device 1 and IGRS Device 2 in Phase 1 (see 9.2.3.2).

Step 4: According to the detailed description of security mechanism negotiated with Device 2, for exammple AuthenServiceProvider, EncryptAlgorithm and MACAlgorithm, Device 1 shall send a Ticket generation service request to AuthenServiceProvider; messages sent include: {Device 1 ID, Device 2 ID, description ID of security mechanism of trusted third party that is supported by both Device 1 and 2, nonce. The token generated by Device 1 is based on the security relationship with AuthenServiceProvider}. The message format is as follows (see Table 29).

Table 29 – Authentication request based on authentication, encrypted message transmission, and authentication mechanism of trusted third party

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-TargetServiceId: Target Service ID	Required field, type is unsignedInt (0 is reserved)
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-Device1Id: Device1 ID	Required field, type is URI, defined in 8.1.2
01-Device2Id: Device2 ID	Required field, type is URI, defined in 8.1.2
01-DeviceSecurityId: Device security mechanism supported by both Device 1 and 2 description ID	Required field, type is URI, defined in 8.1.5.5
	Shall be empty
Content-type:multipart/byteranges; boundary=NONCE_TOKEN	Required field
	Shall be empty
--NONCE_TOKEN	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Nonce in step 4	Required field
--NONCE_TOKEN	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Token in step 4	Required field
--NONCE_TOKEN--	Required field

Step 5: After the AuthenServiceProvider receives Ticket generation request from device 1, it shall implement a validity check to Token of Device 1; if it is valid, it shall generate a Ticket based on messages {Device 1 ID, Device 2 ID, description ID of security mechanism of trusted third party that is supported by both Device 1 and 2}. AuthenServiceProvider generates PipeKey and MACKey, and then it encrypts the message {Device 1 ID, description ID of security mechanism of a trusted third party that is supported by both Device 1 and 2, PipeKey, MACKey, valid Time} as Cipher1 with the pre-share key with Device 1; and encrypts the message {Device 2 ID, Device 1 ID, description ID of security mechanism of trusted third party that is supported by both Device 1 and 2, PipeKey, MACKey, valid Time} as Cipher2 using the pre-shared key with Device 2, and encapsulates Cipher 1 and Cipher 2 as Ticket generation response to send to Device 1; the response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 5 is as follows (see Table 30).

Table 30 – Authentication response based on authentication, encrypted message transmission, and authentication mechanism of trusted third party

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
Content-type:multipart/byteranges; boundary=CIPHER1_CIPHER2	Required field
Content-length: Message body length	Required field
	Shall be empty
--CIPHER1_CIPHER2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Cipher1 in step 5	Required field
--CIPHER1_CIPHER2	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Cipher2 in step 5	Required field
--CIPHER1_CIPHER2--	Required field

Step 6: After Device 1 receives a Ticket generation response from AuthenServiceProvider, it shall receive message Cipher 1 and Cipher 2 and decrypt for Cipher 1 to extract message {Device 1 ID, description ID of the security mechanism of the trusted third party that is supported by both Device 1 and 2, PipeKey, MACKey, valid Time}, and then encrypt {Device 1 ID, description ID of security mechanism of the trusted third party that is supported by both Device 1 and 2, PipeKey, MACKey, valid Time} as Cipher3 with PipeKey, meanwhile encapsulating Cipher 2 and Cipher 3 as an authentication request to send to Device 2.

The message format of step 6 is as follows (see Table 31).

Table 31 – Authentication result request based on authentication, encrypted message transmission, and authentication mechanism of trusted third party

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResultRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt
01-DeviceSecurityId: Device security mechanism supported by both Device 1 and 2 description ID	Required field, type is URI, defined in 8.1.5.5
Content-type:multipart/byteranges; boundary=CIPHER2_CIPHER3	Required field
Content-length: Message body length	Required field
	Shall be empty
--CIPHER2_CIPHER3	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Cipher2 in step 5	Required field
--CIPHER2_CIPHER3	Required field
Content-type:application/octet-stream	Required field
	Shall be empty
Cipher3 in step 6	Required field
--CIPHER2_CIPHER3--	Required field

Step 7: After Device 2 receives the authentication request from Device 1, it shall receive the message {Cipher2, Cipher3}; first it decrypts Cipher 2 and extracts message {Device 2 ID, Device 1 ID, description ID of security mechanism of the trusted third party that is supported by both Device 1 and 2, PipeKey, MACKey, valid Time}, and then decrypts Cipher 3 with PipeKey to extract message {Device 1 ID, description ID of security mechanism of the trusted third party that is supported by both Device 1 and 2, PipeKey, MACKey, valid Time}; comparing the two clear text messages, if the content in related fields is identical, the authentication is successful, and it shall set up security context of this pipe based on EncryptAlgorithm, MACAlgorithm, PipeKey, and MACKey provided by trusted third party security mechanism and send an authentication OK message to Device 1. Otherwise it shall send the authentication Failed message. The response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 7 is as follows (see Table 32).

Table 32 – Authentication result response based on authentication, encrypted message transmission, and authentication mechanism of trusted third party

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:AuthenticateResultResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

9.2.3.4 Secure device pipe setup confirmation phase

Step 8: If the pipe setup is successful for Device 1, Device 1 shall send a pipe setup OK confirmation to Device 2, otherwise it shall send a pipe setup Failed confirmation.

The message format of step 8 is as follows:

Table 33 – Secure device pipe setup confirmation request

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended command line
Host: Target Host IP:port	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeResultRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-CreatePipeResult:OK/FAILED	Required field, type is string; result of pipe creation, if success, OK; if not, FAILED
01-TargetDeviceId: Target Device ID	required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)

Step 9: When Device 2 receives a pipe setup OK confirmation from Device 1, and it has successfully implemented the related protocol steps from the previous period, it shall return a pipe setup OK confirmation to Device 1, otherwise it shall return pipe setup Failed confirmation. The response message shall be returned to requesting device within 30 s after the request is sent.

The message format of step 9 is as follows (see Table 34).

Table 34 – Secure device pipe setup confirmation response

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreatePipeResultResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, refer to Clause 11
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

This phase primarily shows pipe setup OK and Failed. All the protocol steps in Phase 1 and Phase 2 shall be able to jump directly to any step in this phase.

This phase accomplishes a key objective through two steps: after the pipe is successfully set up, the security mechanism, encryption algorithm, and security context selected after the negotiations between both parties shall be in synchronisation.

9.2.4 Secure device pipe teardown

After a pipe between devices is successfully set up, one of the following reasons should result in pipe teardown:

one device is offline

Both sides of a pipe should start an online detection timer after the pipe is set up. If either device has not received the corresponding online advertisement message before the online advertisement period ends, it should send an online detection request message to the other side of pipe (see 9.2.6). If no response is received in 30 s after the request is sent, the device shall be considered offline, and the pipe is cut.

pipe times out

When a pipe between devices meets one of the two conditions below and time is more than 30 s

- a session does not exist. For more related content about session, please refer to 10.5,
- no request based on a session waiting for a response (request in the process of session setup, device/service search request, event subscription request, detailed information of device/service retrieval request).

initiate teardown

After a pipe is setup, the device on either side shall be able to send a pipe teardown notice message to the device on the other side of the pipe when that device is out of network, powered off, or to perform other administration functions. The device receiving this message shall cut off the related pipe immediately.

The secure Device Pipe teardown notice message is defined as follows (see Table 35).

Table 35 – Secure device pipe teardown notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	Extended command line
Host: Target Host IP:port	Required field
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:DetachPipeNotify	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2

9.2.5 Device trust relationship

After a pipe setup is implemented, the devices on both sides of the pipe shall be able to form a specified trust relationship. The IGRS specification defines two types of trust relationships between devices:

- trusted;
- untrusted.

A trust relationship formed between devices after the pipe setup process is shown in Table 36.

Table 36 – Trust relationship formed between devices after pipe setup

Pipe creation mechanism	Result of creation	Device trusted/untrusted relationship
NULL	Success	Untrusted
	Fail	None
PreSharedKey_MAC	Success	Trusted
	Fail	Untrusted
PreSharedKey_Cipher_MAC	Success	Trusted
	Fail	Untrusted
PKICertificate_Cipher_MAC	Success	Trusted
	Fail	Untrusted
3rdPartyAuthenService	Success	Trusted
	Fail	Untrusted

9.2.6 Device online detection

After two IGRS Devices set up a security Device Pipe, either the IGRS Device shall be able to generate a device online detection message to the other IGRS Device. The device receiving the device online detection message shall send a device online detection response message to the initiator.

Device online detection should be used to confirm whether the partner is online between two devices. Such detection is based on a request/response message model. After a secure pipe is set up between devices, the two devices may start a periodic online detection request. The device that receives the request message shall return the corresponding device online

response message. If one device has not received a response message after sending an online detection request message in the specified period, it shall consider the partner to be offline. The maximum waiting time shall be 30 s.

The device online detection message format is defined in Table 37.

Table 37 – Device online detection request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended command line
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:DeviceOnlineDetectionRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)

The response message format of device online detection is defined in Table 38.

Table 38 – Device online detection response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:DeviceOnlineDetectionResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId:Response to detection Device ID	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId:Request detection Device ID	Required field, type is URI, defined in 8.1.2
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message

9.3 Detailed device description document retrieval

9.3.1 Retrieve detailed device description document request

IGRS clients may retrieve the related description information of a device by sending a detailed device description document retrieval request message. The specific message format is defined in Table 39.

Table 39 – Device description document retrieval request message

Message	Field explanation
M-GET /IGRS HTTP/1.1	HTTP COMMAND LINE
HOST: Target Host IP Address:port	Required field
ACCEPT-LANGUAGE:expected description language	Suggested field, and its description conforms to RFC 1766.
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version
01-IGRSMessageType:GetDeviceDescriptionRequest	Required field
01-SourceDeviceId: source device IDthat is sending the request	Required field and type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device ID	Required field and type is URI. For definition, see 8.1.2
01-SequenceId: Device Pipe message sequence ID	Required field, and the type is 32-bit unsigned Int (0 is reserved)
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-GetDeviceDescription-Request"	Required field
Content-Type: text/xml; charset=utf-8	Required field
Content-Length: Message body length	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/ envelope/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/ Soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<ClientId> Client ID who is sending the request </ClientId>	Required field, and the type is 32-bit unsigned Int (0 is reserved)
<SequenceId> Retrieval message sequence ID </SequenceId>	Required field, and the type is 32-bit unsigned Int (0 is reserved)
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.3.2 Retrieve detailed device description document response

The device that receives the device description document request message shall return a device description document response. The format of the response message is defined in Table 40 below. The response message shall be returned to the requesting device within 30 s after the request is sent.

Table 40 – Device description document retrieval response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
CONTENT-LENGTH: Message body length	Required field
CONTENT-TYPE:text/xml; Charset=utf-8	Required field
CONTENT-LANGUAGE:Document description language	If and only if request message includes ACCEPT-LANGUAGE field, this field is a required field, language description rule refer to RFC 1766
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field
01-IGRSMessageType:GetDeviceDescriptionResponse	Required field; content should be as specified here.
01-AcknowledgeId: Device Pipe message acknowledgement ID	Required field, and the type is 32-bit unsignedInt (0 is reserved). It is same to “SequenceID” field in the requesting message.
01-SourceDeviceId:source device ID that is sending response.	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device ID	Required field, and type is URI. For definition, see 8.1.2
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-GetDeviceDescription-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop e/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap /encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=“http://www.igrs.org/spec1.0”>	Required field, see Clause B.1
<ClientId> The client ID who is sending response. </ClientId>	Required field and the type is 32-bit unsignedInt (0 is reserved).
<ReturnCode> Response return code </ReturnCode>	Required field, and for definition, see Clause 11
<AcknowledgeId> Response acknowledgement ID </AcknowledgeId>	Required field and the type is 32-bit unsignedInt (0 is reserved). It is same to “SequenceID” field in the requesting message
<DeviceDescription> service description document </DeviceDescription>	The service description document constructed based on device description template
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.4 Retrieve detailed device description document based on non-secure pipe

IGRS Devices may support a user-defined message mechanism to retrieve a detailed device description document through a non-secure pipe. As shown in Table 7, IGRS Devices may get the device description document by using the URL of the description document indicated in the “location” field.

9.5 Device group setup

An IGRS Device group is divided into three sub-types: global peer-to-peer device group, specified peer-to-peer device group, and centralised device group. IGRS Devices may setup and join a peer-to-peer device group or a centralised device group according to the application requirement. One device may join multiple device groups simultaneously.

9.5.1 Global peer-to-peer device group

Before an IGRS Device joins any device groups, the default shall be a global peer-to-peer device group. The global peer-to-peer device group does not need a device group advertisement and does not have a device group ID.

9.5.2 Specified peer-to-peer device group

9.5.2.1 Specified peer-to-peer device group setup

Any device may set up a specified peer-to-peer device group. The process of setting up a peer-to-peer device group is as follows.

a) Place the identifier and name of this device group into the description of this device.

b) Advertise the identifier of this device group in device advertisement.

c) Periodically advertise this device Group as explained in 9.5.2.2.

Each device shall administer access security among devices in a specified peer-to-peer device group.

9.5.2.2 Specified peer-to-peer device group advertisement

In specifying a peer-to-peer device group, by default, the device setting up this device group shall be responsible for this device group advertisement. If this device is offline, then one device in the device group shall be selected to take charge of the group advertisement in one device group advertisement interval. All group advertisements are implemented based on multicast.

The device group advertisement message format of a specified peer-to-peer device group is defined in Table 41.

Table 41 – Device group advertisement message of specified peer-to-peer device group

Message	Field explanation
NOTIFY * HTTP/1.1	Extended HTTP COMMAND LINE
Host:239.255.255.250:3880	Required field
Cache-control:max-age=Max advertisement valid time	Required field, type is 32 bit unsignedInt (0 is reserved) when device receives this message and the time has expired, then the device group no longer exists; before this max–age is reached, device-in-charge shall send new device group advertisement to reset this time, unit is in second
Location: http://www.igrs.org/devicegroup/peer	Required field, see Clause B.8
NT: uuid: Advertising Device Group ID	Required field, see 8.1.3
NTS:isdp:groupalive	Required field, ISDP required
SERVER: OS/version IGRS/1.0 product/version	Required field
USN: uuid: Advertising Device Group ID	Required field, see 8.1.3
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:PeerDeviceGroupAdvertisement	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: Source Device ID that is sending group advertisement	Required field, type is URI, defined in 8.1.2
Content-Type:text/xml; charset=utf-8	Required field
Content-Length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-PeerDeviceGroup-Advertisement"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<DeviceGroupId> Device Group ID </DeviceGroupId>	Required field, see 8.1.3
<DeviceGroupName> Device Group Name </DeviceGroupName>	Required field, type is string
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.5.2.3 Join specified peer-to-peer device group

After another device receives the advertisement of this device group, it should decide whether to join this peer-to-peer device group. The process of joining a specified peer-to-peer device group is the same as Step a) in 9.5.2.1 of the specified peer-to-peer device group setup described above. No identity authentication shall be required.

9.5.2.4 Leave specified peer-to-peer device group

A device in a specified peer-to-peer device group may leave the specified peer-to-peer device group. The device shall remove the device group identifier and device group name from the device description and shall no longer be responsible and/or ready to implement the advertisement of this device group. When a device leaves a specified peer-to-peer device group normally, it shall send a quit group message based on multicast. This message does not require a response. The format is defined in Table 42.

Table 42 – Device leaves a specified peer-to-peer device group quit group message

Message	Field Explanation
NOTIFY * HTTP/1.1	Extended HTTP COMMAND LINE
Host:239.255.255.250:3880	Required field
NT: uuid: Advertising Device Group ID	Required field, see 8.1.3
NTS:isdp:quitgroup	Required field
USN:uuid:Advertising Device Group ID	Required field, see 8.1.3
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: QuitPeerDeviceGroupNotify	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-DeviceGroupId: Device Group ID of device group to be dismissed	Required field, type is URI, defined in 8.1.3
01-SourceDeviceId:Device ID of device group to be dismissed	Required field, type is URI, defined in 8.1.2

9.5.3 Centralised device group

9.5.3.1 Centralised device group setup

Any device that is designated a master device may set up a centralised device group. This device serving as the master device of a centralised device group shall be responsible for the device group advertisement in the network. Other devices may decide whether to join this centralised device group after receiving the advertisement message.

9.5.3.2 Centralised device group advertisement

A device group advertisement of centralised device group shall only be implemented by the master device of the device group. The message format is defined in Table 43.

Table 43 – Device group advertisement message of master-slave device group

Message	Field explanation
NOTIFY * HTTP/1.1	Extended HTTP COMMAND LINE
Host:239.255.255.250:3880	Required field
Cache-control:max-age=Max advertisement valid time	Required field, type is 32 bit unsignedInt (0 is reserved) when device receives this message and the time has expired, then the device group no longer exists; before this max–age is reached, master device shall send new device group advertisement to reset this time, unit is in second
Location: http://www.igrs.org/devicegroup/centralised	Required field, see Clause B.6
NT:uuid:Advertising Device Group ID	Required field, see 8.1.3
NTS:isdp:groupalive	Required field, ISDP requirement
SERVER: OS/version IGRS/1.0 product/version	Required field
USN:uuid:Advertising Device Group ID	Required field, see 8.1.3
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CentralisedDeviceGroupAdvertisement	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: Source Device ID that is sending group advertisement	Required field, type is URI, defined in 8.1.2
Content-Type:text/xml; charset=utf-8	Required field
Content-Length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-CentralisedDeviceGroup-Advertisement"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop e/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap /encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<DeviceGroupId> Device Group ID </DeviceGroupId>	Required field, see 8.1.3
<DeviceGroupName> Device Group Name </DeviceGroupName>	Required field, type is string
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.5.3.3 Join centralised device group

When a device wants to join a centralised device group, this device shall generate a join request to the master device in a centralised device group through a secure pipe or unsecure pipe. After the device passes identification authentication with the master device in this centralised device group, the master device shall return a response with the joining result. Access security between slave devices in a centralised device group shall be administered

commonly by the master device. If security authentication interaction is between two devices, the master device shall be considered as a trusted third party. As an authentication server, the master device shall be responsible for identification authentication among slave devices.

The format of request message to join a master-slave device group is defined in Table 44.

Table 44 – Request message to join a master-slave device group

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extend HTTP COMMAND LINE
Host: Target Host IP Address:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: JoinCentralisedDeviceGroupRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-DeviceGroupId:Device group ID of the device group to join	Required field, type is URI, defined in 8.1.3
01-SourceDeviceId: Device ID of request device	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId: target master device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Request Message Sequence ID	Required field, type is 32 bit unsignedInt (0 is reserved)

After the master device receives a join request, it shall decide whether to implement identification authentication with the request device and give the corresponding response based on the requirement of this device. The response message shall be returned to the requesting device within 30 s after the request is sent.

The response message format is defined in Table 45.

Table 45 – Response message to join a master-slave device group

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1,0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:JoinDeviceGroupResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: Source Master Device ID	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId: Device ID of device that sent join request	Required field, type is URI, defined in 8.1.2
01-ReturnCode: Response status code	Required field, see Clause 11
01-DeviceGroupId: Device group ID of device group to join	Required field, type is URI, defined in 8.1.3
01-AcknowledgeId: Response Message sequence ID	Required field, type is unsignedInt (0 is reserved), same as the sequence ID in request message

9.5.3.4 Centralised device group dissolve

Two cases shall cause a device in a centralised device group to withdraw from the group as follows.

a) The master device of this device group is offline. After the slave device receives the last advertisement from this device group and has not received a new advertisement message within the valid time specified in the previous advertisement, it shall regard the master device as offline. If a pipe exists between the slave device and the master device, the slave device should send a device online detection request to the master device through the device online detection mechanism. It shall confirm whether the master device is offline by the response status.

b) The master device of this device group sends a dissolve notice. The device group dissolve notification message sent by the master device is a multicast message. The message format is defined in Table 46.

Table 46 – Device group dissolve notification message sent by master device

Message	Field explanation
NOTIFY * HTTP/1.1	Extended HTTP COMMAND LINE
Host:239.255.255.250:3880	Required field
NT:uuid:Advertising Device ID	Required field, see 8.1.2
NTS:isdp:groupbyebye	Required field
USN:uuid:Advertising Device ID	Required field, see 8.1.2
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field
01-IGRSMessageType: QuitCentralisedDeviceGroupAdvertisement	Required field
01-DeviceGroupId:Device group ID of device group to quit	Required field, type is URI, defined in 8.1.3
01-SourceDeviceId: master Device ID	Required field, type is URI, defined in 8.1.2

A slave device should send the notice to withdraw from the centralised group to the master device. When the slave device leaves the master-slave device group normally, it shall send a withdraw notification message (does not require a response) to the master device. The message format is defined in Table 47.

Table 47 – Withdraw notification message sent by slave device

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	Extended HTTP COMMAND LINE
Host: Target Host IP Address:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field
01-IGRSMessageType: QuitCentralisedDeviceGroupNotify	Required field
01-TargetDeviceId:Master Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId: Device ID of device quitted from device group	Required field, type is URI, defined in 8.1.2
01-DeviceGroupId: device group ID of device group request to quit	Required field, type is URI, defined in 8.1.3

9.6 Device search

9.6.1 Device search based on multicast

An IGRS Device should receive a network device messages existing on the network by listening to device online advertisement messages. Also, it should obtain device information by sending a device search message and waiting for a device search response message to return.

IGRS Devices adopt the ISDP protocol mechanism.

An IGRS Device may send device search messages, as specified in Clause A.2 to the multicast address 239.255.255.250:3880, and search the device in compliance with the search conditions in the entire network.

An IGRS Device search message format based on multicast is shown in Table 48.

Table 48 – Device search request message

Message	Field explanation
M-SEARCH * HTTP/1.1	Extended HTTP command line by ISDP
Host:239.255.255.250:3880	Required field
MAN:”isdp:discover”	Required field
MX:max time to wait response	Required field, in seconds. Response should be randomly delayed with a range of 0 s to MX s. Maximum MX value is 120 s.
ST:urn:schemas-IGRS-org:device:IGRS-device:1 (Search type)	Required field, content shall be urn:schemas-IGRS-org:device:IGRS-device:1
Man:”http://www.igrs.org/spec1.0”; ns= 01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: SearchDeviceRequest	Required field
01-SourceDeviceId: Source Device ID	Required field, type is URI, defined in 8.1.2
01-SequenceId: Sequence ID of search request message	Required field, type is unsignedInt (0 is reserved)
01-clientId: ID of client that initiated search request	Required field, type is unsignedInt (0 is reserved)
01-SearchAll:TRUE	Optional field, type is string ,search all devices
01-SearchByDeviceName: Device name	Optional field, type is string ,search by device name
01-SearchByDeviceType: Device type	Optional field, type is URI, defined in 8.1.4 ,search by device type
01-SearchByDeviceId: Device ID	Required field, type is URI, defined in 8.1.2 ,search by device ID
01-SearchByDeviceGroupId: Device group ID	Optional field, type is URI, defined in 8.1.3 ,search all devices of this device group
01-SearchByServiceType: Service type	Optional field, type is URI, defined in 8.2.3 ,search device includes this service type
01-SearchByServiceName: Service name	Optional field, type is string,search device includes a specific service name
At least one of the fields among SearchAll, SearchByDeviceName, SearchByDeviceGroupId, SearchByDeviceName, SearchByDeviceType, SearchByServiceType, SearchByServiceName and SearchByDeviceId shall be valid. When SearchAll is true, the other fields shall be ignored; if SearchAll is false, the search result should match with all fields of search precondition.	

An IGRS Device that receives a device search request shall detect the search condition. If the search condition is met, the device shall send a device search response message to the IGRS

Device that generated the device search. The search response message of this device shall be sent unicast through UDP (target port number: 3880).

The device search response message format is defined in Table 49.

Table 49 – Device search response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Cache-control:max-age=Max time to receive response Message;no-cache="Ext"	Required field, max-age type is unsignedInt (0 is reserved), in seconds. The field function is same to max-age field in device online advertisement message and it indicates the possible device online duration.
Ext:	Required field
Server:OS/version IGRS/1.0 product/version	Required field
Location: url of device description document	Required field, support unsecure pipes among IGRS Devices to get device description documents. Its value is the URL of the documents. When device does not support device description documents access through non-secure pipes, then this field shall be: http://www.igrs.org/device, see Clause B.7
ST:urn:schemas-IGRS-org:device:IGRS-device:1	Required field, content shall be x, with x being exactly the content of the coresponding left hand column
USN:uuid:Response Device ID::Device Type ID	Required field, device ID defined in 8.1.2, device type ID defined in 8.1.4
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SearchDeviceResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId:Response Device ID	Required field, for definition, refer to 8.1.2
01-TargetDeviceId: device ID that sends request	Required field, for definition, refer to 8.1.2
01-SecureListenerList: IP & port list of secure Device Pipes that are initiating device searching listening	Optional field and the type is string. The type is "IP: Port" and multiple IP addresses are separated with ";". Secure Device Pipe listener port is fixed at 3880
01-ListenerList: IP & port list of non-secure Device Pipesthat are initiating device searching listening.	Required field and the type is string. The type is "IP: Port" and multiple IP addresses are separated with ";". One port number shall be 3880
01-DeviceSecurityIdList:Security mechanism list of the device	Required field, type is device security ID list, see 8.1.5, separated with ";"
Content-type:text/xml; charset=utf-8	Required field
Content-length:Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-SearchDevice-Response"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<ReturnCode> Response status code </ReturnCode>	Required field, see Clause 11 for definition

Message	Field explanation
<AcknowledgeId> Searching response device sequence ID </AcknowledgeId>	Required field, the type is 32-bit unsignedInt (0 is reserved). It is same to SequenceId of searching request message
<TargetClientId> Target client ID </TargetClientId>	Required field, type is unsignedInt (0 is reserved)
<SearchResult>	Required field
<DeviceInfoList>	Required field
<DeviceInfo>	Required field
<DeviceId> Device ID </DeviceId>	Required field, type is URI, defined in 8.1.2, device ID meets search criteria
<DeviceGroupIdList> <DeviceGroupId> Device group ID </DeviceGroupId> <!-- other Device group ID --!> </DeviceGroupIdList>	Required field, DeviceGroupIdand type is URI. For definition, see 8.1.3, device group ID list meets search criteria
<DeviceName> Device name </DeviceName>	Required field, type is string, Device name meets search criteria
<DeviceSecurityIdList> <DeviceSecurityId> Device Security ID </DeviceSecurityId> <!--other device security ID--!> </DeviceSecurityId>	Required field, DeviceSecurityId and type is URI. For definition, see 8.1.5, device security mechanism list meets search criteria
<DeviceType> Device type </DeviceType>	Required field, type is URI, defined in 8.1.4
<DeviceAdvertisementTerm> Device Advertisement Period </DeviceAdvertisementTerm>	Optional field, in seconds, device advertisement period that meets search criteria, type is unsignedInt
<ConfigId> Device configuration change ID </ConfigId>	Required field, and type is 32-bit unsignedInt (0 is reserved). The configuration changes, this field indicates the change. ConfigId shall be increased by one (1) whenever there is a configuration change. This field will return to 1 when upper limit is reached
<BootId> Device Reboot ID </BootId>	Required field, and type is 32-bit UnsignedInt (0 is reserved). When the device reboots, this field indicates this change. BootID value shall be increased by one (1) whenever there is a reboot. This field will return to 1 when upper limit is reached
<SecureListenerList> <SecureListener> IP address and port list for secure Device Pipe </SecureListener> <!--other SecureListener--> </SecureListenerList>	Optional field and the type is string. The mode is “IP:Port”. Mutliple IP addresses can be separated by “;”. Secure device port number is 3880

Message	Field explanation
<ListenerList> <Listener> IP address and port list for non-secure Device Pipe </Listener> <!--other Listener--> </ListenerList>	Optional field and the type is string. The mode is “IP:Port”. Mutliple IP addresses can be separated by “;”. One Listener port Number shall be 3880
</DeviceInfo>	Required field
<!--other device info-->	
</DeviceInfoList>	Required field
</SearchResult>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.6.2 Device search by proxy

An IGRS Device may send out a device search request through a Device Pipe set up with another device. The message format of device search request is defined in Table 50.

Table 50 – Device search request that slave device generates to master device

Message	Field explanation
M-Post /IGRS HTTP/1.1	Extended command lin
HOST: Target device IP: port	Required field
MAN:”http://www.igrs.org/spec1.0”; ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SearchDeviceRequestOnDevice	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-SourceDeviceId:Source Device ID	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId:Target Device ID	Required field, type is URI, defined in 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-SearchDevice-Request-OnDevice”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field

<SourceClientId> Source Client ID </SourceClientId>	Required field, type is unsignedInt (0 is reserved)
Message	**Field explanation**
<SearchAll> TRUE </SearchAll>	Required field, type is boolean
<SearchByDeviceName> Device name </SearchByDeviceName>	Optional field, type is string
<SearchByServiceType> Service type </SearchByServiceType>	Optional field, type is URI, defined in 8.2.3
<SearchByServiceName> Service name </SearchByServiceName>	Optional field, type is string
<SearchByDeviceGroupId> Device group ID </SearchByDeviceGroupId>	Optional field, type is URI, defined in 8.1.3
<SearchByDeviceId> Device ID </SearchByDeviceId>	Optional field, type is URI, defined in 8.1.2
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field
At least one of the fields among SearchAll, SearchByDeviceName, SearchByDeviceGroupId, SearchByDeviceName, SearchByDeviceType, SearchByServiceType, SearchByServiceName and SearchByDeviceId shall be valid. When SearchAll is true, the other fields shall be ignored. If SearchAll is false, the search result should match with all fields of search precondition.	

After receiving a search request, the device shall decide whether to send a response based on its status. If a master device receives a request from a slave device, it shall generate a response regardless of whether a match is found, or, if the two devices have no master-slave relationship, the requested device may generate a response by searching all device messages it knows, or it may ignore the request message. The response message format is defined in Table 51.

Table 51 – Device search response message

Message	Field explanation
HTTP/1.1 200 OK	Http Command Line
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: SearchDeviceResponseOnDevice	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-TargetDeviceId: Target Device ID	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId:Source Device ID	Required field, type is URI, defined in 8.1.2
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
Ext:	Required field
Cache-control:max-age=possible online time of the device;no-cache="Ext"	Required field, max-age type is unsignedInt (0 is reserved), in seconds. The field function is same to max-age field in device online advertisement message and it indicates the possible device online duration
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-SearchDevice-Response-OnDevice"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<ReturnCode> Response status code </ReturnCode>	Required field, refer to Clause 11
<TargetClientId> Target client ID </TargetClientId>	Required field, type is unsignedInt (0 is reserved)
<SearchResult>	Required field
<DeviceInfoList>	Required field
<DeviceInfo>	Required field
<DeviceId> Device Id </DeviceId>	Required field, type is URI, defined in 8.1.2
<DeviceGroupIdList> <DeviceGroupId> Device group ID </DeviceGroupId> <!--other Device group ID--> </DeviceGroupIdList>	Required field, device group ID type is URI, defined in 8.1.3, device group ID list
<DeviceName> Device name</DeviceName>	Required field, type is string
<DeviceAdvertisementTerm> Device Advertisement Period </DeviceAdvertisementTerm>	Optional field, in seconds, device advertisement period that meets search criteria, type is unsignedInt

Message	Field explanation
<DeviceSecurityIdList> <DeviceSecurityId> Device Security mechanism </DeviceSecurityId> <!--other device security mechanism description--> </DeviceSecurityIdList>	Required field, type is URI, defined in 8.1.5
<DeviceType> Device type </DeviceType>	Required field
<ConfigId> Device configuration change ID </ConfigId>	Required field, and type is 32 bit unsignedInt (0 is reserved). The configuration changes, this field indicates the change. ConfigId shall be increased by one (1) whenever there is a configuration change. This field will return to 1 when upper limit is reached
<BootId> Device Reboot ID </BootId>	Required field, and type is 32 bit unsignedInt (0 is reserved). When the device reboots, this field indicates this change. BootId value shall be increased by one (1) whenever there is a reboot. This field will return to 1 when upper limit is reached
<SecureListenerList> <SecureListener> IP address and port list for secure Device Pipe </SecureListener> <!--other SecureListener--> </SecureListenerList>	Optional field and the type is string. The mode is “IP:Port”. Mutliple IP addresses can be separated by “;”. Secure device port number is 3880
<ListenerList> <Listener> IP address and port list for non-secure Device Pipe </Listener> <!--other Listener--> </ListenerList>	Optional field and the type is string. The mode is “IP:Port”. Mutliple IP addresses can be separated by “;”. One Listener port number shall be 3880
</DeviceInfo>	Required field
<!--other device information-->	
</DeviceInfoList>	Required field
</SearchResult>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
<SOAP-ENV:Envelope>	Required field

9.7 Device online/offline event subscription

9.7.1 Device online/offline event subscription request

An IGRS Device may a send device online/offline subscription request to another IGRS Device via an IGRS pipe to obtain a specified device online/offline event.

The device event subscription request message format is as follows (see Table 52).

Table 52 – Device event subscription request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SubscribeDeviceEventRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-SourceDeviceId:Source Device ID that request subscription	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId:Target Device ID that receives subscription request	Required field, type is URI, defined in 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-SubscribeDeviceEvent-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> Source Client ID </SourceClientId>	Required field, type is unsignedInt (0 is reserved)
<SubscriptionId> subscription ID </SubscriptionId>	Required field, type is unsignedInt (0 is reserved)
<ExpectedSubscriptionTerm> Expected valid subscription time </ExpectedSubscriptionTerm>	Required field, in seconds, type is unsignedInt (0 is reserved)
<SubscriptionFilter>	Required field
<SubscribeAll>TRUE </SubscribeAll>	Optional field, type is string, when True, subscribe all devices' online/offline events
<SubscribeByDeviceName> Device name </SubscribeByDeviceName>	Optional field, type is string. Subscribe online/offline events of specified device by device name
<SubscribeByDeviceId> Device ID </SubscribeByDeviceId>	Optional field, type is URI, defined in 8.1.2. Subscribe online/offline events of specified device by device ID

Message	Field explanation
<SubscribeByServiceType> Service type </SubscribeByServiceType>	Optional field, type is URI, defined in 8.2.3. Subscribe online/offline events of specified device by service type in the device
<SubscribeByDeviceGroupId> Device group ID </SubscribeByDeviceGroupId>	Optional field, type is URI, defined in 8.1.3. Subscribe online/offline events of all devices in a specific device group
<SubscribeByDeviceType> Device type </SubscribeByDeviceType>	Optional field, type is URI, defined in 8.1.4. Subscribe online/offline events of specified device by device type
<SubscribeByServiceName> Service name </SubscribeByServiceName>	Optional field, type is string, subscribe online/offline events of specified device by the service in the device
</SubscriptionFilter>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field
At least one of the fields among SubscribeAll, SubscribeByDeviceName, SubscribeByDeviceId, SubscribeByServiceType, SubscribeByDeviceType, SubscribeByDeviceGroupId and SubscribeByServiceName shall be valid. When SubscribeAll is true, the other fields shall be ignored. If SubscribeAll is false, the search result should match with all fields of search precondition.	

9.7.2 Device online/offline event subscription renewal request

If a client wants to renew the device online/offline subscription event after the previous subscription expires, it shall use the previous subscription ID to send renewal request. If the renewal is successful, the client may continue to subscribe to the event after that event expires. The request message shall be sent based on a Device Pipe. The message format is as follows (seeTable 53).

Table 53 – Device online/offline event subscription renewal request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:RenewSubscriptionDeviceEventRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-SourceDeviceId:Source Device ID that request subscription renewal	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId:Target Device ID that receives subscription renewal request	Required field, type is URI, defined in 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-RenewSubscriptionDeviceEvent-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> Source Client ID </SourceClientId>	Required field, type is unsignedInt (0 is reserved)
<SubscriptionId> subscription ID </SubscriptionId>	Required field, type is unsignedInt (0 is reserved), same as the previous subscription ID
<ExpectedSubscriptionTerm> Expected valid subscription time </ExpectedSubscriptionTerm>	Required field, in seconds, type is unsignedInt (0 is reserved)
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.7.3 Device online/offline event subscription response

After receiving a device event subscription request message, a master device shall return a subscription response message to the device requesting the subscription request. If the relationship of the two devices is not master/slave, the device that receives the subscription request may return a response message to the other party. The device also may ignore or not return a response to the request. The response message shall be sent via the Device Pipe. The response message shall be returned to the requesting device within 30 s after the request is sent. If a device event subscription is successfully, the device that accepted the subscription shall notify the subscriber. The message format is as follows (see Table 54).

Table 54 – Device online/offline event subscription response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessage:SubscribeDeviceEventResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-AcknowledgeId: Response Message sequence ID	Required field, same with the sequence ID in request message
01-TargetDeviceId:Slave Device ID to request subscription	Required field, type is URI, defined in 8.1.2
01-SourceDeviceId:Master Device ID	Required field, type is URI, defined in 8.1.2
Content-type:text/xml;charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-SubscribeDeviceEvent-Response"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<TargetClientId> Target client ID </TargetClientId>	Required field, type is unsignedInt (0 is reserved)
<SubscriptionId> subscription ID </SubscriptionId>	Required field, type is unsignedInt (0 is reserved. Same ID as the SubscriptionId in request message
<ApprovedSubscriptionTerm> Approved valid subscription time </ApprovedSubscriptionTerm>	Required field, in seconds, type is unsignedInt (0 is reserved). If subscription fails, this value is 0
<ReturnCode> subscription Response status code </ReturnCode>	Required field, refer to Clause 11
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.7.4 Device online/offline event unsubscription

A client may initiate the unsubscription of the previously subscribed event; the device event unsubscription is based on the subscription identifier assigned in the previous subscription. The message format is as follows (see Table 55).

Table 55 – Device online/offline event unsubscription message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	Extended command line
Host: Target Host IP: Port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:UnSubscribeDeviceEventNotify	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId:Slave Device ID to unsubscribe	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId:Master Device ID	Required field, type is URI, defined in 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-UnSubscribeDeviceEvent-Notify"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> Source Client ID </SourceClientId>	Required field, type is unsignedInt (0 is reserved)
<SubscriptionId> subscription ID </SubscriptionId>	Required field, type is unsignedInt (0 is reserved), same as the previous SubscriptionId
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.7.5 Device online/offline event notification

If a device event is generated, the device that has accepted the subscription shall send a related device event notification message to the subscriber device in the message format as goven below (Table 56).

Table 56 – Device online/offline event notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	Extended command line
Host: Target Host IP: Port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: NotifyDeviceEvent	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId:Master Device ID	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId:Slave Device ID to request subscription	Required field, type is URI, defined in 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-DeviceEvent-Notify"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> Source client ID </SourceClientId>	Required field, type is unsignedInt (0 is reserved)
<SubscriptionId> subscription ID </SubscriptionId>	Required field, type is unsignedInt (0 is reserved)
<EventType> Device event type for subscription </EventType>	Required field, type is string
<DeviceInfo>	Required field
<!-- Device sample description of device eventthat triggers the event -->	
<DeviceId> Device ID </DeviceId>	Required field, type is URI, defined in 8.1.2
<DeviceName> Device name </DeviceName>	Required field, type is string
<DeviceSecurityIdList> <DeviceSecurityId> Device Security mechanism </DeviceSecurityId> <!--other device security mechanism description--> </DeviceSecurityIdList>	Required field, type is URI, defined in 8.1.5
<ConfigId> Device configuration change ID </ConfigId>	Required field, and type is 32 bit unsignedInt (0 is reserved). The configuration changes, this field indicates the change. ConfigId shall be increased by one (1) whenever there is a configuration change. This field will return to 1 when upper limit is reached.

Message	Field explanation
<BootId> Device Reboot ID </BootId>	Required field, and type is 32 bit UnsignedInt (0 is reserved). When the device reboots, this field indicates this change. BootId value shall be increased by one (1) whenever there is a reboot. This field will return to 1 when upper limit is reached
<SecureListenerList> <SecureListener> IP address and port list for secure Device Pipe </SecureListener> <!--other SecureListener--> </SecureListenerList>	Optional field and the type is string. The mode is “IP:Port”. Mutliple IP addresses can be separated by “;”. Secure device port number is 3880
<ListenerList> <Listener> IP address and port list for non-secure Device Pipe </Listener> <!--other Listener--> </ListenerList>	Optional field and the type is string. The mode is “IP:Port”. Mutliple IP addresses can be separated by “;”. One Listener port number shall be 3880
</DeviceInfo>	Required field
<DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

9.8 Device group search

9.8.1 Device group search request message

A client on an IGRS Device shall be able to search for a device group of interest on the network using specific search conditions based on multicast. The device group search request and response messages are based on the ISDP protocol. The format of a device group search request message is as follows (see Table 57).

Table 57 – Device group search request message

Message	Field Explanation
M-SEARCH * HTTP/1.1	ISDP command line
HOST:239.255.255.250:3880	Required field
MAN:"isdp:discover"	Required field
MX: Max time to wait response	Required field, in seconds, before responding to search request, device will wait for a random time period(from 0 to MX seconds), then sent back the response message.
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SearchDeviceGroupRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SequenceId: Request Message Sequence ID	Required field, type is unsignedInt (0 is reserved)
01-SourceDeviceId: Device ID initiating search	Required field, type is URI, defined in 8.1.2
01-SecureListenerList: IP & port list of secure Device Pipesthat are initiating device searching listening	Optional field and the type is string. The type is "IP: Port" and multiple IP addresses are separated with ";". Secure Device Pipe listener port is fixed at 3880
01-ListenerList: IP and port list of non-secure Device Pipesthat are initiating device searching listening	Required field and the type is string. The type is "IP: Port" and multiple IP addresses are separated with ";". One port number shall be 3880
01-ClientId: ID of client initiating search request	Required field, type is 32-bit unsignedInt (0 is reserved)
01-SearchAll: YES	Optional field, type is string, if searching the device group
01-SearchByDeviceGroupName: Device group name to be searched	Optional field, type is string
01-SearchByDeviceGroupId:Device group ID to be searched	Optional field, type is string, see 8.1.3
01-SearchByDeviceGroupType: Device group type to be searched	Optional field, type is string. Peer means peer-to–peer. Centralised means master-slave
At least one of the fields among SearchAll, SearchByDeviceName, SearchByDeviceGroupId, SearchByDeviceName, SearchByDeviceType, SearchByServiceType, SearchByServiceName and SearchByDeviceId shall be valid. When SearchAll is true, the other fields shall be ignored; if SearchAll is false, the search result should match with all fields of search precondition.	

9.8.2 Device group search response message

The device group search response message shall be returned using UDP unicast by the device responsible for specified the peer-to-peer device group advertisement, or the master device in a centralised device group. The message format is as follows (see Table 58).

Table 58 – Device group search response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP COMMAND LINE
Ext:	Required field
CACHE-CONTROL: max-age: max time of device advertisement expire time;no-cache=”Ext”	Required field, in seconds
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SearchDeviceGroupResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: Device ID that respond to the search request	Required field, type is URI, defined in 8.1.2
01-TargetDeviceId: Target Device ID that initiate the search request	Required field, type is URI, defined in 8.1.2
01-SecureListenerList: IP and port list of secure Device Pipes that are initiating device searching listening	Optional field and the type is string. The type is “IP: Port” and multiple IP addresses are separated with “;”. Secure Device Pipe listener port is fixed at 3880
Content-type:text/xml; charset=utf-8	Required field
Content-length: Message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-SearchDeviceGroup-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=”www.igrs.org/spec1.0”>	Required field
<ReturnCode> Response status code </ReturnCode>	Required field, see Clause 11 for definition
<AcknowledgeId> Searching response device sequence ID </AcknowledgeId>	Required field, the type is 32 bit unsignedInt (0 is reserved). It is same to SequenceId of searching request message
<ClientId> Target client ID </ClientId>	Required field, type is unsignedInt (0 is reserved)
<SearchResult>	Required field
<DeviceGroupInfoList>	Required field
<DeviceGroupInfo>	Required field
<DeviceGroupId> Device Group ID </DeviceGroupId>	Required field, defined in 8.1.3, device group ID that meets search criteria
<DeviceGroupName> Device Group name</DeviceGroupName>	Required field, type is string
<DeviceGroupAdvertisementTerm> Device Group Advertisement Period </DeviceGroupAdvertisementTerm>	Optional field, in seconds, device group advertisement period that meets search criteria, type is unsignedInt
<DeviceGroupType> Device Group type </DeviceGroupType>	Required field, type is string, Peer means peer to peer, Centralised means master-slave
</DeviceGroupInfo>	Required field

Message	Field explanation
<!--other device Group information-->	
</DeviceGroupInfoList>	Required field
</SearchResult>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
<SOAP-ENV:Envelope>	Required field

10 IGRS resource sharing

10.1 Service online advertisement

10.1.1 Service online advertisement based on mutilcast

When an IGRS Service is added to the network, a service advertisement message shall be sent once via multicast. An IGRS Service advertisement mechanism is based on the ISDP protocol mechanism as specified in Clauses A.2 and A.3. The service online advertisement extends the HTTP method and header. The format of a service online advertisement message is defined in Table 59.

Table 59 – Service online advertisement message

Message	Field explanation
NOTIFY * HTTP/1.1	Extended HTTP command line
Host:239.255.255:3880	Required field
Location: url of device description document	Required field, support unsecure pipes among IGRS Devices to get device description documents. Its value is the URL of the documents. When device does not support device description documents access through non-secure pipes, then this field shall be: http://www.igrs.org/device, see Clause B.7
NT: Service Type Identifier	Required field, and type is URI. For definition, see 8.2.3
NTS:isdp:alive	Required field
SERVER: OS/version IGRS/1.0 product/version	Required field
USN: uuid: device ID of advertisement service:: sevice type identifier	Required field, and type is URI. For definition of Device identifierSee 8.1.2; for definition of service type identifier, see 8.2.3
MAN:"http://www.igrs.org/spec1.0"; ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:ServiceOnlineAdvertisement	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: device identifier	Required field, for definition of device identifier, see 8.1.2
01-ServiceName: service name	Required field, and type is string
01-ServiceType: service type	Required field, and type isURI. For definition see 8.2.3
01-ServiceId: Service Identifier	Required field, for definition see 8.2.2
01-ServiceSecurityIDList: Service Security Mechanism descriptor supported by advertisement service	Required field, and type is URI. See 8.2.5. Security descriptors are separated by ";".
01-SecureListenerList: IP address and Port list used to set up security Device Pipe	Optional field, type is string, format is "IP address:port", muliti IP address:port are spaced by ";" listening port is fixed at 3880
01-ListenerList: IP address and Port list used to set up non-security Device Pipe.	Required field, type is string, format is "IP address:port", muliti IP address:port are spaced by ";" one of the port number in the list shall be 3880

When a service goes offline normally, a service offline advertisement message shall be sent to the network.

The message format of a service offline advertisement is defined in Table 60.

Table 60 – Service offline advertisement message

Message	Field explanation
NOTIFY * HTTP/1.1	Extended command line
HOST:239.255.255.250:3880	Required field
NT: Service Type Identifier	Required field, and type is URI, see 8.2.3
NTS:isdp:byebye	Required field, content shall be x, with x being exactly the content of the corresponding left hand column, see ISDP definition
USN:uuid:device identifier of advertising service :: service type Identifier	Required field, and the type is URI, see 8.1.2 and 8.2.3
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version
01-IGRSMessageType: ServiceOfflineAdvetisement	Required field
01-SourceDeviceId: Device ID of advertisement	Required field, see 8.1.2
01-ServiceId: offline service identifier	Required field, and type is unsignedInt (0 is reserved)

10.1.2 Service online registration and offline notification based on device pipe

In a master-slave device group, when a service on a slave device comes online or goes offline, in addition to advertising via multicast, it shall also send a service online registration or offline notification message through the Device Pipe to the master device. After a device joins a master-slave device group successfully, it shall register all of its services with the master device.

The service online registration message is defined in Table 61.

Table 61 – Service online registration notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	HTTP command extension
HOST: target device IP address:port	Required field
Location: detailed service desription URL	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version
01-IGRSMessageType:RegisterServcieNotify	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: source device ID	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device ID	Required field, and type is URI. For definition, see 8.1.2
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-RegisterService-Notify"	Required field
Content-type:text/xml;charset=utf-8	Required field
Content-length: message body length	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop e/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/ encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<ServiceList>	Required field
<ServiceId> registration service ID </ServiceId>	Required field, and type is 32-bit unsignedInt (0 is reserved)
<ServiceName> registration service name </ServiceName>	Required field, and type is string
<ServiceType> service type </ServiceType>	Required field, see 8.2.3
<!-- other service info -->	
</ServiceList>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

The service offline notification message is defined in Table 62.

Table 62 – Service offline notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	HTTP command extension
HOST: target device IP address:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version
01-IGRSMessageType:UnRegisterServcieNotify	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: source device ID	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device ID (master device)	Required field, and type is URI. For definition, see 8.1.2
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-UnRegisterService-Notify"	Required field
Content-type:text/xml;charset=utf-8	Required field
Content-length: message body length	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<ServiceList>	Required field
<ServiceId> registration service ID </ServiceID>	Required field, and type is 32-bit unsignedInt (0 is reserved)
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.2 Service search

10.2.1 Service search based on multicast

IGRS clients may search appropriate services through ISDP. This multicast-based service search request message is defined in Table 63.

Table 63 – Multicast-based service search request message

Message	Field explanation
M-SEARCH * HTTP/1.1	ISDP command line
Host:239.255.255.250:3880	Required field
MAN:"isdp:discover"	Required field
MX: waiting acknowledgement max time	Required field, in seconds. Maximum value 120s
ST:urn:schemas-IGRS-org:service:IGRS-service:1	Required field
Man:"http://www.igrs.org/spec1.0"; ns= 01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version
01-IGRSMessageType: SearchServiceRequest	Required field
01-SourceDeviceId: source device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SequenceId: search message requesting sequence ID	Required field, type is unsignedInt (0 is reserved)
01-clientId: the device identifierthat initiate searching	Required field, and type is unsignedInt
01-SearchAll:TRUE	Optional field, and type is string. Search all services (when TRUE)
01-SearchByDeviceName: device name	Optional field, and type is string. Search all services on some device.
01-SearchByDeviceId: device identifier	Optional field, and type is URI. For definition, see 8.1.2. Search all services on some devices.
01-SearchByServiceType: service type	Optional field, and type is URI. For definition, see 8.2.3. Search all service according to some service types.
01-SearchByDeviceGroupId: device group identifier	Optional field, and type is URI. For definition, see 8.1.3. Search all services in some device groups.
01-SearchByServiceName:service name	Optional field, and type is string. Search all services with some service names.
At least one of the fields among SearchAll, SearchByDeviceName, SearchByDeviceId, SearchByServiceType, SearchByDeviceGroupId and SearchByServiceName shall be valid. When SearchAll is true, the other fields shall be ignored. If SearchAll is false, the search result should match with all fields of search precondition.	

If service located on a device meets the appropriate service search condition, this device shall return a search response based on UDP. For example, this device may be the master device in the master/slave group and includes the services provided by itself and the services on slave devices that it can index in the device group. The message includes the basic service description information, with message format shown in Table 64.

Table 64 – UDP unicast-based service search request message

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control: no-cache=”Ext”	Required field
SERVER: OS/Version IGRS/1.0 Product/version	Required field
Location: url of device description document	Required field, support unsecure pipes among IGRS Devices to get device description documents. Its value is the URL of the documents. When the device does not support device description documents access through non-secure pipes, then this field shall be: http://www.igrs.org/device, see Clause B.7
ST:urn:schemas-IGRS-org:service:IGRS-service:1	Required field
USN:uuid:acknowledgement device identifier::Service type identifier	Required field, and type is URI. For definition, see 8.1.2 and 8.2.3. When searching using service type ID or only one service has responded, service type ID is that particular service type ID, else, the service type ID value shall be “more”
Man:”http://www.igrs.org/spec1.0”; ns= 01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SearchServiceResponse	Required field
01-SourceDeviceId responding device identifier	Required field
01-TargetDeviceId: search device identifier	Required field
01-DeviceSecurityIdList	Required field
01-SecureListenerList	Optional field
01-ListenerList	Required field
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-SearchService-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV=”http://www.w3.org/2002/12/soap-envelope” SOAP-ENV:encodingStyle=”http://schemas.xmlsoap.org/soap/encoding/”>	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=” http://www.igrs.org/spec1.0”>	Required field, see Clause B.1
<AcknowledgeId> response message sequence ID </ AcknowledgeId>	Required field
<TargetClientId> target client identifier </TargetClientId>	Required field, and type is unsignedInt (0 is reserved)
<ReturnCode> search return code </ReturnCode>	Required field
<SearchResult>	Required field
<ServiceInfoList>	Required field
<ServiceInfo>	Required field
<DeviceId> device identifier on that the service is running </DeviceId>	Required field, and type is IGRS Device identifier
<ServiceId> service identifier </ServiceId>	Required field, and type is unsignedInt (0 is reserved)

Message	Field explanation
<ServiceName> service name </ServiceName>	Required field, and type is string
<ServiceType> service type </ServiceType>	Required field, and type is IGRS Service type
<ServiceSecurityIdList> <ServiceSecurityId> service security mechanism description </SeviceSecurityId> <!–– other service security mechanism description --> </ServiceSecurityIdList>	Required field, and type is URI. For definition, see 8.2.5
</ServiceInfo>	Required field
<!-- Other service information -->	
</ServiceInfoList>	Required field
</SearchResult>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.2.2 Service search by proxy

An IGRS client may send service search request to other IGRS Device in the network by the Device Pipe.

The service search request message by proxy is defined in Table 65.

Table 65 – Service search request message by proxy

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP extended command line
HOST:target host IP: port	Required field
MAN:“http://www.igrs.org/spec1.0“; ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: SearchServiceRequestOnDevice	Required field
01-SequenceId: requesting message sequence ID	Required field, and type is unsignedInt (0 is reserved)
01-SourceDeviceId:device identifierthat initiate searching	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId:target device ID	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-SearchServiceRequest-OnDevice”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> source client ID </SourceClientId>	Required field,and type is unsignedInt (0 is reserved)
<SearchAll> TRUE </SearchAll>	Optional field, and type is string. Search all services. When this field is valid, the following searching conditions will not be considered.
<SearchByDeviceName> device name </SearchByDeviceName>	Optional field, and type is string. Search service on the device with some name
<SearchByServiceType> service type </SearchByServiceType>	Optional field, and type is URI. For definition, see 8.2.3. The searching is executed according to search type
<SearchByServiceName> service name </SearchByServiceName>	Optional field, and type is string. Searching is executed according to name
<SearchByDeviceGroupId> device group identifier </SearchByDeviceGroupId>	Optional field, and type is URI. For definition, see 8.1.3. Search service within some device group
<SearchByDeviceId> device identifier </SearchByDeviceId>	Optional field, and type is URI. For definition, see 8.1.2. Search service on some device
</DeviceOperation>	Required field
</SOAP-ENV:Body> </SOAP-ENV:Envelope>	Required field
At least one of the fields among SearchAll, SearchByDeviceName, SearchByDeviceId, SearchByServiceType, SearchByDeviceGroupId and SearchByServiceName shall be valid. When SearchAll is true, the other fields shall be ignored; if SearchAll is false, the search result should match with all fields of search precondition.	

If there is any service that meets the search condition among those indexed by the device receiving the search request, the device may respond according its status. For the master device in the master/slave group, it includes services it provides and services on slave devices that it can index in the device group. This master device shall return a service search response. The other types of devices may respond or ignore the request. The response message shall be returned to the requesting device within 30 s after the request is sent.

The response message format is defined in Table 66.

Table 66 – Service search response message by proxy

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version
01-IGRSMessageType: SearchServiceResponseOnDevice	Required field
01-TargetDeviceId:device identifierthat initiates searching request	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: acknowledgement device ID	Required field, and type is URI. For definition, see 8.1.2
01-AcknowledgeId:acknowledgement message ID	Required field, and type is unsignedInt (0 is reserved). It is same to SequenceId in requesting message
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-SearchServiceResponse-OnDevice”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=“www.igrs.org/spec1.0“>	Required field
<ReturnCode> Response status code </ReturnCode>	Required field, refer to Clause 11
<TargetClientId> target client identifier </TargetClientId>	Required field,and type is unsignedInt (0 is reserved)
<SearchResult>	Required field
<ServiceInfoList>	Required field
<ServiceInfo>	Required field
<DeviceId> device identifier onthat the service is running </DeviceId>	Required field, and type is URI. For definition, see 8.1.2
<ServiceId> service identifier </ServiceId>	Required field, and type is unsignedInt (0 is reserved)
<ServiceName> service name </ServiceName>	Required field, and type is string
<ServiceType> service type </ServiceType>	Required field, and type is URI. For definition see 8.2.3

Message	Field explanation
<ServiceSecurityIdList> <ServiceSecurityId> service security mechanism description </ServiceSecurityId> <!--other service security mechanism descriptor --> </ServiceSecurityIdList>	Required field, and type is URI. For definition, see 8.2.5
<Location> url pointing to service description document </location>	Optional, and type is URI
</ServiceInfo>	Required field
<!--other service information-->	
</ServiceInfoList>	Required field
</SearchResult>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.3 Service online/offline event subscription

10.3.1 Service online/offline event subscription request

An IGRS client may send an online/offline event subscription request message to another IGRS Device to subscribe to all service online/offline events of that device. This message is sent through the Device Pipe. The message format of the service online/offline event subscription request is defined in Table 67.

Table 67 – Service online/offline event subscription request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP extended command line
Host:target host IP:port	Required field
MAN:“http://www.igrs.org/spec1.0“;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SubscribeServiceEventRequest	Required field
01-SequenceId:requesting message sequence ID	Required field, and type is unsignedInt (0 is reserved)
01-SourceDeviceId:the device IDthat initiates request	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId:target device ID	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-SubscribeServiceEvent-Request”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=”http://www.igrs.org/spec1.0”>	Required field, see Clause B.1
<SourceClientId> source client ID </SourceClientId>	Required field, and type is unsignedInt (0 is reserved)
<SubscriptionId> subscription identifier </SubscriptionId>	Required field, and type is unsignedInt (0 is reserved). It is assigned by the subscription requesting device and sent to the subscribed device
<ExpectedSubscriptionTerm> Expected subscription term </ExpectedSubscriptionTerm>	Required field, in seconds
<SubscriptionFilter>	Required field
<SubscribeAll>TRUE </SubscribeAll>	Optional field, and type is string. Subscribe all online and offline eventsthat can be indexed on the device
<SubscribeByServiceName> service name </SubscribeByServiceName>	Optional field, and type is string. Subscribe some online and offline events based on service name
<SubscribeByServiceId> service identifier </SubscribeByServiceId>	Optional field, and type is URI, For definition, see 8.1.2. Subscribe some service online/offline events based on service identifier
<SubscribeByServiceType> service type </SubscribeByServiceType>	Optional field, and type is URI. For definition, see 8.2.3. Subscribe service online/offline events of some service type

Message	Field explanation
<SubscribeByDeviceGroupId> device group identifier </SubscribeByDeviceGroupId>	Optional field, and type is URI. For definition, see 8.1.3. Subscribe all service online/offline events within some device group
<SubscribeByDeviceName> device name </SubscribeByDeviceName>	Optional field, and type is string. Subscribe service online/offline events of some device
<SubscribeByDeviceId> device identifier </SubscribeByDeviceId>	Optional field. For definition, see 8.1.2. Subscribe service online/offline events of some device
</SubscriptionFilter>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field
At least one of the fields among SubscribeAll, SubscribeByDeviceName, SubscribeByDeviceId, SubscribeByServiceType, SubscribeByDeviceGroupId and SubscribeByServiceName shall be valid. When SubscribeAll is true, the other fields shall be ignored; if SubscribeAll is false, the subscription result should match with all fields of search precondition.	

10.3.2 Service online/offline event subscription renewal request

In order to continue a subscription after the current subscription expires, the IGRS Client shall send a subscription renewal request based on the subscription identifier used in the current subscription before it expires. This message is sent through the Device Pipe. The message format of the service online/offline event subscription renewal request is defined in Table 68.

Table 68 – Service online/offline event subscription renewal request

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP extended command line
Host:target host IP:port	Required field
MAN:“http://www.igrs.org/spec1.0“;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:RenewSubscriptionServiceEventRequest	Required field
01-SequenceId:requesting message sequence ID	Required field, and type is unsignedInt (0 is reserved)
01-SourceDeviceId:device IDthat initiates request	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId:target device ID	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-RenewSubscriptionServiceEvent-Request”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=”http://www.igrs.org/spec1.0”>	Required field, see Clause B.1
<SourceClientId> source client ID </SourceClientId>	Required field, and type is unsignedInt (0 is reserved)
<SubscriptionId> subscription identifier </SubscriptionId>	Required field, and type is unsignedInt (0 is reserved). Same as previous subscription ID
<ExpectedSubscriptionTerm> Expected subscription term </ExpectedSubscriptionTerm>	Required field, in seconds
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.3.3 Service online/offiline event subscription response

In a master-slave device group, if the client of the slave device sends a subscription or a subscription renewal request, the master device shall return a subscription response to the client. If no master-slave relationship exists between subscription sending and receiving devices, the device that received the request may return a subscription response. If the subscription was successful, after receiving a service event subscription request, the subscribed device shall return a subscription response to the subscriber device. This response message is sent through the Device Pipe and shall be returned to the requesting device within 30 s after the request is sent. The message format is shown in Table 69.

Table 69 – Service online/offline event subscription response

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessage:SubscribeServiceEventResponse	Required field
01-AcknowledgeId:acknowledgment message sequence ID	Required field, and it is same as the sequence no. of requesting message
01-TargetDeviceId: the device IDthat initiates subscription	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: the subscribed device identifier	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-SubscribeDeviceEvent-Response"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<TargetClientId> target client identifier </TargetClientId>	Required field, and type is unsignedInt (0 is reserved)
<SubscriptionId> subscription identifier </SubscriptionId>	Required field, and type is unsignedInt (0 is reserved). Same as Subscription ID in request message
<ApprovedSubscriptionTerm> approved subscription term </ApprovedSubscriptionTerm>	Required field, in seconds, type is 32-bit unsignedInt (0 is reserved), if subscription failed, this value is 0
<ReturnCode> subscription return status code </ReturnCode>	Required field, and see Clause 11
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.3.4 Service online/offline event unsubscription

The clients may unsubscribe from an event that has been subscribed. The service event unsubscription is based on the subscription identifier assigned previously. This message is sent through the Device Pipe and a response shall not be needed. The message format is defined in Table 70.

Table 70 – Service online/offline event unsubscription message

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended command line
Host:target host IP:port	Required field
MAN:“http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:UnSubscribeDeviceEventNotify	Required field
01-SourceDeviceId: the slave device identifierthat revokes subscription	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device identifier	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-UnSubscribeDeviceEvent-Notify”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> source client identifier </SourceClientId>	Required field, and type is unsignedInt
<SubscriptionId> Subscription identifier </SubscriptionId>	Required field, and type is unsignedInt (0 is reserved)
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.3.5 Service online/offline event notification

When a service online/offline event is triggered, the subscribed device shall send the corresponding event notification to the subscriber device. The message format is defined in Table 71.

Table 71 – Service online/offline event notification message

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended command line
Host:target host IP:port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: NotifyServiceEvent	Required field
01-SourceDeviceId: subscribed device identifier	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId: device identifierthat initiates subscription	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length:message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-ServiceEvent-Notify"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="www.igrs.org/spec1.0">	Required field
<TargetClientId> target client identifier </TargetClientId>	Required field, and type is unsignedInt (0 is reserved)
<SubscriptionId> subscription ID </SubscriptionId>	Required field, and type is unsignedInt (0 is reserved)
<EventType> Online/Offline </EventType>	Required field, and type is string, service event type.
<ServiceInfo>	Required field
<!--the subscribing device simple description information of -->	
<DeviceId> device identifier onthat the service is running </DeviceId>	Required field, and type is URI. For definition, see 8.1.2
<ServiceName> service name </ServiceName>	Required field, and type is string
<ServiceId> service identifier </ServiceId>	Required field, and type is unsignedInt (0 is reserved)
<ServiceType> service type </ServiceType>	Required field, and type is URI. For definition, see 8.2.3
<ServiceSecurityIdList> <ServiceSecurityId> security mechanism adopted by the device </ServiceSecurityId> <!--other service security mechanism descriptor --> </ServiceSecurityIdList>	Required field, and type is URI. For definition, see 8.2.5
</ServiceInfo>	Required field
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.4 Service description document retrieval

10.4.1 Retrieve service description document request

IGRS Clients may get detailed IGRS Service description information by sending a service description retrieval request message. The message format is shown in Table 72.

Table 72 – Retrieve service description document request message

Message	Field explanation
M-GET /IGRS HTTP/1.1	HTTP command line
HOST: target host IP address :port	Required field
ACCEPT-LANGUAGE: expected description language	Recommended field, and the description conforms to RFC 1766.
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field, IGRS version
01-IGRSMessageType:GetServiceDescriptionRequest	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-SourceDeviceId: Source device IDthat is sending request	Required field, and the type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device ID	Required field, and the type is URI. For definition, see 8.1.2
01-SequenceId: Device Pipe message sequence ID	Required field, and the type is 32-bit unsignedInt (0 is reserved)
Content-type:text/xml; charset=utf-8	Required field
Content-length:message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-GetServiceDescription-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/ soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<DeviceOperation xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<ClientId> Client ID who is sending service description request </ClientId>	Required field, and the type is 32-bit unsignedInt (0 is reserved)
<ServiceId> target service ID </ServiceId>	Required field. For definition, see 8.2.2.
<SequenceId> service description request message sequence ID </SequenceId>	Required field, and the type is 32-bit unsignedInt (0 is reserved)
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
/SOAP-ENV:Envelope>	Required field

10.4.2 Retrieve service description document response

After receiving a retrieval service description document request message, the device shall return a response through the Device Pipe to the requesting device within 30 s after the request is sent. The response message format of the device that receives the retrieval service description document request message is defined in Table 73.

Table 73 – Retrieve service description document response message

Message	Field Explanation
HTTP/1.1 200 OK	HTTP command line
CONTENT-LENGTH: message body content length	Required field
CONTENT-TYPE:text/xml; charset=utf-8	Required field
CONTENT-LANGUAGE: documentation description language	When/only when the requesting message includes ACCEPT-LANGUAGE field, and the field is a Required field. For the language description, see RFC 1766
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see Clause B.1
01-IGRSVersion:IGRS/1.0	Required field
01-IGRSMessageType:GetServiceDescriptionResponse	Required field, content shall be x, with x being exactly the content of the corresponding left hand column
01-AcknowledgeId:Device Pipe response acknowledgement ID	Required field, and type is 32-bit unsignedInt (0 is reserved). It is same as “SequenceId” in the request message
01-SourceDeviceId: source device IDthat is sending response	Required field and type is URI. For definition, see 8.1.2.
01-TargetDeviceId: target device ID	Required field, and type is URI. For definition, see 8.1.2.
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-GetServiceDescription-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body>	Required field
<DeviceOperation xmlns=“http://www.igrs.org/spec1.0”>	Required field, see Clause B.1
<ClientId> client ID who is sending response </ClientId>	Required field, and type is 32-bit unsignedInt (0 is reserved)
<ServiceId> Responding service ID	Required field, and type is 32-bit unsignedInt (0 is reserved). For

Message	Field Explanation
</ServiceId>	definition, see 8.2.2
<AcknowledgeId> getting service description response acknowledgement ID </AcknowledgeId>	Required field, and the type is 32-bit unsignedInt (0 is reserved). Same as SequenceId in the request message
<ReturnCode> getting service description return code </ReturnCode>	Required field. For definition, see Clause 11
<ServiceDescription> Service description document </ServiceDescription>	The service description is based on service description template.
</DeviceOperation>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.4.3 Other approaches to retrieve service description documents

Each IGRS Service may define a retrieval interface for specific service description documents by using device description documents. In addition to the approaches introduced in 10.4.1 and 10.4.2, through the provided interface, IGRS Clients may retrieve corresponding service description documents through an unsecure pipe.

10.5 Session

10.5.1 Session setup condition

IGRS clients may get a service security mechanism description via a related IGRS Service advertisement and the detailed service description document. To invoke the service, an IGRS Client shall first set up a session with the IGRS Service according to the requirement described by the service security mechanism.

10.5.2 Common session setup and teardown process

The interactive steps of a session setup process are described as follows.

Step 1

A client generates a token based on the appropriate access control descriptor, authentication mechanism, cryptographic algorithm, etc., and sends a request to set up a session with the target service to the device providing that service using the message format defined in Table 74.

Table 74 – Common session setup request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended HTTP command line
HOST: target host IP: port	Required field
MAN:"http://www.igrs.org/session";ns=01	Required field, see Clause B.2
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:CreateSessionRequest	Required field
01-TargetDeviceId: target device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: source device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SequenceId: Device Pipe message requesting sequence ID	Required field, and type is unsignedInt (0 is reserved)
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-CreateSession-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="www.igrs.org/spec1.0">	Required field
<SourceClientId> source client ID </SourceClientId>	Required field, and type is unsignedInt (0 is reserved)
<TargetServiceId> target service ID </TargetServiceId>	Required field, and type is unsignedInt (0 is reserved)
<SequenceId> session creation sequence ID </SequenceId>	Required field, and type is unsignedInt (0 is reserved)
<UserInfo>	Required field
<SourceUserId> Source user ID </SourceUserId>	Required field, and type is string. For definition, see 8.4
<ServiceSecurityId> service security mechanism description identifier </ServiceSecurityId>	Required field. For definition, see 8.2.5
<Token> the Token content generated by user </Token>	Required field, and type is base64binary
<!-- other info about the client-->	Optional
</UserInfo>	Required field
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

A token is a kind of certificate with a special trust attribute. The token defined in this protocol is used mainly for identity authentication during the session setup. The setup and structure are defined in Table 75.

Table 75 – Token setup and structure

Selected status authentication mechanism	*A*uthenticationAlgorithm	Token
URN:IGRS:Security:NULL		
URN:IGRS:ServiceSecurity:PreSharedKey	PreSharedKeyAuthen	{nonce, PreSharedKeyAuthen(nonce|| client ID, preshared secret key)}
URN:IGRS: ServiceSecurity:PKICertificate	PublicKeyAuthen	{nonce, client public key certificate, (signature result of message {nonce||client ID} using private key based on PublicKeyAuthen)}
URN:IGRS: ServiceSecurity:thirdPartyAuthenService	Kerberosv5	Generate Ticket based on the third party authen secret key algorithm

Step 2

After receiving a session setup request from a client, the service provisioning device shall check the validity of the token in the request based on the user identifier of the request and the user authentication algorithm. If the check is valid, the service provisioning device shall return “session setup succeed” response; otherwise, it shall send a “session setup fail” response. The response message shall be returned to the requesting device within 30 s after the request is sent. The format is defined in Table 76.

Table 76 – Common session setup response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/session”;ns=01	Required field, see Clause B.2
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType: CreateSessionResponse	Required field
01-TargetDeviceId: target device ID	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: source device ID	Required field, and type is URI. For definition, see 8.1.2
01-AcknowledgeId: Device Pipe acknowledgement sequence ID	Required field, and type is unsignedInt (0 is reserved). It is same as the Device Pipe SequenceId in the request
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-CreateSession-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<SourceServiceId> source service ID </SourceServiceId>	Required field, and type is unsignedInt (0 is reserved)
<TargetClientId> target client ID </TargetClientId>	Required field, and type is unsignedInt (0 is reserved)
<TargetUserId> target user ID </TargetUserId>	Required field, and type is string. For definition, see 8.4
<AcknowledgeId> acknowledgement ID </AcknowledgeId>	Required field, and type is unsignedInt. (0 is reserved) It is same as the SequenceID in the request message body
<ReturnCode> return code of the session creation process </ReturnCode>	Required field, see Clause 11
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

Step 3

Tear down the session and generate a notification using the message format defined in Table 77.

Table 77 – Common session teardown notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	Extended HTTP command line
Host: host IP address:port	Required field
MAN:"http://www.igrs.org/session";ns=01	Required field, see Clause B.2
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:DestroySessionNotify	Required field
01-SourceDeviceId: source device identifier	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId:target device identifier	Required field, and type is URI. For definition, see 8.1.2
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-DestroySession-Notify"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="www.igrs.org/spec1.0">	Required field
<SourceServiceId> source service identifer </SourceServiceId>	Optional field, and type is unsignedInt (0 is reserved). When service side is ending the session, this field is required, else this field does not exist
<SourceClientId> source client identifer </SourceClientId>	Optional field, and type is unsignedInt (0 is reserved). When client side is ending the session, this field is required, else this field does not exist
<TargetServiceId> target service ID </TargetServiceId>	Optional field, and type is unsignedInt (0 is reserved). When client side is ending the session, this field is required, else this field does not exist
<TargetClientId> target client ID </TargetClientId>	Optional field, and type is unsignedInt (0 is reserved). When service side terminates the session this field is required. Otherwise the field does not exist
<Token> Token of session </Token>	Required field, and type is base64binary
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.5.3 Session setup when service access control in master/slave device group is not consistent with device pipe security attribute

When the access security control of a service requires a secure Device Pipe between the requesting client device and the device that provides this service, but the Device Pipe setup between these two devices does not agree with this requirement, and if both devices belong to the same master/slave device group, the session shall be set up according to the following steps.

a) Same as step 1 in 10.5.2.

b) Same as step 2 in 10.5.2, and the response status code in message shall be “Failed device authority”.

c) The client selects the appropriate service security mechanism descriptor based on a trusted third party. The descriptor is selected from the Encryption Algorithm list supported by the service description of the service provider. The client encapsulates the client device identifier, service provider device identifier, and the Encryption Algorithm description ID into a request to the master device for session encryption key generation. The message format is defined in Table 78.

Table 78 – Retrieve session encryption key generation request

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended HTTP command line
HOST: target host IP: port	Required field
MAN:"http://www.igrs.org/session";ns=01	Required field, see Clause B.2
01-IGRSVersion:IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:ApplySessionKeyRequest	Required field
01-TargetDeviceId: target device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: source device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SequenceId: Device Pipe message requesting ID	Required field, and type is unsignedInt (0 is reserved)
Content-type: text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-ApplySessionKey-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="www.igrs.org/spec1.0">	Required field
<ClientId> source client Identifier </ClientId>	Required field, and type is unsignedInt (0 is reserved)
<ServiceId> target service identifier </ServiceId>	Required field, and type is unsignedInt (0 is reserved)
<AuthenServiceId> Authen Service Identifier </AuthenServiceId>	Required field, and type is unsignedInt
<SequenceId> requesting session encryption secret key sequence ID </SequenceId>	Required field, and type is unsignedInt (0 is reserved)
<ServiceHOSTDeviceId> Target service device ID </ServiceHOSTDeviceId>	Required field, for definition see 8.1.2
<ServiceSecurityId> Service security mechanism description identifier </ServiceSecurityId>	Required field, and type is URI. For definition, see 8.2.5
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

d) After the master device receives a session encryption key generation request from a client, it shall generate a random bit string in proper length to be the session encryption key based on the Encryption Algorithm in the request. It shall also encrypt the session encryption key generated, client identifier, and related Encryption Algorithm description ID into Cipher 1 with the pre-shared key security mechanism set up between the master device and client device. Meanwhile, it encrypts {service identifier, client identifier, service security mechanism identifier based on trusted third party that is supported by both the client and target service, session encryption key} into Cipher 2 with the pre-shared key and mutually shared encryption algorithm between the master device and service provider

device. It then sends Cipher 1 and Cipher 2 to the client. This message shall be returned to the requesting device within 30 s after the request is sent. The message format is defined in Table 79.

Table 79 – Retrieve session encryption key generation response

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/session”;ns=01	Required field, see Clause B.2
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:ApplySessionKeyResponse	Required field
01-TargetDeviceId:target device identifier	Required field, see 8.1.2
01-SourceDeviceId:source device identifier	Required field, see 8.1.2
01-AcknowledgeId:Device Pipe acknowledgement sequence ID	Required field, and it is the same as the request sequence ID of Device Pipe
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-ApplySessionKey-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="www.igrs.org/spec1.0">	Required field
<ServiceId> source service identifier </ServiceId>	Required field, and type is unsignedInt (0 is reserved)
<ClientId> target client identifier </ClientId>	Required field, and type is unsignedInt (0 is reserved)
<ReturnCode> return code of session encryption secret key </ReturnCode>	Required field, see Clause 11
<AcknowledgeId> sequence ID of the acknowledgement session encryption secret key </AcknowledgeId>	Required field, and it is the same as the sequence ID of the session encryption secret key
<Cipher1> cipher1 from master device </Cipher1>	Required field, type is base64 binary
<TargetServiceHOSTDeviceId> target service slave device identifier </TargetServiceHOSTDeviceId>	Required field, see 8.1.2
<TargetServiceId> target service identifier on the slave device </TargetServiceId>	Required field, type is 32-bit unsignedInt (0 is reserved)
<Cipher2> Cipher2 from master device </Cipher2>	Required field, type is base64 binary
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

e) After the client receives the session encryption key retrieval response from the master device, it shall extract message Cipher 1 and Cipher 2, and decrypt Cipher 1 with pre-

shared key of master device to retrieve session encryption key. It then sends Cipher 2 to the service provider device. The message format is defined in Table 80.

Table 80 – Session encryption key transfer request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended HTTP command line
Host:target host IP:port	Required field
MAN:"http://www.igrs.org/session";ns=01	Required field, see Clause B.2
01-IGRSVersion: IGRS/1.0	Required field, IGRS version
01-IGRSMessageType:TransferSessionKeyRequest	Required field
01-TargetDeviceId:target device ID	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId:source device ID	Required field, and type is URI. For definition, see 8.1.2
01-SequenceId: Device Pipe requesting message ID	Required field, and type is unsignedInt (0 is reserved)
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-TransferSessionKey-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="www.igrs.org/spec1.0">	Required field
<ClientId> client Identifier </ClientId>	Required field, and type is unsignedInt (0 is reserved)
<ServiceId> target service identifier </ServiceId>	Required field, and type is unsignedInt (0 is reserved)
<SequenceId> sequence ID of session encryption key transfer message </SequenceId>	Required field, and type is unsignedInt (0 is reserved)
<TrustedthirdDeviceId> trusted third party device identifier </TrustedthirdDeviceId>	Required field, see 8.1.2
<Cipher2> cipher 2 </Cipher2>	Required field, type is base64 binary
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

f) After the service provider receives the session encryption key request and extracts Cipher 2, it shall decrypt Cipher 2 with the pre-shared key between itself and the master device. It shall confirm the client identifier, service security mechanism identifier based on a trusted third party that is supported by both the client and target service, and the session encryption key. Then a response message shall be sent to the client within 30 s after the request was sent. The client shall be set as a trusted device immediately. The response message format is defined in Table 81.

Table 81 – Session encryption key transfer response message

Message	Field Explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control:no-cache=”Ext”	Required field
MAN:”http://www.igrs.org/session”;ns=01	Required field, see Clause B.2
01-IGRSVersion: IGRS/1.0	Required field, IGRS version
01-IGRSMessageType:TransferSessionKeyResponse	Required field
01-TargetDeviceId: target device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: source device identifier	Required field, and type is URI. For definition, see 8.1.2
01-AcknowledgeId: Device Pipe acknowledgement message sequence ID	Required field, and it is the same as the sequence ID of the Device Pipe request message
Content-type:text/xml; charset=utf-8	Required field
Content-length: message body length	Required field
MAN:”http://www.w3.org/2002/12/soap-envelope”;ns=02	Required field
02-SoapAction:”IGRS-TransferSessionKey-Response”	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="www.igrs.org/spec1.0">	Required field
<TargetClientId> target client identifier </TargetClientId>	Required field, and type is unsignedInt (0 is reserved)
<ReturnCode> response status code </ReturnCode>	Required field, see Clause 11
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

g) After the client receives the session encryption key transfer response from the service provider, it shall regenerate a session setup request based on the access control strategy requirement from the service provider. The client sends this request to the service provider device after encrypting with the session encryption key. The message format is same as in step 1 of 10.5.2.

h) The service provider device decrypts for the session setup received with the session encryption key, and implements a validity check of the Token in the request based on the user identifier and user authentication algorithm in the request. If the check is valid, it shall return a session setup success response to the client; otherwise it shall return a session setup fail response. The response format is the same as in step 2 of 10.5.2. The response shall also be encrypted with the session key. If the session setup is enabled, the message interacting on this session shall be encrypted and transmitted with the session encryption key until this session is torn down.

i) Session teardown. The message format is the same as in step 3 of 10.5.2.

10.6 Service invocation

10.6.1 Service invocation request message

The client may invoke a target service based on the session setup between the client and the target service. For the definition of the service invocation request message, see Table 82.

Table 82 – Service invocation request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	Extended HTTP command line
Host: target host IP address: port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:InvokeServiceRequest	Required field
01-SourceDeviceId: requesting device identifier	Required field, and type is URI. For definition, see 8.1.2
01-TargetDeviceId: target device identifier	Required field, and type is URI. For definition, see 8.1.2
Content-Type:text/xml; charset=utf-8	Required field
Content-Length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-InvokeService-Request"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<SourceClientId> source client identifier </SourceClientId>	Required field, and type is unsignedInt (0 is reserved)
<TargetServiceId> target service identifier </TargetServiceId>	Required field, and type is unsignedInt (0 is reserved)
<SequenceId> invocation sequence ID </SequenceId>	Required field, and type is unsignedInt (0 is reserved)
<!--Here is the specific invocation acknowledgement message-->	
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.6.2 Service invocation response message

The response message is defined in Table 83.

Table 83 – Service invocation response message

Message	Field explanation
HTTP/1.1 200 OK	HTTP command line
Ext:	Required field
Cache-control:no-cache="Ext"	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:InvokeServiceResponse	Required field
01-TargetDeviceId: target device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: requesting device identifier	Required field, and type is URI. For definition, see 8.1.2
Content-Type:text/xml; charset=utf-8	Required field
Content-Length: Message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-InvokeService-Response"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="http://www.igrs.org/spec1.0">	Required field, see Clause B.1
<SourceServiceId> source service identifier </SourceServiceId>	Required field, and type is unsignedInt (0 is reserved)
<TargetClientId> target client identifier </TargetClientId>	Required field, and type is unsignedInt (0 is reserved)
<AcknowledgeId> invocation response sequence ID </AcknowledgeId>	Required field, and type is unsignedInt (0 is reserved), same as the SequenceID in the invocation request
<ReturnCode> response status code </ReturnCode>	Required field, see Clause 11
<!--Here is the specific invocation acknowledgement message-->	
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.6.3 Notification message based on session

In an IGRS session, clients and services may send notification messages (see 7.2.1) to each other. The message is defined in Table 84.

Table 84 – Notification message based on session

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	Extended HTTP command line
Host: target host IP address: port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see Clause B.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:SendNotification	Required field
01-TargetDeviceId:target device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SourceDeviceId: source device identifier	Required field, and type is URI. For definition, see 8.1.2
01-SequenceId: Device pipe acknowledgement message sequence ID	Required field, and type is unsignedInt (0 is reserved). Same as the request message Device Pipe ID
Content-Type:text/xml; charset=utf-8	Required field
Content-Length: message body length	Required field
MAN:"http://www.w3.org/2002/12/soap-envelope";ns=02	Required field
02-SoapAction:"IGRS-Send-Notification"	Required field
	Shall be empty
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2002/12/soap-envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns="http://www.igrs.org/spec1.0" request="NoResponse">	Required field, "request" attribute indicates that no response is needed, see Clause B.1
<SourceSessionEndpoint> source service or client identifer </SourceSessionEndpoint>	Required field, and type is unsignedInt (0 is reserved). If message is sent from client to service, this value is client ID; otherwise, it is service ID
<TargetSessionEndpoint> target client or service ID </TargetSessionEndpoint>	Required field, and type is unsignedInt (0 is reserved). If message is sent from service to client, this value is client ID; otherwise, it is service ID
<!--Here is the specific event information -->	
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

11 Request/response status codes

In IGRS request/response messages, the response status code in the response message is composed of three integers. The specific category and definition are defined in Table 85 and Table 86.

Table 85 – Response status code category

Response Status Code	Content
1xx	General information response status code
2xx	Device (device group) related message response status code
3xx	Service related message response status code
4xx	Session related message response status code
5xx	Security related status code
6xx	Internal error code

Table 86 – Response status code definition

Response status code	Content
100	Success
101	Internal service failure
102	Target device protocol version not supported
201	Requested device prohibited
203	Device group joining failed, authentication failed
204	Device detailed description information access failed, device detailed description information does not exist or is blank
205	Device online/offline event subscription failed
206	Device (Service) online/offline event subscription does not exist
207	No matched results
303	Service invocation failed, unidentified invocation request
304	Service detailed description information does not exist or blank
305	Service invocation failed, session is not setup
307	Subscription does not exist
309	Status data requested to be subscribed does not exist
400	Session setup failed, authentication failed
401	Target service does not exist
402	Failed user authority
403	Failed device authority
404	Device pipe setup failed, authentication failed
405	Device pipe setup failed, target device does not exist
406	Concurrence number exceeds limit, session setup failed
501	Security mechanism has been selected, but negotiation specific to the actual security mechanism is needed
502	Security negotiation request ignored
503	Security negotiation process suspended
504	Security negotiation failed
505	ID authentication failed
506	Ticket request failed
507	Ticket transmission failed
508	ACL checking failed
600	Unmatched protocol version, request message unresolved

Annex A
(normative)

IGRS service discovery protocols (ISDP)

A.1 General

Based on Simple Service Discovery Protocol (SSDP), the IGRS Service Discovery Protocol (ISDP) extends SSDP as specified in Clause 1 of ISO/IEC 29341-1 with the additional device group discovery specified in Clause A.2 and Clause A.3. ISDP provides a mechanism whereby network clients and client groups, with little or no static configuration, can discover network services.

A.2 ISDP message format

A.2.1 General

ISDP uses part of the header format of HTTP 1.1 as defined in RFC 2616. However, it is NOT based on full HTTP 1.1 as it uses UDP instead of TCP, and it has its own processing rules.

All ISDP messages shall be formatted according to 4.1 “Message Types” of RFC 2616. ISDP messages shall have a start-line and a list of message headers. While ISDP shares the definitions of SSDP start-lines and message header, it also has additional requirements. ISDP messages shall only use one of the three start-lines as defined in A.2.2. For each type of ISDP messages, ISDP also restricts the number of allowed fields and message headers as defined in A.2.3.

A.2.2 ISDP start-lines

Each ISDP message shall have exactly one start-line. See A.2.3 and A.2.4 below for the definition of all possible ISDP messages. The start-line shall be formatted either as defined in 5.1 “Request-Line” or 6.1 “Status-Line” of RFC 2616. Furthermore, the start-line shall be one of the following three. Please refer to SSDP and HTTP extensions (defined in IETF RFC 2774) for more details.

NOTIFY * HTTP/1.1\r\n

M-SEARCH * HTTP/1.1\r\n

HTTP/1.1 200 OK\r\n

A.2.3 ISDP message headers

The message headers in an ISDP message shall be formatted according to 4.2 “Message Headers” of RFC 2616. This specifies that each header field consists of a name followed by a colon (":") and the field value. ISDP restricts allowed field values; they should not span multiple lines.

The following HTTP headers shall be included in a NOTIFY * HTTP/1.1\r\n ISDP message:HOST,CACHE-CONTROL,LOCATION,NT,NTS,SERVER,USN. Please refer to SSDP and HTTP extensions for more details.

The following HTTP headers shall be included in an M-SEARCH * HTTP/1.1\r\n ISDP message:HOST,MAN,MX,ST. Please refer to SSDP and HTTP extensions for more details.

A.2.4 ISDP processing rules

When an ISDP message is received that is not formatted according to A.2.2, receivers should silently discard the message. Receivers may try to parse such ISDP messages to try to interoperate.

When parsing ISDP headers, receivers shall parse all required ISDP-defined header fields (see A.2.2 and A.2.3) and may skip all other header fields. Receivers shall be able to skip header fields they do not understand.

A.3 ISDP usage in IGRS specification

This clause defines the ISDP protocol used in IGRS Device Advertisement, Device Search, Device Group Advertisement, Device Group Search, Service Advertisement and Service Search messages.

ISDP supports Device Group Advertisement and Device Group Search. Some of the field values required by HTTP headers are defined below.

HOST

Required. Multicast channel shall be 239.255.255.250:3880.

CACHE-CONTROL

Required. Used in device and device group advertisement mechanisms.

LOCATION

Required. Contains a URL to description documents of devices, device groups or services. The formats of IGRS device and service description documents are defined in 8.1.1 and 8.2.1

NT

Notification Type. Shall be one of the following:
- uuid:device identifier
  - Type is URI, refer to 8.1.2
- uuid:device group identifier
  - Definition of device refer to 8.1.3
- uuid:service type identifier
  - Type is URI, refer to 8.2.3

NTS

Notification Sub-type. Shall be one of the following:
- isdp:alive
- isdp:byebye
- isdp:groupalive
- isdp:groupbyebye
- isdp:quitgroup

SERVER

OS name/OS version, **IGRS/1.0**, product name/product version.

USN

Unique Service Name. Identifies a unique instance of a device or service. Shall be one of the following:
- uuid:device identifier
  - Type is URI, refer to 8.1.2.
- uuid:device group identifier
  - Definition of device refer to 8.1.3.
- uuid:service type identifier
  - Type is URI, refer to 8.2.3

uuid:device identifier:device type identifier
device identifier defined in 8.1.2, and device type identifier defined in 8.1.4.
uuid:device identifier:service type identifier
device identifier defined in 8.1.2, and service type identifier defined in 8.2.3.

ST

Search Target. Required by device and service search messages. Shall be one of the following:
urn:schemas-IGRS-org:device:IGRS-device:1
urn:schemas-IGRS-org:service:IGRS-service:1

Annex B
(normative)

Description documents

B.1 Specification description

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:igrsExt="http://www.igrs.ogr/igrs/ServiceDescription"
xmlns:igrs="http://www.igrs.org/igrs/ContentIndex"
targetNamespace="http://www.igrs.org/igrs/ContentIndex" name="ContentIndex">
        <types>
                <element name="Type_ObjectId" type="xsd:string"/>
                <element name="Type_ContentList" type="ContentListType"/>
                <element name="Type_ContentListScale" type="xsd:string"/>
                <element name="Type_FilterRule" type="xsd:string"/>
                <element name="Type_BrowseFlag" type="BrowseFlagType"/>
                <element name="Type_SortRule" type="xsd:string"/>
                <element name="Type_Count" type="xsd:Int" minOccurs="0"/>
                <element name="Type_TransferInstanceId" type="xsd:unsignedInt"
minOccurs="0"/>
                <element name="Type_TransferState" type="TransferStateType"
minOccurs="0"/>
                <element name="Type_Length" type="xsd:string" minOccurs="0"/>
                <element name="Type_TagList" type="xsd:string" minOccurs="0"/>
                <element name="Type_URI" type="xsd:string" minOccurs="0"/>
                <element name="Type_ContentUpdateId" type="xsd:unsignedInt"/>

                <element name="Type_MediaFormat" type="MediaFormatType"/>
                <element name="Type_ServiceAttributeName"
type="ServiceAttributeNameType"/>
                <element name="Type_SubscriptionId" type="xsd:unsignedInt"/>
                <element name="Type_SearchAttributeName" type="xsd:string"/>
                <element name="Type_AttributeValueList" type="xsd:string"/>
                <element name="Type_SearchCapabilityList">
                        <complexType>
                                <sequence>
                                        <element name="SearchCapability" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                                </sequence>
                        </complexType>
                </element>
                <element name="Type_SortCapabilityList">
                        <complexType>
                                <sequence>
                                        <element name="SortCapability" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                                </sequence>
                        </complexType>
                </element>
                <element name="Type_SearchAttributeValueList">
                        <complexType>
                                <sequence>
                                        <element name="SearchAttributeValue"
type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
                                </sequence>
```

```
                </complexType>
            </element>

            <complexType name="ContentListType">
                <sequence>
                    <element name="Container" type="igrs:ContainerType"
minOccurs="0" maxOccurs="unbounded"/>
                    <element name="Item" type="igrs:ItemType" minOccurs="0"
maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <complexType name="ContainerType">
                <sequence>
                    <element name="ContainerAttribute"
type="igrs:ContainerAttributeType"/>
                </sequence>
                <attribute name="Num_containers" type="xsd:string"/>
                <attribute name="Num_items" type="xsd:string"/>
            </complexType>
            <complexType name="ContainerAttributeType">
                <sequence>
                    <element name="ObjectId" type="xsd:string"/>
                    <element name="ParentId" type="xsd:string" minOccurs="0"/>
                    <element name="DeviceId" type="xsd:string" minOccurs="0"/>
                    <element name="DeviceName" type="xsd:string"
minOccurs="0"/>
                    <element name="ObjectPath" type="xsd:string"
minOccurs="0"/>
                    <element name="ObjectName" type="xsd:string"/>
                    <element name="ObjectStoreAttribute">
                        <complexType>
                            <sequence>
                                <element name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                                <element name="Hide" type="xsd:string"
minOccurs="0"/>
                                <element name="Document"
type="xsd:string" minOccurs="0"/>
                                <element name="Content"
type="xsd:string" minOccurs="0"/>
                            </sequence>
                        </complexType>
                    </element>
                    <element name="CreateTime" type="xsd:string"
minOccurs="0"/>
                    <element name="LastAccessTime" type="xsd:string"
minOccurs="0"/>
                    <element name="LastWriteTime" type="xsd:string"
minOccurs="0"/>
                    <element name="AdditionalInfo" type="xsd:string"
minOccurs="0"/>
                </sequence>
            </complexType>
            <complexType name="ItemType">
                <sequence>
                    <element name="ItemAttribute" type="igrs:ItemAttributeType"/>
                    <element name="ItemReferences" minOccurs="0">
                        <complexType>
                            <sequence>
                                <element name="KeyFrame"
minOccurs="0">
```

```
                                                <attribute name="Id"
type="igrs:Type_ObjectId"/>
                                            </element>
                                            <element name="MicroPhotograph"
minOccurs="0">
                                                <attribute name="Id"
type="igrs:Type_ObjectId"/>
                                            </element>
                                        </sequence>
                                    </complexType>
                                </element>
                            </sequence>
                        </complexType>
                        <complexType name="ItemAttributeType">
                            <sequence>
                                <element name="Type_ObjectId" type="xsd:string"/>
                                <element name="ParentId" type="xsd:string" minOccurs="0"/>
                                <element name="DeviceId" type="xsd:string" minOccurs="0"/>
                                <element name="DeviceName" type="xsd:string"
minOccurs="0"/>
                                <element name="ObjectPath" type="xsd:string"
minOccurs="0"/>
                                <element name="ObjectName" type="xsd:string"/>
                                <element name="ObjectStoreAttribute">
                                    <complexType>
                                        <sequence>
                                            <element name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                                            <element name="Hide" type="xsd:string"
minOccurs="0"/>
                                            <element name="Document"
type="xsd:string" minOccurs="0"/>
                                            <element name="Content"
type="xsd:string" minOccurs="0"/>
                                        </sequence>
                                    </complexType>
                                </element>
                                <element name="CreateTime" type="xsd:string"
minOccurs="0"/>
                                <element name="LastAccessTime" type="xsd:string"
minOccurs="0"/>
                                <element name="LastWriteTime" type="xsd:string"
minOccurs="0"/>
                                <element name="ObjectType" type="igrs:ObjectTypeType"
minOccurs="0"/>
                                <element name="Size" type="xsd:string" minOccurs="0"/>
                                <element name="Duration" type="xsd:string" minOccurs="0"/>
                                <element name="Encrypted" type="xsd:string" minOccurs="0"/>
                            <element name="DRMContendId" type="xsd:string" minOccurs="0"/>
                                <element name="Type_MediaFormat"
type="MediaFormatType"/>
                                <element name="ObjectURI" type="xsd:string"/>
                                <element name="Type_ConnectionManagementServiceId"
type="xsd:unsignedInt"/>
                                <element name="Choices" minOccurs="0">
                                    <complexType>
                                        <sequence>
                                            <element name="Choice" minOccurs="0"
maxOccurs="unbounded">
                                                <complexType>
                                                    <sequence>
```

```
                                                                    <element
name="Selection" minOccurs="0" maxOccurs="unbounded">
        <complexType>
        <attribute name="Id" type="xsd:string"/>
        <attribute name="Name" type="xsd:string"/>
        </complexType>
                                                                    </element>
                                                            </sequence>
                                                            <attribute      name="Id"
type="xsd:string"/>
                                                            <attribute   name="Name"
type="xsd:string"/>
                                                            <attribute name="Default"
type="xsd:string"/>
                                                            <attribute    name="Type"
type="xsd:string"/>
                                                    </complexType>
                                            </element>
                                    </sequence>
                            </complexType>
                    </element>
            </sequence>
    </complexType>
    <complexType name="ObjectTypeType">
            <choice>
                    <element name="Audio" type="igrs:AudioType"/>
                    <element name="Video" type="xsd:string"/>
                    <element name="Photo" type="xsd:string"/>
                    <element name="Doc" type="xsd:string"/>
                    <element name="Camera" type="xsd:string"/>
                    <element name="Screen" type="xsd:string"/>
            </choice>
    </complexType>
    <complexType name="AudioType">
            <sequence>
                    <element name="Singer" type="xsd:string" minOccurs="0"/>
                    <element name="Genre" type="xsd:string" minOccurs="0"/>
                    <element name="MusicDisc" type="xsd:string" minOccurs="0"/>
                    <element name="Author" type="xsd:string" minOccurs="0"/>
            </sequence>
    </complexType>

    <complexType name="BrowseFlagType">
            <choice>
                    <element                    name="Constant_ContainerSelfInfo"
type="xsd:string"/>
                    <element                name="Constant_ContainerChildrenInfo"
type="xsd:string"/>
            </choice>
    </complexType>

    <complexType name="TransferStateType">
            <choice>
                    <element name="IN_PROGRESS" type="xsd:string"/>
                    <element name="STOPPED" type="xsd:string"/>
                    <element name="ERROR" type="xsd:string"/>
                    <element name="COMPLETED" type="xsd:string"/>
            </choice>
```

```
            </complexType>

            <complexType name="MediaFormatType">
                  <sequence>
                        <element          name="AudioFormat"          type="xsd:string"
minOccurs="0"/>
                        <element          name="VideoFormat"          type="xsd:string"
minOccurs="0"/>
                        <element          name="PhotoFormat"          type="xsd:string"
minOccurs="0"/>
                        <element     name="RemoteDesktop"     type="RemoteDesktop"
minOccurs="0"/>
                        <element       name="VideoCamera"       type="VideoCamera"
minOccurs="0"/>
                  </sequence>
                  <attribute name="Name" type="xsd:string"/>
                  <attribute name="Type" type="xsd:string"/>
            </complexType>
            <complexType name="ServiceAttributeNameType">
                  <choice>
                        <element name="CONTENTUPDATEID" type="xsd:string"/>
                        <element name="SORTCAPS" type="xsd:string"/>
                        <element name="OBJECTID" type="xsd:string"/>
                        <element name="TRANSFERSTATE" type="xsd:string"/>
                  </choice>
            </complexType>
      </types>

      <message name="GetSearchCapabilityListRequest"/>
      <message name="GetSearchCapabilityListResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="SearchCaps" ref="igrs:Type_SearchCapabilityList"/>
      </message>
      <message name="GetSearchAttributeCapabilityListRequest"/>
      <message name="GetSearchAttributeCapabilityListResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part                                          name="SearchAttributeCaps"
ref="igrs:Type_SearchAttributeCapabilityList"/>
      </message>
      <message name="GetSortCapabilityListRequest"/>
      <message name="GetSortCapabilityListResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="SortCaps" ref="igrs:Type_SortCapabilityList"/>
      </message>
      <message name="GetContentUpdateIdRequest"/>
      <message name="GetContentUpdateIdResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ContentUpdateId" ref="igrs:Type_ContentUpdateId"/>
      </message>
      <message name="BrowseRequest">
            <part name="ObjectId" ref="igrs:Type_ObjectId"/>
            <part name="BrowseFlag" ref="igrs:Type_BrowseFlag"/>
            <part name="BrowseRule" ref="igrs:Type_FilterRule"/>
            <part name="Offset" ref="igrs:Type_Offset"/>
            <part name="RequestCount" ref="igrs:Type_Count"/>
            <part name="SortRule" ref="igrs:Type_SortRule"/>
         <part name="ResultScale" ref="igrs:Type_ContentListScale"/>
      </message>
      <message name="BrowseResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="Result" ref="igrs:Type_ContentList"/>
            <part name="NumberReturned" ref="igrs:Type_Count"/>
```

```
        <part name="ContainerNumberTotal" ref="igrs:Type_Count"/>
        <part name="ItemNumberTotal" ref="igrs:Type_Count"/>
</message>
<message name="SearchRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="SearchRule" ref="igrs:Type_SearchRule"/>
        <part name="Offset" ref="igrs:Type_Offset"/>
        <part name="RequestCount" ref="igrs:Type_Count"/>
        <part name="SortRule" ref="igrs:Type_SortRule"/>
</message>
<message name="SearchResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="Result" ref="igrs:Type_ContentList"/>
        <part name="NumberReturned" ref="igrs:Type_Count"/>
        <part name="NumberTotalMatched" ref="igrs:Type_Count"/>
</message>
<message name="SearchAttributeValueRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="SearchAttributeName" type="xsd:string"/>
        <part name="Offset" ref="igrs:Type_Offset"/>
        <part name="RequestCount" ref="igrs:Type_Count"/>
</message>
<message name="SearchAttributeValueResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="AttributeValueList" type="Type_AttributeValueList"/>
        <part name="NumberReturned" ref="igrs:Type_Count"/>
        <part name="NumberTotalMatched" ref="igrs:Type_Count"/>
</message>
<message name="ConvertMediaFormatRequest">
    <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="CurrentMediaFormat" ref="igrs:Type_MediaFormat"/>
        <part name="TargetMediaFormat" ref="igrs:Type_MediaFormat"/>
</message>
<message name="ConvertMediaFormatResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="CreateObjectRequest">
        <part name="ContainerId" ref="igrs:Type_ObjectId"/>
        <part name="Elements" ref="igrs:Type_ContentList"/>
</message>
<message name="CreateObjectResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="Result" ref="igrs:Type_ContentList"/>
</message>
<message name="DestroyObjectRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
</message>
<message name="DestroyObjectResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="UpdateObjectRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="CurrentTag" ref="igrs:Type_TagList"/>
        <part name="NewTag" ref="igrs:Type_TagList"/>
</message>
<message name="UpdateObjectResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="ImportResourceRequest">
        <part name="SourceURI" ref="igrs:Type_URI"/>
        <part name="DestinationURI" ref="igrs:Type_URI"/>
```

```
</message>
<message name="ImportResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="ExportResourceRequest">
        <part name="SourceURI" ref="igrs:Type_URI"/>
        <part name="DestinationURI" ref="igrs:Type_URI"/>
</message>
<message name="ExportResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="DeleteResourceRequest">
        <part name="ResourceURI" ref="igrs:Type_URI"/>
</message>
<message name="DeleteResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="StopTransferResourceRequest">
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="StopTransferResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetTransferProgressInfoRequest">
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="GetTransferProgressInfoResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransferStatus" ref="igrs:Type_TransferState"/>
        <part name="TransferLength" ref="igrs:Type_Length"/>
        <part name="TransferTotal" ref="igrs:Type_Total"/>
</message>
<message name="SubscribeContentUpdateEventRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
</message>
<message name="SubscribeContentUpdateEventResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="UnsubscribeContentUpdateEventRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
</message>
<message name="UnsubscribeContentUpdateEventResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="SubscribeServiceAttributeRequest">
        <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
</message>
<message name="SubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnsubscribeServiceAttributeRequest">
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnsubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>

<portType name="ContentIndexPortType">
        <operation name="GetSearchCapabilityList">
```

```
        <input message="igrs:GetSearchCapabilityListRequest"/>
        <output message="igrs:GetSearchCapabilityListResponse"/>
</operation>
<operation name="GetSearchAttributeCapabilityList">
        <input message="igrs:GetSearchAttributeCapabilityListRequest"/>
        <output message="igrs:GetSearchAttributeCapabilityListResponse"/>
</operation>
<operation name="GetSortCapabilityList">
        <input message="igrs:GetSortCapabilityListRequest"/>
        <output message="igrs:GetSortCapabilityListResponse"/>
</operation>
<operation name="GetContentUpdateId">
        <input message="igrs:GetContentUpdateIdRequest"/>
        <output message="igrs:GetContentUpdateIdResponse"/>
</operation>
<operation name="Browse">
        <input message="igrs:BrowseRequest"/>
        <output message="igrs:BrowseResponse"/>
</operation>
<operation name="Search">
        <input message="igrs:SearchRequest"/>
        <output message="igrs:SearchResponse"/>
</operation>
<operation name="SearchAttributeValue">
        <input message="igrs:SearchAttributeValueRequest"/>
        <output message="igrs:SearchAttributeValueResponse"/>
</operation>
<operation name="ConvertMediaFormat">
        <input message="igrs:ConvertMediaFormatRequest"/>
        <output message="igrs:ConvertMediaFormatResponse"/>
</operation>
<operation name="CreateObject">
        <input message="igrs:CreateObjectRequest"/>
        <output message="igrs:CreateObjectResponse"/>
</operation>
<operation name="DestroyObject">
        <input message="igrs:DestroyObjectRequest"/>
        <output message="igrs:DestroyObjectResponse"/>
</operation>
<operation name="UpdateObject">
        <input message="igrs:UpdateObjectRequest"/>
        <output message="igrs:UpdateObjectResponse"/>
</operation>
<operation name="ImportResource">
        <input message="igrs:ImportResourceRequest"/>
        <output message="igrs:ImportResourceResponse"/>
</operation>
<operation name="ExportResource">
        <input message="igrs:ExportResourceRequest"/>
        <output message="igrs:ExportResourceResponse"/>
</operation>
<operation name="DeleteResource">
        <input message="igrs:DeleteResourceRequest"/>
        <output message="igrs:DeleteResourceResponse"/>
</operation>
<operation name="StopTransferResource">
        <input message="igrs:StopTransferResourceRequest"/>
        <output message="igrs:StopTransferResourceResponse"/>
</operation>
<operation name="GetTransferProcessInfo">
        <input message="igrs:GetTransferProcessRequest"/>
        <output message="igrs:GetTransferProcessResponse"/>
```

```
                </operation>
                <operation name="SubscribeContentUpdateEvent">
                        <input message="igrs:SubscribeContentUpdateEventRequest"/>
                        <output message="igrs:SubscribeContentUpdateEventResponse"/>
                </operation>
                <operation name="UnsubscribeContentUpdateEvent">
                        <input message="igrs:UnsubscribeContentUpdateEventRequest"/>
                        <output message="igrs:UnsubscribeContentUpdateEventResponse"/>
                </operation>
                <operation name="SubscribeServiceAttribute">
                        <input message="igrs:SubscribeServiceAttributeRequest"/>
                        <output message="igrs:SubscribeServiceAttributeResponse"/>
                </operation>
                <operation name="UnsubscribeServiceAttribute">
                        <input message="igrs:UnsubscribeServiceAttributeRequest"/>
                        <output message="igrs:UnsubscribeServiceAttributeResponse"/>
                </operation>

                <igrs:serviceAttribute      name="ContentUpdateId"      type="xsd:unsignedInt"
notifiable="true"/>
                <igrs:serviceAttribute      name="SortCaps"      type="Type_SortCapabilityList"
notifiable="true"/>
                <igrs:serviceAttribute           name="ObjectId"           type="Type_ObjectId"
notifiable="true"/>
                <igrs:serviceAttribute      name="TransferState"      type="Type_TransferState"
minOccurs="0" notifiable="true"/>

        </portType>
        <binding name="ContentIndexServiceIGRSPipe" type="igrs:ContentIndexPortType">
                <operation name="GetSearchCapabilityList">
                </operation>
                <operation name="GetSortCapabilityList">
                </operation>
                <operation name="GetContentUpdateId">
                </operation>
                <operation name="Browse">
                </operation>
                <operation name="Search">
                </operation>
                <operation name="SearchAttributeValue">
                </operation>
                <operation name="ConvertMediaFormat">
                </operation>
                <operation name="CreateObject">
                </operation>
                <operation name="DestroyObject">
                </operation>
                <operation name="UpdateObject">
                </operation>
                <operation name="ImportResource">
                </operation>
                <operation name="ExportResource">
                </operation>
                <operation name="DeleteResource">
                </operation>
                <operation name="StopTransferResource">
                </operation>
                <operation name="GetTransferProcessInfo">
                </operation>
                <operation name="SubscribeContentUpdateEvent">
                </operation>
                <operation name="UnsubscribeContentUpdateEvent">
```

```
            </operation>
            <operation name="SubscribeServiceAttribute">
            </operation>
            <operation name="UnsubscribeServiceAttribute">
            </operation>
        </binding>
        <service name="ContentIndexService">
            <port                                    name="ContentIndexService"
binding="igrs:ContentIndexServiceIGRSPipe">
            </port>
        </service>


        <types>
            <element name="Type_ProtocolInfoList" type="ProtocolInfoListType"/>
            <element name="Type_ProtocolInfo" type="ProtocolInfoType"/>
            <element name="Type_ConnectionManagementServiceId" type="xsd:string"/>
            <element name="Type_ConnectionId" type="xsd:unsignedInt"/>
            <element      name="Type_TransportInstanceId"      type="xsd:unsignedInt"
minOccurs="0"/>
            <element name="Type_ConnectionIdList" type="ConnectionIdListType"/>
            <element     name="Type_ConnectionState"     type="ConnectionStateType"
minOccurs="0"/>
            <element name="Type_IPList" type="IPListType" minOccurs="0"/>
            <element name="Type_MediaFormatList" type="MediaFormatListType"/>
            <element                                    name="Type_ServiceAttributeName"
type="ServiceAttributeNameType" minOccurs="0"/>
            <element name="Type_SubscriptionId" type="xsd:string" minOccurs="0"/>
            <element                        name="Type_RenderingManagementInstanceID"
type="xsd:unsignedInt"/>
            <element                                    name="Type_ConnectionRoleFlag"
type="ConnectionRoleFlagType"/>
            <complexType name="ProtocolInfoListType">
                <sequence>
                    <element  name="Type_ProtocolInfo"  type="ProtocolInfoType"
maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <complexType name="ProtocolInfoType">
                <sequence>
                    <element name="TransportProtocol">
                        <complexType>
                            <sequence>
                                <element  name="Port"  type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                            </sequence>
                            <attribute name="Name" type="xsd:string"/>
                        </complexType>
                    </element>
                    <element name="ControlProtocol" minOccurs="0">
                        <complexType>
                            <sequence>
                                <element  name="Port"  type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                            </sequence>
                            <attribute name="Name" type="xsd:string"/>
                        </complexType>
                    </element>
                    <element        name="ParameterList"        type="xsd:string"
minOccurs="0"/>
                    <element        name="AdditionalInfo"        type="xsd:string"
minOccurs="0"/>
```

```
                </sequence>
            </complexType>
            <complexType name="ConnectionIdListType">
                <sequence>
                    <element name="ConnectionId" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <complexType name="ConnectionStateType">
                <choice>
                    <element name="OK" type="xsd:string"/>
                    <element name="DISCONNECTED" type="xsd:string"/>
                    <element name="CONTENTFORMATMISMATCH"
type="xsd:string"/>
                    <element name="INSUFFICIENTBANDWIDTH"
type="xsd:string"/>
                    <element name="UNRELIABLECHANNEL" type="xsd:string"/>
                    <element name="UNKNOWN" type="xsd:string"/>
                </choice>
            </complexType>
            <complexType name="IPListType">
                <sequence>
                    <element name="IP" type="xsd:string"
maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <complexType name="MediaFormatListType">
                <sequence>
                    <element name="MediaFormat" type="MediaFormatType"
maxOccurs="unbounded"/>
                </sequence>
            </complexType>
            <complexType name="MediaFormatType">
                <sequence>
                    <element name="AudioFormat" type="xsd:string"
minOccurs="0"/>
                    <element name="VideoFormat" type="xsd:string"
minOccurs="0"/>
                    <element name="PhotoFormat" type="xsd:string"
minOccurs="0"/>
                    <element name="ReomteDesktop" type="RemoteDesktop"
minOccurs="0"/>
                    <element name="VideoCamera" type="VideoCamera"
minOccurs="0"/>
                </sequence>
                <attribute name="Name" type="xsd:string"/>
                <attribute name="Type" type="xsd:string"/>
            </complexType>
            <complexType name="ServiceAttributeNameType">
                <choice>
                    <element name="CONNECTIONSTATE" type="xsd:string"/>
                    <element name="CONNECTIONIDLIST" type="xsd:string"/>
                    <element name="PROTOCOLINFOLIST" type="xsd:string"/>
                    <element name="MEDIAFORMATLIST" type="xsd:string"/>
                    <element name="IPLIST" type="xsd:string"/>
                </choice>
            </complexType>
            <complexType name="ConnectionRoleFlagType">
                <choice>
                    <element name="ASSERVER" type="xsd:string"/>
                    <element name="ASCLIENT" type="xsd:string"/>
                </choice>
```

```
            </complexType>
        </types>

        <message name="GetProtocolInfoRequest"/>
        <message name="GetProtocolInfoResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ProtocolInfoList" type="Type_ProtocolInfoList"/>
            <part name="MediaFormatList" type="Type_MediaFormatList"/>
            <part name="IPList" type="Type_IPList"/>
        </message>
        <message name="PrepareForConnectionRequest">
            <part name="RemoteProtocolInfo" type="Type_ProtocolInfo"/>
            <part    name="PeerCMSId"    type="Type_ConnectionManagementServiceId"
miniOccurs="0"/>
            <part name="PeerConnectionId" type="Type_ConnectionId" miniOccurs="0"/>
            <part name="ConnectionRoleFlag" type="Type_ConnectionRoleFlag"/>
            <part name="PeerIPList" type="Type_IPList" minOccurs="0"/>
        </message>
        <message name="PrepareForConnectionResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ConnectionId" type="Type_ConnectionId"/>
            <part       name="TransportInstanceId"       type="Type_TransportInstanceId"
miniOccurs="0"/>
            <part       name="RmsId"       type="Type_RenderingManagementInstanceId"
minOccurs="0"/>
            <part name="UsableIPList" type="Type_IPList" miniOccurs="0"/>
        </message>
        <message name="ReleaseConnectionRequest">
            <part name="ConnectionId" type="Type_ConnectionId"/>
        </message>
        <message name="ReleaseConnectionResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="GetActiveConnectionIdListRequest"/>
        <message name="GetActiveConnectionIdlistResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ConnectionIdList" type="Type_ConnectionIdList"/>
        </message>
        <message name="GetCurrentConnectionInfoRequest">
            <part name="ConnectionId" type="Type_ ConnectionId"/>
        </message>
        <message name="GetCurrentConnectionInfoResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="TransportInstanceId" type="Type_TransportInstanceId"/>
            <part name="ProtocolInfo" type="Type_ ProtocolInfo"/>
            <part name="PeerCMSId" type="Type_ConnectionManagementServiceId"/>
            <part name="PeerConnectionId" type="Type_ConnectionId"/>
            <part name="ConnectionState" type="Type_ConnectionState"/>
            <part       name="RcsId"       type="Type_RenderingManagementInstanceId"
miniOccurs="0"/>
        </message>
        <message name="SubscribeServiceAttributeRequest">
            <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
        </message>
        <message name="SubscribeServiceAttributeResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="SubscriptionId" type="Type_SubscriptionId"/>
        </message>
        <message name="UnsubscribeServiceAttributeRequest">
            <part name="SubscriptionId" type="Type_SubscriptionId"/>
        </message>
        <message name="UnsubscribeServiceAttributeResponse">
```

```
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <portType name="ConnectionManagementPortType">
            <operation name="GetProtocolInfo">
                <input message="igrs:GetProtocolInfoRequest"/>
                <output message="igrs:GetProtocolInfoResponse"/>
            </operation>
            <operation name="PrepareForConnection">
                <input message="igrs:PrepareForConnectionRequest"/>
                <output message="igrs:PrepareForConnectionResponse"/>
            </operation>
            <operation name="ReleaseConnection">
                <input message="igrs:ReleaseConnectionRequest"/>
                <output message="igrs:ReleaseConnectionResponse"/>
            </operation>
            <operation name="GetActiveConnectionIdList">
                <input message="igrs:GetActiveConnectionIdListRequest"/>
                <output message="igrs:GetActiveConnectionIdListResponse"/>
            </operation>
            <operation name="GetCurrentConnectionInfo">
                <input message="igrs:GetCurrentConnectionInfoRequest"/>
                <output message="igrs:GetCurrentConnectionInfoResponse"/>
            </operation>
            <operation name="SubscribeServiceAttribute">
                <input message="igrs:SubscribeServiceAttributeRequest"/>
                <output message="igrs:SubscribeServiceAttributeResponse"/>
            </operation>
            <operation name="UnsubscribeServiceAttribute">
                <input message="igrs:UnsubscribeServiceAttributeRequest"/>
                <output message="igrs:UnsubscribeServiceAttributeResponse"/>
            </operation>

            <igrs:serviceAttribute  name="ConnectionState"  type="Type_ConnectionState"
minOccurs="0" notifiable="true"/>
            <igrs:serviceAttribute name="ConnectionIdList"  type="Type_ConnectionIdList"
notifiable="true"/>
            <igrs:serviceAttribute   name="ProtocolInfoList"   type="Type_ProtocolInfoList"
notifiable="true"/>
            <igrs:serviceAttribute  name="MediaFormatList"  type="Type_MediaFormatList"
notifiable="true"/>
            <igrs:serviceAttribute name="IPList" type="Type_IPList" notifiable="true"/>

        </portType>
        <binding                              name="ConnectionManagementServiceIGRSPipe"
type="igrs:ConnectionManagementPortType">
            <operation name="GetProtocolInfo">
            </operation>
            <operation name="PrepareForConnection">
            </operation>
            <operation name="ReleaseConnection">
            </operation>
            <operation name="GetActiveConnectionIdList">
            </operation>
            <operation name="GetCurrentConnectionInfo">
            </operation>
            <operation name="SubscribeServiceAttribute">
            </operation>
            <operation name="UnsubscribeServiceAttribute">
            </operation>
        </binding>
        <service name="ConnectionManagementService">
```

```
            <port                                    name="ConnectionManagementService"
binding="igrs:ConnectionManagementServiceIGRSPipe">
            </port>
        </service>


        <types>
            <element name="Type_TransportState" type="TransprotStateType"/>
            <element name="Type_StorageMediaName" type="StorageMediaNameType"/>
            <element name="Type_PlayMode" type="PlayModeType"/>
            <element name="Type_PlaySpeed" type="PlaySpeedType"/>
            <element name="Type_TrackNumber" type="xsd:unsignedInt"/>
            <element name="Type_MediaTimeLength" type="xsd:string"/>
            <element name="Type_TransportURIList" type="TransportURIListType"/>
            <element name="Type_TransportURI" type="xsd:string"/>
            <element name="Type_ControlURI" type="xsd:string" minOccurs="0"/>
            <element name="Type_Item" type="xsd:string" minOccurs="0"/>
            <element name="Type_ItemList" type="ItemListType" minOccurs="0"/>
            <element name="Type_SeekMode" type="SeekModeType"/>
            <element name="Type_SeekTargetPosition" type="xsd:string"/>
            <element name="Type_TransportInstanceId" type="xsd:unsignedInt"/>
            <element                                    name="Type_ServiceAttributeName"
type="ServiceAttributeNameType" minOccurs="0"/>
            <element name="Type_SubscriptionId" type="xsd:string" minOccurs="0"/>
            <element name="Type_Count" type="xsd:unsignedInt" minOccurs="0"/>
            <element name="Type_TimeLength" type="xsd:unsignedInt" minOccurs="0"/>
          <element       name="Type_TimeShiftSwitch"       type="TimeShiftSwitchType"
minOccurs="0"/>

    <complexType name="TimeShiftSwitchType">
                <choice>
                    <element name="OPEN" type="xsd:string"/>
                    <element name="CLOSE" type="xsd:string"/>
                </choice>
            </complexType>
            <complexType name="TransprotStateType">
                <choice>
                    <element name="PLAYING" type="xsd:string"/>
                    <element name="PAUSED_PLAYBACK" type="xsd:string"/>
                    <element name="PAUSED_RECORDING" type="xsd:string"/>
                    <element name="STOPPED" type="xsd:string"/>
                    <element name="RECORDING" type="xsd:string"/>
                    <element name="TRANSITIONING" type="xsd:string"/>
                    <element name="NO_MEDIA_PRESENT" type="xsd:string"/>
                    <element name="COMPLETE" type="xsd:string"/>
                    <element name="ERROR_OCCURRED" type="xsd:string"/>

                </choice>
            </complexType>
            <complexType name="StorageMediaNameType">
                <choice>
                    <element name="UNKNOWN" type="xsd:string"/>
                    <element name="DV" type="xsd:string"/>
                    <element name="MINI-DV" type="xsd:string"/>
                    <element name="VHS" type="xsd:string"/>
                    <element name="W-VHS" type="xsd:string"/>
                    <element name="S-VHS" type="xsd:string"/>
                    <element name="D-VHS" type="xsd:string"/>
                    <element name="VHSC" type="xsd:string"/>
                    <element name="VIDEO8" type="xsd:string"/>

                    <element name="HI8" type="xsd:string"/>
```

```
                        <element name="CD-ROM" type="xsd:string"/>
                        <element name="CD-DA" type="xsd:string"/>
                        <element name="CD-R" type="xsd:string"/>
                        <element name="CD-RW" type="xsd:string"/>
                        <element name="VIDEO-CD" type="xsd:string"/>
                        <element name="SACD" type="xsd:string"/>
                        <element name="MD-AUDIO" type="xsd:string"/>
                        <element name="MD-PICTURE" type="xsd:string"/>
                        <element name="DVD-ROM" type="xsd:string"/>
                        <element name="DVD-VIDEO" type="xsd:string"/>
                        <element name="DVD-R" type="xsd:string"/>
                        <element name="DVD+RW" type="xsd:string"/>
                        <element name="DVD-RW" type="xsd:string"/>
                    </choice>
                </complexType>
                <complexType name="PlayModeType">
                    <choice>
                        <element name="NORMAL" type="xsd:string"/>
                        <element name="SHUFFLE" type="xsd:string"/>
                        <element name="REPEAT_ONE" type="xsd:string"/>
                        <element name="REPEAT_ALL" type="xsd:string"/>
                        <element name="RANDOM" type="xsd:string"/>
                        <element name="DIRECT_1" type="xsd:string"/>
                        <element name="INTRO" type="xsd:string"/>
                    </choice>
                </complexType>
                <complexType name="PlaySpeedType">
                    <choice>
                        <element name="NORMAL" type="xsd:string"/>
                        <element name="FASTFORWARD" type="xsd:string"/>
                        <element name="SLOWFORWARD" type="xsd:string"/>
                        <element name="FASTBACKWARD" type="xsd:string"/>
                    </choice>
                </complexType>
                <complextype name="TransportURIListType">
                   <sequence>
                     <element name="TransportURI" type="xsd:string" minOccurs="1"
maxOccurs="unbounded"/>
                   </sequence>
                </complextype>
                <complextype name="ItemListType">
                   <sequence>
                     <element name="Item" type="ItemType" minOccurs="1"/>
                   </sequence>
                </complextype>
                <complexType name="ItemType">
                    <sequence>
                        <element name="ItemAttribute" type="ItemAttributeType"/>
                        <element name="ItemReferences" minOccurs="0">
                            <complexType>
                                <sequence>
                                    <element name="KeyFrame"
minOccurs="0">
                                        <attribute name="Id"
type="igrs:Type_ObjectId"/>
                                    </element>
                                    <element name="MicroPhotograph"
minOccurs="0">
                                        <attribute name="Id"
type="igrs:Type_ObjectId"/>
                                    </element>
                                </sequence>
```

```
                        </complexType>
                    </element>
                </sequence>
            </complexType>
            <complexType name="ItemAttributeType">
                <sequence>
                    <element name="ObjectId" type="xsd:string"/>
                    <element name="ParentId" type="xsd:string" minOccurs="0"/>
                    <element name="DeviceId" type="xsd:string" minOccurs="0"/>
                    <element name="DeviceName" type="xsd:string"
minOccurs="0"/>
                    <element name="ObjectPath" type="xsd:string"
minOccurs="0"/>
                    <element name="ObjectName" type="xsd:string"
minOccurs="0"/>
                    <element name="ObjectStoreAttribute">
                        <complexType>
                            <sequence>
                                <element name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                                <element name="Hide" type="xsd:string"
minOccurs="0"/>
                                <element name="Document"
type="xsd:string" minOccurs="0"/>
                                <element name="System"
type="xsd:string" minOccurs="0"/>
                            </sequence>
                        </complexType>
                    </element>
                    <element name="CreateTime" type="xsd:string"
minOccurs="0"/>
                    <element name="LastAccessTime" type="xsd:string"
minOccurs="0"/>
                    <element name="LastWriteTime" type="xsd:string"
minOccurs="0"/>
                    <element name="ObjectType" type="igrs:ObjectTypeType"
minOccurs="0"/>
                    <element name="Size" type="xsd:string" minOccurs="0"/>
                    <element name="Duration" type="xsd:string" minOccurs="0"/>
                    <element name="Encrypted" type="xsd:string" minOccurs="0"/>
                    <element name="DRMContendId" type="xsd:string" minOccurs="0"/>
                    <element name="Type_MediaFormat">
                        <complexType>
                            <sequence>
                                <element name="AudioFormat"
type="xsd:string" minOccurs="0"/>
                                <element name="VideoFormat"
type="xsd:string" minOccurs="0"/>
                                <element name="PhotoFormat"
type="xsd:string" minOccurs="0"/>
                                <element name="RemoteDesktop"
type="RemoteDesktop" minOccurs="0"/>
                                <element name="VideoCamera"
type="VideoCamera" minOccurs="0"/>
                            </sequence>
                            <attribute name="Name" type="xsd:string"/>
                            <attribute name="Type" type="xsd:string"/>
                        </complexType>
                    </element>
                    <element name="ObjectURI" type="xsd:string"/>
                    <element name="Type_ConnectionMangementServiceId"
type="xsd:unsignedInt"/>
```

```
                        <element name="Choices" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element  name="Choice"  minOccurs="0"
maxOccurs="unbounded">
                                                        <complexType>
                                                                <sequence>
                                                                        <element
name="Selection" minOccurs="0" maxOccurs="unbounded">
        <complexType>
        <attribute name="Id" type="xsd:string"/>
        <attribute name="Name" type="xsd:string"/>
        </complexType>
                                                                        </element>
                                                                </sequence>
                                                                <attribute         name="Id"
type="xsd:string"/>
                                                                <attribute    name="Name"
type="xsd:string"/>
                                                                <attribute   name="Default"
type="xsd:string"/>
                                                                <attribute      name="Type"
type="xsd:string"/>
                                                        </complexType>
                                                </element>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ObjectTypeType">
                <choice>
                        <element           name="Audio"           type="igrs:AudioType"
minOccurs="0"/>
                        <element name="Video" type="xsd:string" minOccurs="0"/>
                        <element name="Photo" type="xsd:string" minOccurs="0"/>
                        <element name="Doc" type="xsd:string" minOccurs="0"/>
                        <element name="Camera" type="xsd:string" minOccurs="0"/>
                        <element name="Screen" type="xsd:string" minOccurs="0"/>
                </choice>
        </complexType>
        <complexType name="AudioType">
                <sequence>
                        <element name="Singer" type="xsd:string" minOccurs="0"/>
                        <element name="Genre" type="xsd:string" minOccurs="0"/>
                        <element name="MusicDisc" type="xsd:string" minOccurs="0"/>
                        <element name="Author" type="xsd:string" minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="SeekModeType">
                <choice>
                        <element name="TRACK_NR" type="xsd:string"/>
                        <element name="TAPE_INDEX" type="xsd:string"/>
                        <element name="ABS_COUNT" type="xsd:string"/>
                        <element name="REL_COUNT" type="xsd:string"/>
                        <element name="ABS_TIME" type="xsd:string"/>
                        <element name="REL_TIME" type="xsd:string"/>
                        <element name="FRAME" type="xsd:string"/>
```

```
                </choice>
        </complexType>
        <complexType name="ServiceAttributeNameType">
                <choice>
                        <element name="TRANSPORTSTATE" type="xsd:string"/>
                        <element name="CURRENTSPEED" type="xsd:string"/>
                        <element name="TRANSPROTURILIST" type="xsd:string"/>
                        <element name="PLAYMODE" type="xsd:string"/>
                </choice>
        </complexType>
</types>

<message name="SetTransportURIListRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="TransportURIList" type="igrs:Type_TransportURIList"/>
        <part name="ItemList" type="Type_ItemList"/>
</message>
<message name="SetTransportURIListResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetTransportInfoRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetTransportInfoResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentTransportState" type="Type_TransportState"/>
        <part name="CurrentSpeed" type="Type_PlaySpeed"/>
</message>
<message name="PlayRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="Speed" type="Type_PlaySpeed"/>
        <part name="Offset" type="Type_Count"/>
</message>
<message name="PlayResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="NextRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="NextResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="PreviousRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="PreviousResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="StopRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="StopResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="PauseRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="PauseResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="ResumeRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
```

```
</message>
<message name="ResumeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="SeekRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="Unit" type="Type_SeekMode"/>
        <part name="Target" type="Type_SeekTargetPosition"/>
</message>
<message name="SeekResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="TimeShiftRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="ShiftTime" type="Type_TimeLength"/>
        <part name="TimeShiftSwitch" type="Type_TimeShiftSwitch"/>
</message>
<message name="TimeShiftResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetPlayURIListRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetPlayURIListResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransportURIList" type="Type_TransportURIList"/>
</message>
<message name="GetAllMediaInfoRequest">
    <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetAllMediaInfoResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="ItemList" type="Type_ItemList"/>
</message>
<message name="GetCurrentMediaInfoRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetCurrentMediaInfoResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="ElapsedDuration" type="Length"/>
        <part name="CurrentURI" type="TransportURI"/>
        <part name="Item" type="Type_Item"/>
</message>
<message name="GetCurrentPalyModeRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetCurrentPalyModeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="PlayMode" type="Type_PlayMode"/>
</message>
<message name="SetPlayModeRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="NewPlayMode" type="Type_PlayMode"/>
</message>
<message name="SetPlayModeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="SetChoiceSelectionRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="ChoiceType" type="xsd:string"/>
        <part name="SelectionId" type="xsd:string"/>
</message>
```

```
<message name="SetChoiceSelectionResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="SubscribeServiceAttributeRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
</message>
<message name="SubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnsubscribeServiceAttributeRequest">
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnsubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<portType name="MediaServerTransportPortType">
        <operation name="SetTransportURIList">
                <input message="igrs:SetTransportURIListRequest"/>
                <output message="igrs:SetTransportURIListResponse"/>
        </operation>
        <operation name="GetTransportInfo">
                <input message="igrs:GetTransportInfoRequest"/>
                <output message="igrs:GetTransportInfoResponse"/>
        </operation>
        <operation name="Play">
                <input message="igrs:PlayRequest"/>
                <output message="igrs:PlayResponse"/>
        </operation>
        <operation name="Next">
                <input message="igrs:NextRequest"/>
                <output message="igrs:NextResponse"/>
        </operation>
        <operation name="Previous">
                <input message="igrs:PreviousRequest"/>
                <output message="igrs:PreviousResponse"/>
        </operation>
        <operation name="Stop">
                <input message="igrs:StopRequest"/>
                <output message="igrs:StopResponse"/>
        </operation>
        <operation name="Pause">
                <input message="igrs:PauseRequest"/>
                <output message="igrs:PauseResponse"/>
        </operation>
        <operation name="Resume">
                <input message="igrs:ResumeRequest"/>
                <output message="igrs:ResumeResponse"/>
        </operation>
        <operation name="Seek">
                <input message="igrs:SeekRequest"/>
                <output message="igrs:SeekResponse"/>
        </operation>
        <operation name="TimeShift">
                <input message="igrs:TimeShiftRequest"/>
                <output message="igrs:TimeShiftResponse"/>
        </operation>
        <operation name="GetPlayURIList">
                <input message="igrs:GetPlayURIListRequest"/>
                <output message="igrs:GetPlayURIListResponse"/>
        </operation>
```

```
                <operation name="GetAllMediaInfo">
                        <input message="igrs:GetAllMediaInfoRequest"/>
                        <output message="igrs:GetAllMediaInfoResponse"/>
                </operation>
                <operation name="GetCurrentMediaInfo">
                        <input message="igrs:GetCurrentMediaInfoRequest"/>
                        <output message="igrs:GetCurrentMediaInfoResponse"/>
                </operation>
                <operation name="GetPlayMode">
                        <input message="igrs:GetPalyModeRequest"/>
                        <output message="igrs:GetPalyModeResponse"/>
                </operation>
                <operation name="SetPlayMode">
                        <input message="igrs:SetPlayModeRequest"/>
                        <output message="igrs:SetPlayModeResponse"/>
                </operation>
                <operation name="SetChoiceSelection">
                        <input message="igrs:SetChoiceSelectionRequest"/>
                        <output message="igrs:SetChoiceSelectionResponse"/>
                </operation>
                <operation name="SubscribeServiceAttribute">
                        <input message="igrs:SubscribeServiceAttributeRequest"/>
                        <output message="igrs:SubscribeServiceAttributeResponse"/>
                </operation>
                <operation name="UnsubscribeServiceAttribute">
                        <input message="igrs:UnsubscribeServiceAttributeRequest"/>
                        <output message="igrs:UnsubscribeServiceAttributeResponse"/>
                </operation>

                <igrs:serviceAttribute                          name="CurrentTransportState"
type="Type_CurrentTransportState" notifiable="true"/>
                <igrs:serviceAttribute     name="CurrentSpeed"     type="Type_CurrentSpeed"
notifiable="true"/>
                <igrs:serviceAttribute name="TransportUIRList"  type="Type_TransportUIRList"
notifiable="true"/>
                <igrs:serviceAttribute          name="PlayMode"          type="Type_PlayMode"
notifiable="true"/>

        </portType>
        <binding                  name="MediaServerTransportManagementServiceIGRSPipe"
type="igrs:MediaServerTransportManagementPortType">
                <operation name="SetTransportURIList">
                </operation>
                <operation name="GetTransportInfo">
                </operation>
                <operation name="Play">
                </operation>
                <operation name="Pause">
                </operation>
                <operation name="Resume">
                </operation>
                <operation name="Stop">
                </operation>
                <operation name="Seek">
                </operation>
                <operation name="TimeShift">
                </operation>
                <operation name="GetAllMediaInfo">
                </operation>
                <operation name="GetCurrentMediaInfo">
                </operation>
                <operation name="GetPalyMode">
```

```
                </operation>
                <operation name="SetPlayMode">
                </operation>
                <operation name="SetChoiceSelection">
                </operation>
                <operation name="SubscribeServiceAttribute">
                </operation>
                <operation name="UnsubscribeServiceAttribute">
                </operation>
        </binding>
        <service name="MediaServerTransportManagementService">
                <port                    name="MediaServerTransportManagementService"
binding="igrs:MediaServerTransportManagementServiceIGRSPipe">
                </port>
        </service>

        <types>
                <element name="Type_TransportState" type="TransportStateType"/>
                <element name="Type_StorageMediaName" type="StorageMediaNameType"/>
                <element name="Type_PlayMode" type="PlayModeType"/>
                <element name="Type_PlaySpeed" type="PlaySpeedType"/>
                <element name="Type_TrackNumber" type="xsd:unsignedInt"/>
                <element name="Type_MediaTimeLength" type="xsd:string"/>
                <element name="Type_TransportURIList" type="TransportURIListType"/>
                <element name="Type_TransportURI" type="xsd:string"/>
                <element name="Type_ControlURI" type="xsd:string" minOccurs="0"/>
                <element name="Type_Item" type="xsd:string" minOccurs="0"/>
                <element name="Type_ItemList" type="ItemListType" minOccurs="0"/>
                <element name="Type_SeekMode" type="SeekModeType"/>
                <element name="Type_SeekTargetPosition" type="xsd:string"/>
                <element name="Type_TransportInstanceId" type="xsd:unsignedInt"/>
                <element                                 name="Type_ServiceAttributeName"
type="ServiceAttributeNameType" minOccurs="0"/>
                <element name="Type_SubscriptionId" type="xsd:string" minOccurs="0"/>
                <element name="Type_Count" type="xsd:unsignedInt" minOccurs="0"/>
        <element name="Type_RecordInput" type="RecordInputType"/>
        <element name="Type_RecordBitRateType" type="RecordBitRateTypeType"/>
        <element name="Type_RecordBitRate" type="xsd:unsignedInt" minOccurs="0"/>
        <element          name="Type_RecordAudioSampleRate"          type="xsd:unsignedInt"
minOccurs="0"/>
        <element name="Type_RecordAudioBitRate" type="xsd:unsignedInt" minOccurs="0"/>
        <element name="Type_RecordVideoFormat" type="xsd:string" minOccurs="0"/>
        <element name="Type_TimeLength" type="xsd:unsignedInt" minOccurs="0"/>
        <element name="Type_TimeShiftSwitch" type="TimeShiftSwitchType" minOccurs="0"/>

        <complexType name="TimeShiftSwitchType">
                        <choice>
                                <element name="OPEN" type="xsd:string"/>
                                <element name="CLOSE" type="xsd:string"/>
                        </choice>
                </complexType>
        <complexType name="RecordInputType">
                        <choice>
                                <element name="DIGITAL" type="xsd:string"/>
                                <element name="ANALOG" type="xsd:string"/>
                        </choice>
                </complexType>
        <complexType name="RecordBitRateTypeType">
                        <choice>
                                <element name="FIXED" type="xsd:string"/>
                                <element name="VARIABLE" type="xsd:string"/>
                        </choice>
```

```
</complexType>
<complexType name="TransprotStateType">
	<choice>
		<element name="PLAYING" type="xsd:string"/>
		<element name="PAUSED_PLAYBACK" type="xsd:string"/>
		<element name="PAUSED_RECORDING" type="xsd:string"/>
		<element name="STOPPED" type="xsd:string"/>
		<element name="RECORDING" type="xsd:string"/>
		<element name="TRANSITIONING" type="xsd:string"/>
		<element name="NO_MEDIA_PRESENT" type="xsd:string"/>
		<element name="COMPLETE" type="xsd:string"/>
		<element name="ERROR_OCCURRED" type="xsd:string"/>

	</choice>
</complexType>
<complexType name="StorageMediaNameType">
	<choice>
		<element name="UNKNOWN" type="xsd:string"/>
		<element name="DV" type="xsd:string"/>
		<element name="MINI-DV" type="xsd:string"/>
		<element name="VHS" type="xsd:string"/>
		<element name="W-VHS" type="xsd:string"/>
		<element name="S-VHS" type="xsd:string"/>
		<element name="D-VHS" type="xsd:string"/>
		<element name="VHSC" type="xsd:string"/>
		<element name="VIDEO8" type="xsd:string"/>

		<element name="HI8" type="xsd:string"/>
		<element name="CD-ROM" type="xsd:string"/>
		<element name="CD-DA" type="xsd:string"/>
		<element name="CD-R" type="xsd:string"/>
		<element name="CD-RW" type="xsd:string"/>
		<element name="VIDEO-CD" type="xsd:string"/>
		<element name="SACD" type="xsd:string"/>
		<element name="MD-AUDIO" type="xsd:string"/>
		<element name="MD-PICTURE" type="xsd:string"/>
		<element name="DVD-ROM" type="xsd:string"/>
		<element name="DVD-VIDEO" type="xsd:string"/>
		<element name="DVD-R" type="xsd:string"/>
		<element name="DVD+RW" type="xsd:string"/>
		<element name="DVD-RW" type="xsd:string"/>
	</choice>
</complexType>
<complexType name="PlayModeType">
	<choice>
		<element name="NORMAL" type="xsd:string"/>
		<element name="SHUFFLE" type="xsd:string"/>
		<element name="REPEAT_ONE" type="xsd:string"/>
		<element name="REPEAT_ALL" type="xsd:string"/>
		<element name="RANDOM" type="xsd:string"/>
		<element name="DIRECT_1" type="xsd:string"/>
		<element name="INTRO" type="xsd:string"/>
	</choice>
</complexType>
<complexType name="PlaySpeedType">
	<choice>
		<element name="NORMAL" type="xsd:string"/>
		<element name="FASTFORWARD" type="xsd:string"/>
		<element name="SLOWFORWARD" type="xsd:string"/>
		<element name="FASTBACKWARD" type="xsd:string"/>
	</choice>
</complexType>
```

```
<complextype name="TransportURIListType">
  <sequence>
    <element name="TransportURI" type="string" minOccurs="1"
maxOccurs="unbounded"/>
  </sequence>
</complextype>
<complextype name="ItemListType">
  <sequence>
    <element name="Item" type="ItemType" minOccurs="1"/>
  </sequence>
</complextype>
<complexType name="ItemType">
    <sequence>
        <element name="ItemAttribute" type="ItemAttributeType"/>
        <element name="ItemReferences" minOccurs="0">
            <complexType>
                <sequence>
                    <element name="KeyFrame"
minOccurs="0">
                        <attribute name="Id"
type="igrs:Type_ObjectId"/>
                    </element>
                    <element name="MicroPhotograph"
minOccurs="0">
                        <attribute name="Id"
type="igrs:Type_ObjectId"/>
                    </element>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>
<complexType name="ItemAttributeType">
    <sequence>
        <element name="ObjectId" type="xsd:string"/>
        <element name="ParentId" type="xsd:string" minOccurs="0"/>
        <element name="DeviceId" type="xsd:string" minOccurs="0"/>
        <element name="DeviceName" type="xsd:string"
minOccurs="0"/>
        <element name="ObjectPath" type="xsd:string"
minOccurs="0"/>
        <element name="ObjectName" type="xsd:string"
minOccurs="0"/>
        <element name="ObjectStoreAttribute">
            <complexType>
                <sequence>
                    <element name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                    <element name="Hide" type="xsd:string"
minOccurs="0"/>
                    <element name="Document"
type="xsd:string" minOccurs="0"/>
                    <element name="System"
type="xsd:string" minOccurs="0"/>
                </sequence>
            </complexType>
        </element>
        <element name="CreateTime" type="xsd:string"
minOccurs="0"/>
        <element name="LastAccessTime" type="xsd:string"
minOccurs="0"/>
```

```
                        <element name="LastWriteTime" type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectType" type="igrs:ObjectTypeType"
minOccurs="0"/>
                        <element name="Size" type="xsd:string" minOccurs="0"/>
                        <element name="Duration" type="xsd:string" minOccurs="0"/>
                        <element name="Encrypted" type="xsd:string" minOccurs="0"/>
                    <element name="DRMContendId" type="xsd:string" minOccurs="0"/>
                        <element name="Type_MediaFormat">
                            <complexType>
                                <sequence>
                                    <element name="AudioFormat"
type="xsd:string" minOccurs="0"/>
                                    <element name="VideoFormat"
type="xsd:string" minOccurs="0"/>
                                    <element name="PhotoFormat"
type="xsd:string" minOccurs="0"/>
                                    <element name="RemoteDesktop"
type="RemoteDesktop" minOccurs="0"/>
                                    <element name="VideoCamera"
type="VideoCamera" minOccurs="0"/>
                                </sequence>
                                <attribute name="Name" type="xsd:string"/>
                                <attribute name="Type" type="xsd:string"/>
                            </complexType>
                        </element>
                        <element name="ObjectURI" type="xsd:string"/>
                        <element name="Type_ConnectionMangementServiceId"
type="xsd:unsignedInt"/>
                        <element name="Choices" minOccurs="0">
                            <complexType>
                                <sequence>
                                    <element name="Choice" minOccurs="0"
maxOccurs="unbounded">
                                        <complexType>
                                            <sequence>
                                                <element
name="Selection" minOccurs="0" maxOccurs="unbounded">
        <complexType>
        <attribute name="Id" type="string"/>
        <attribute name="Name" type="string"/>
        </complexType>
                                                </element>
                                            </sequence>
                                            <attribute name="Id"
type="string"/>
                                            <attribute name="Name"
type="string"/>
                                            <attribute name="Default"
type="string"/>
                                            <attribute name="Type"
type="string"/>
                                        </complexType>
                                    </element>
                                </sequence>
                            </complexType>
                        </element>
                    </sequence>
```

```
            </complexType>
            <complexType name="ObjectTypeType">
                  <choice>
                        <element name="Audio" type="igrs:AudioType"
minOccurs="0"/>
                        <element name="Video" type="xsd:string" minOccurs="0"/>
                        <element name="Photo" type="xsd:string" minOccurs="0"/>
                        <element name="Doc" type="xsd:string" minOccurs="0"/>
                        <element name="Camera" type="xsd:string" minOccurs="0"/>
                        <element name="Screen" type="xsd:string" minOccurs="0"/>
                  </choice>
            </complexType>
            <complexType name="AudioType">
                  <sequence>
                        <element name="Singer" type="xsd:string" minOccurs="0"/>
                        <element name="Genre" type="xsd:string" minOccurs="0"/>
                        <element name="MusicDisc" type="xsd:string" minOccurs="0"/>
                        <element name="Author" type="xsd:string" minOccurs="0"/>
                  </sequence>
            </complexType>
            <complexType name="SeekModeType">
                  <choice>
                        <element name="TRACK_NR" type="xsd:string"/>
                        <element name="TAPE_INDEX" type="xsd:string"/>
                        <element name="ABS_COUNT" type="xsd:string"/>
                        <element name="REL_COUNT" type="xsd:string"/>
                        <element name="ABS_TIME" type="xsd:string"/>
                        <element name="REL_TIME" type="xsd:string"/>
                        <element name="FRAME" type="xsd:string"/>
                  </choice>
            </complexType>
            <complexType name="ServiceAttributeNameType">
                  <choice>
                        <element name="TRANSPORTSTATE" type="xsd:string"/>
                        <element name="CURRENTSPEED" type="xsd:string"/>
                        <element name="TRANSPROTURILIST" type="xsd:string"/>
                        <element name="PLAYMODE" type="xsd:string"/>
                  </choice>
            </complexType>
      </types>

      <message name="SetTransportURIListRequest">
            <part name="InstanceId" type="Type_TransportInstanceId"/>
            <part name="TransportURIList" type="Type_TransportURIList"/>
            <part name="ItemList" type="Type_ItemList"/>
      </message>
      <message name="SetTransportURIListResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
      </message>
      <message name="GetTransportInfoRequest">
            <part name="InstanceId" type="Type_TransportInstanceId"/>
      </message>
      <message name="GetTransportInfoResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="CurrentTransportState" type="Type_TransportState"/>
            <part name="CurrentSpeed" type="Type_PlaySpeed"/>
      </message>
      <message name="PlayRequest">
            <part name="InstanceId" type="Type_TransportInstanceId"/>
            <part name="Speed" type="Type_PlaySpeed"/>
            <part name="Offset" type="Type_Count"/>
      </message>
```

```
<message name="PlayResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="NextRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="NextResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="PreviousRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="PreviousResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="StopRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="StopResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="PauseRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="PauseResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="ResumeRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="ResumeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="SeekRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="Unit" type="Type_SeekMode"/>
        <part name="Target" type="Type_SeekTargetPosition"/>
</message>
<message name="SeekResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="TimeShiftRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="ShiftTime" type="Type_TimeLength"/>
        <part name="TimeShiftSwitch" type="Type_TimeShiftSwitch"/>
</message>
<message name="TimeShiftResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetPlayURIListRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetPlayURIListResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransportURIList" type="igrs:TransportURIList"/>
</message>
<message name="GetAllMediaInfoRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
</message>
<message name="GetAllMediaInfoResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="ItemList" type="Type_ItemList"/>
```

```
        </message>
        <message name="GetCurrentMediaInfoRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
        </message>
        <message name="GetCurrentMediaInfoResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="ElapsedDuration" type="Type_Length"/>
                <part name="CurrentURI" type="Type_TransportURI"/>
                <part name="Item" type="Type_ContentList"/>
        </message>
        <message name="GetCurrentPlayModeRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
        </message>
        <message name="GetCurrentPlayModeResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="PlayMode" type="Type_PlayMode"/>
        </message>
        <message name="SetPlayModeRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
                <part name="NewPlayMode" type="Type_PlayMode"/>
        </message>
        <message name="SetPlayModeResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="RecordRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
                <part name="RecordInput" type="Type_RecordInput"/>
                <part name="RecordBitRateType" type="Type_RecordBitRateType"/>
                <part name="RecordBitRate" type="xsd:unsignedInt" miniOccur="0"/>
                <part name="RecordVbrArg" type="xsd:unsignedInt" miniOccur="0"/>
                <part name="RecordAudioSampleRate" type="xsd:unsignedInt"
miniOccur="0"/>
                <part name="RecordAudioBitRate" type="xsd:unsignedInt" miniOccur="0"/>
                <part name="RecordVedioFormat" type="xsd:string" miniOccur="0"/>
        </message>
        <message name="RecordResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="PauseRecordRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
        </message>
        <message name="PauseRecordResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="ResumeRecordRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
        </message>
        <message name="ResumeRecordResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="StopRecordRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
        </message>
        <message name="StopRecordResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="SetChoiceSelectionRequest">
                <part name="InstanceId" type="Type_TransportInstanceId"/>
                <part name="ChoiceType" ref="xsd:string"/>
                <part name="SelectionId" ref="xsd:string"/>
        </message>
        <message name="SetChoiceSelectionResponse">
```

```
        <part name="ReturnCode" type="xsd:unsignedInt"/>
    </message>
    <message name="SubscribeServiceAttributeRequest">
        <part name="InstanceId" type="Type_TransportInstanceId"/>
        <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
    </message>
    <message name="SubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
    </message>
    <message name="UnsubscribeServiceAttributeRequest">
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
    </message>
    <message name="UnsubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
    </message>

    <portType name="MediaRendererTransportManagementPortType">
        <operation name="SetTransportURIList">
            <input message="igrs:SetTransportURIListRequest"/>
            <output message="igrs:SetTransportURIRListesponse"/>
        </operation>
        <operation name="GetTransportInfo">
            <input message="igrs:GetTransportInfoRequest"/>
            <output message="igrs:GetTransportInfoResponse"/>
        </operation>
        <operation name="Play">
            <input message="igrs:PlayRequest"/>
            <output message="igrs:PlayResponse"/>
        </operation>
        <operation name="Next">
            <input message="igrs:NextRequest"/>
            <output message="igrs:NextResponse"/>
        </operation>
        <operation name="Previous">
            <input message="igrs:PreviousRequest"/>
            <output message="igrs:PreviousResponse"/>
        </operation>
        <operation name="Stop">
            <input message="igrs:StopRequest"/>
            <output message="igrs:StopResponse"/>
        </operation>
        <operation name="Pause">
            <input message="igrs:PauseRequest"/>
            <output message="igrs:PauseResponse"/>
        </operation>
        <operation name="Resume">
            <input message="igrs:ResumeRequest"/>
            <output message="igrs:ResumeResponse"/>
        </operation>
        <operation name="Seek">
            <input message="igrs:SeekRequest"/>
            <output message="igrs:SeekResponse"/>
        </operation>
        <operation name="TimeShift">
            <input message="igrs:TimeShiftRequest"/>
            <output message="igrs:TimeShiftResponse"/>
        </operation>
        <operation name="GetPlayURIList">
            <input message="igrs:GetPlayURIListRequest"/>
            <output message="igrs:GetPlayURIListResponse"/>
        </operation>
```

```
		<operation name="GetAllMediaInfo">
			<input message="igrs:GetAllMediaInfoRequest"/>
			<output message="igrs:GetAllMediaInfoResponse"/>
		</operation>
		<operation name="GetCurrentMediaInfo">
			<input message="igrs:GetCurrentMediaInfoRequest"/>
			<output message="igrs:GetCurrentMediaInfoResponse"/>
		</operation>
		<operation name="GetCurrentPlayMode">
			<input message="igrs:GetPlayModeRequest"/>
			<output message="igrs:GetPlayModeResponse"/>
		</operation>
		<operation name="SetPlayMode">
			<input message="igrs:SetPlayModeRequest"/>
			<output message="igrs:SetPlayModeResponse"/>
		</operation>
		<operation name="Record">
			<input message="igrs:RecordRequest"/>
			<output message="igrs:RecordResponse"/>
		</operation>
		<operation name="PauseRecord">
			<input message="igrs:PauseRecordRequest"/>
			<output message="igrs:PauseRecordResponse"/>
		</operation>
		<operation name="ResumeRecord">
			<input message="igrs:ResumeRecordRequest"/>
			<output message="igrs:ResumeRecordResponse"/>
		</operation>
		<operation name="StopRecord">
			<input message="igrs:StopRecordRequest"/>
			<output message="igrs:StopRecordResponse"/>
		</operation>
		<operation name="SetChoiceSelection">
			<input message="igrs:SetChoiceSelectionRequest"/>
			<output message="igrs:SetChoiceSelectionResponse"/>
		</operation>
		<operation name="SubscribeServiceAttribute">
			<input message="igrs:SubscribeServiceDataRequest"/>
			<output message="igrs:SubscribeServiceDataResponse"/>
		</operation>
		<operation name="UnsubscribeServiceAttribute">
			<input message="igrs:UnsubscribeServiceDataRequest"/>
			<output message="igrs:UnsubscribeServiceDataResponse"/>
		</operation>

		<igrs:serviceAttribute name="CurrentTransportState"
type="Type_CurrentTransportState" notifiable="true"/>
		<igrs:serviceAttribute name="CurrentSpeed" type="Type_CurrentSpeed"
notifiable="true"/>
		<igrs:serviceAttribute name="TransportUIRList" type="Type_TransportUIRList"
notifiable="true"/>
		<igrs:serviceAttribute name="PlayMode" type="Type_PlayMode"
notifiable="true"/>

	</portType>
	<binding name="MediaRendererTransportManagementServiceIGRSPipe"
type="igrs:MediaRendererTransportManagementPortType">
		<operation name="SetTransportURIList">
		</operation>
		<operation name="GetTransportInfo">
		</operation>
		<operation name="Play">
```

```
                </operation>
                <operation name="Next">
                </operation>
                <operation name="Previous">
                </operation>
                <operation name="Pause">
                </operation>
                <operation name="Resume">
                </operation>
                <operation name="Stop">
                </operation>
                <operation name="Seek">
                </operation>
                <operation name="TimeShift">
                </operation>
                <operation name="GetPlayURIList">
                </operation>
                <operation name="GetAllMediaInfo">
                </operation>
                <operation name="GetCurrentMediaInfo">
                </operation>
                <operation name="GetCurrentPlayMode">
                </operation>
                <operation name="SetPlayMode">
                </operation>
                <operation name="Record">
                </operation>
                <operation name="SetChoiceSelection">
                </operation>
                <operation name="SubscribeServiceData">
                </operation>
                <operation name="UnsubscribeServiceData">
                </operation>
        </binding>
        <service name="MediaRendererTransportManagementService">
                <port                    name="MediaRendererTransportManagementService"
binding="igrs:MediaRendererTransportManagementServiceIGRSPipe">
                </port>
        </service>

        <types>
                <element name="Type_PresetProfileList" type="xsd:string"/>
                <element name="Type_Brightness" type="xsd:unsignedInt" minOccurs="0"/>
                <element name="Type_Contrast" type="xsd:unsignedInt" minOccurs="0"/>
                <element name="Type_Sharpness" type="xsd:unsignedInt" minOccurs="0"/>
                <element name="Type_VideoGain" type="xsd:unsignedInt" minOccurs="0"/>
                <element          name="Type_VideoGreyLevel"          type="xsd:unsignedInt"
minOccurs="0"/>
                <element        name="Type_ColorTemperature"        type="xsd:unsignedInt"
minOccurs="0"/>
                <element name="Type_MuteState" type="xsd:boolean" minOccurs="0"/>
                <element name="Type_Volume" type="xsd:unsignedInt" minOccurs="0"/>
                <element name="Type_LoudnessSwitch" type="xsd:boolean" minOccurs="0"/>
                <element name="Type_PlayChannel" type="PlayChannelType"/>
                <element                          name="Type_RenderingManagementInstanceId"
type="xsd:unsignedInt"/>
                <element name="Type_PresetProfile" type="xsd:string"/>
                <element name="Type_DisplayWindowId" type="xsd:unsignedInt"/>
                <element name="Type_DisplayWindowSize" type="xsd:unsignedInt"/>
                <element name="Type_DisplayWindowPosition" type="xsd:unsignedInt"/>
                <element name="Type_DisplayWindowInfo" type="DisplayWindowInfoType"/>
```

```
            <element                                    name="Type_DisplayWindowInfoList"
type="DisplayWindowInfoListType"/>
            <element name="Type_DisplayAttribute" type="DisplayAttribueType"/>

            <complexType name="PlayChannelType">
                  <choice>
                        <element name="MASTER" type="xsd:string"/>
                        <element name="LF" type="xsd:string"/>
                        <element name="RF" type="xsd:string"/>
                        <element name="CF" type="xsd:string"/>
                        <element name="LFE" type="xsd:string"/>
                        <element name="LS" type="xsd:string"/>
                        <element name="RS" type="xsd:string"/>
                        <element name="LFC" type="xsd:string"/>
                        <element name="RFC" type="xsd:string"/>
                        <element name="SD" type="xsd:string"/>
                        <element name="SL" type="xsd:string"/>
                        <element name="SR" type="xsd:string"/>
                        <element name="T" type="xsd:string"/>
                        <element name="B" type="xsd:string"/>
                  </choice>
            </complexType>
            <complexType name="DisplayWindowInfoType">
    <sequence>
                        <element   name="DisplayWindow"   type="DisplayWindowType"
minOccurs="1" />
                  </sequence>
            </complexType>
            <complexType name="DisplayWindowType">
                  <sequence>
                        <element    name="Type_DisplayWindowId"    type="xsd:string"
minOccurs="1" />
                        <element name="Type_DisplayWindowPosition">
                              <complexType>
                                    <sequence>
                                          <element                          name="CenterX"
type="xsd:unsignedInt" minOccurs="0"/>
                                          <element                          name="CenterY"
type="xsd:unsignedInt" minOccurs="0"/>
                                          <element                          name="CenterZ"
type="xsd:unsignedInt" minOccurs="0"/>
                                    </sequence>
                              </complexType>
                        </element>
                        <element name="Type_DisplayWindowSize">
                              <complexType>
                                    <sequence>
                                                <element    name="WindowWidth"
type="xsd:unsignedInt" minOccurs="0"/>
                                                <element   name="WindowHeight"
type="xsd:unsignedInt" minOccurs="0"/>
                                    </sequence>
                              </complexType>
                        </element>
                        <element  name="DisplayURI"  type="xsd:string"  minOccurs="1"
maxOccurs="1"/>
                        <element          name="DisplayMode"          type="xsd:string"
minOccurs="1" maxOccurs="1"/>
                  </sequence>
            </complexType>
            <complexType name="DisplayWindowInfoListType">
                  <sequence>
```

```
                    <element name="DisplayWindowInfo"
type="DisplayWindowInfoType" minOccurs="1" />
                </sequence>
            </complexType>
        </types>

        <message name="ListPresetsRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        </message>
        <message name="ListPresetsResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="CurrentPresetNameList" ref="igrs:Type_PresetNameList"/>
        </message>
        <message name="SelectPresetsRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
            <part name="PresetName" ref="igrs:Type_PresetName"/>
        </message>
        <message name="SelectPresetsResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="GetBrightnessRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        </message>
        <message name="GetBrightnessResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="CurrentBrightness" ref="igrs:Type_Brightness"/>
        </message>
        <message name="SetBrightnessRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
            <part name="DesiredBrightness" ref="igrs:Type_Brightness"/>
        </message>
        <message name="SetBrightnessResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="GetContrastRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        </message>
        <message name="GetContrastResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="CurrentContrast" ref="igrs:Type_Contrast"/>
        </message>
        <message name="SetContrastRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
            <part name="DesiredContrast" ref="igrs:Type_Contrast"/>
        </message>
        <message name="SetContrastResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="GetSharpnessRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        </message>
        <message name="GetSharpnessResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="CurrentSharpness" ref="igrs:Type_Sharpness"/>
        </message>
        <message name="SetSharpnessRequest">
            <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
            <part name="DesiredSharpness" ref="igrs:Type_Sharpness"/>
        </message>
        <message name="SetSharpnessResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
```

```
<message name="GetVideoGainRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
</message>
<message name="GetVideoGainResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentVideoGain" ref="igrs:Type_VideoGain"/>
</message>
<message name="SetVideoGainRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="DesiredVideoGain" ref="igrs:Type_VideoGain"/>
</message>
<message name="SetVideoGainResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetVideoGreyLevelRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
</message>
<message name="GetVideoGreyLevelResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentVideoGreyLevel" ref="igrs:Type_VideoGreyLevel"/>
</message>
<message name="SetVideoGreyLevelRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="DesiredVideoGreyLevel" ref="igrs:Type_VideoGreyLevel"/>
</message>
<message name="SetVideoGreyLevelResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetColorTemperatureRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
</message>
<message name="GetColorTemperatureResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentColorTemperature" ref="igrs:Type_ColorTemperature"/>
</message>
<message name="SetColorTemperatureRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="DesiredColorTemperature" ref="igrs:Type_ColorTemperature"/>
</message>
<message name="SetColorTemperatureResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetMuteRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="Channel" ref="igrs:Type_PlayChannel"/>
</message>
<message name="GetMuteResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentMute" ref="igrs:Type_Mute"/>
</message>
<message name="SetMuteRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="Channel" ref="igrs:Type_PlayChannel"/>
        <part name="DesiredMute" ref="igrs:Type_Mute"/>
</message>
<message name="SetMuteResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetVolumeRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="Channel" ref="igrs:Type_PlayChannel"/>
</message>
```

```
<message name="GetVolumeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentVolume" ref="igrs:Type_Volume"/>
</message>
<message name="SetVolumeRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="Channel" ref="igrs:Type_PlayChannel"/>
        <part name="DesiredVolume" ref="igrs:Type_Volume"/>
</message>
<message name="SetVolumeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetLoudnessRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="Channel" ref="igrs:Type_PlayChannel"/>
</message>
<message name="GetLoudnessResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="CurrentLoudness" ref="igrs:Type_LoudnessSwitch"/>
</message>
<message name="SetLoudnessRequest">
        <part name="InstanceId" ref="igrs:Type_RenderingManagementInstanceId"/>
        <part name="Channel" ref="igrs:Type_PlayChannel"/>
        <part name="DesiredLoudness" ref="igrs:Type_Loudness"/>
</message>
<message name="SetLoudnessResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="QueryServiceAttributeRequest">
        <part name="Name" type="xsd:string"/>
</message>
<message name="QueryServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="AttributeType" type="xsd:string"/>
        <part name="AllowedValueRange" type="xsd:string" minOccurs="0"/>
        <part name="AllowedValueList" type="xsd:string" minOccurs="0"/>
</message>
<message name="GetAllDisplayWindowInfoRequest">
</message>.
<message name="GetAllDisplayWindowInfoResponse">
   <part name="ReturnCode" type="xsd:unsignedInt"/>
   <part name="DisplayWindowInfoList" type="igrs:Type_DisplayWindowInfoList"/>
</message>
<message name="GetPlayerDisplayAttributeRequest">
</message>.
<message name="GetPlayerDisplayAttributeResponse">
   <part name="ReturnCode" type="xsd:unsignedInt"/>
   <part name="PlayerDisplayAttribute" type="igrs:Type_DisplayWindowInfoList"/>
</message>
<message name="GetDisplayWindowSizeRequest">
   <part name="WindowId" type="Type_DisplayWindowId"/>
</message>.
<message name="GetDisplayWindowSizeResponse">
   <part name="ReturnCode" type="xsd:unsignedInt"/>
   <part name="DisplayWindowSize" type="igrs:Type_DisplayWindowSize"/>
</message>
<message name="SetDisplayWindowSizeRequest">
   <part name="WindowId" type="Type_DisplayWindowId"/>
   <part name="DesiredDisplayWindowSize" type="igrs:Type_DisplayWindowSize"/>
</message>.
<message name="SetDisplayWindowSizeResponse">
   <part name="ReturnCode" type="xsd:unsignedInt"/>
```

```
        </message>
        <message name="GetDisplayWindowPositionRequest">
           <part name="WindowId" type="Type_DisplayWindowId"/>
        </message>.
        <message name="GetDisplayWindowPositionResponse">
           <part name="ReturnCode" type="xsd:unsignedInt"/>
           <part name="DisplayWindowSize" type="igrs:Type_DisplayWindowPosition"/>
        </message>
        <message name="SetDisplayWindowPositionRequest">
           <part name="WindowId" type="Type_DisplayWindowId"/>
           <part                                    name="DesiredDisplayWindowPosition"
type="igrs:Type_DisplayWindowPosition"/>
        </message>.
        <message name="SetDisplayWindowPositionResponse">
           <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <portType name="RenderingManagementPortType">
                <operation name="ListPresets">
                        <input message="igrs:ListPresetsRequest"/>
                        <output message="igrs:ListPresetsResponse"/>
                </operation>
                <operation name="SelectPresets">
                        <input message="igrs:SelectPresetsRequest"/>
                        <output message="igrs:SelectPresetsResponse"/>
                </operation>
                <operation name="GetBrightness">
                        <input message="igrs:GetBrightnessRequest"/>
                        <output message="igrs:GetBrightnessResponse"/>
                </operation>
                <operation name="SetBrightness">
                        <input message="igrs:SetBrightnessRequest"/>
                        <output message="igrs:SetBrightnessResponse"/>
                </operation>
                <operation name="GetContrast">
                        <input message="igrs:GetContrastRequest"/>
                        <output message="igrs:GetContrastResponse"/>
                </operation>
                <operation name="SetContrast">
                        <input message="igrs:SetContrastRequest"/>
                        <output message="igrs:SetContrastResponse"/>
                </operation>
                <operation name="GetSharpness">
                        <input message="igrs:GetSharpnessRequest"/>
                        <output message="igrs:GetSharpnessResponse"/>
                </operation>
                <operation name="SetSharpness">
                        <input message="igrs:SetSharpnessRequest"/>
                        <output message="igrs:SetSharpnessResponse"/>
                </operation>
                <operation name="GetVideoGain">
                        <input message="igrs:GetVideoGainRequest"/>
                        <output message="igrs:GetVideoGainResponse"/>
                </operation>
                <operation name="SetVideoGain">
                        <input message="igrs:SetVideoGainRequest"/>
                        <output message="igrs:SetVideoGainResponse"/>
                </operation>
                <operation name="GetVideoGreyLevel">
                        <input message="igrs:GetVideoGreyLevelRequest"/>
                        <output message="igrs:GetVideoGreyLevelResponse"/>
                </operation>
                <operation name="SetVideoGreyLevel">
```

```
            <input message="igrs:SetVideoGreyLevelRequest"/>
            <output message="igrs:SetVideoGreyLevelResponse"/>
        </operation>
        <operation name="GetColorTemperature">
            <input message="igrs:GetColorTemperatureRequest"/>
            <output message="igrs:GetColorTemperatureResponse"/>
        </operation>
        <operation name="SetColorTemperature">
            <input message="igrs:SetColorTemperatureRequest"/>
            <output message="igrs:SetColorTemperatureResponse"/>
        </operation>
        <operation name="GetMute">
            <input message="igrs:GetMuteRequest"/>
            <output message="igrs:GetMuteResponse"/>
        </operation>
        <operation name="SetMute">
            <input message="igrs:SetMuteRequest"/>
            <output message="igrs:SetMuteResponse"/>
        </operation>
        <operation name="GetVolume">
            <input message="igrs:GetVolumeRequest"/>
            <output message="igrs:GetVolumeResponse"/>
        </operation>
        <operation name="SetVolume">
            <input message="igrs:SetVolumeRequest"/>
            <output message="igrs:SetVolumeResponse"/>
        </operation>
        <operation name="GetLoudness">
            <input message="igrs:GetLoudnessRequest"/>
            <output message="igrs:GetLoudnessResponse"/>
        </operation>
        <operation name="SetLoudness">
            <input message="igrs:SetLoudnessRequest"/>
            <output message="igrs:SetLoudnessResponse"/>
        </operation>
        <operation name="QueryServiceAttribute">
            <input message="igrs:QueryServiceAttributeRequest"/>
            <output message="igrs:QueryServiceAttributeResponse"/>
        </operation>
        <operation name="GetAllDisplayWindowinfo">
            <input message="igrs:GetAllDisplayWindowsRequest"/>
            <output message="igrs:GetAllDisplayWindowsResponse"/>
        </operation>
        <operation name="GetPlayerDisplayAttribute">
            <input message="igrs:GetPlayerDisplayAttributeRequest"/>
            <output message="igrs:GetPlayerDisplayAttributeResponse"/>
    </operation>.
    <operation name="GetDisplayWindowSize">
       <input message="igrs:GetDisplayWindowSizeRequest"/>
            <output message="igrs:GetDisplayWindowSizeResponse"/>
    </operation>.
    <operation name="SetDisplayWindowSize">
       <input message="igrs:SetDisplayWindowSizeRequest"/>
            <output message="igrs:SetDisplayWindowSizeResponse"/>
    </operation>.
    <operation name="GetDisplayWindowPosition">
       <input message="igrs:GetDisplayWindowPositionRequest"/>
            <output message="igrs:GetDisplayWindowPositionResponse"/>
    </operation>.
    <operation name="SetDisplayWindowPosition">
       <input message="igrs:SetDisplayWindowPositionRequest"/>
            <output message="igrs:SetDisplayWindowPositionResponse"/>
```

```
            </operation>.

            <igrs:serviceAttribute name="CurrentPresetProfile" type="Type_PresetProfileList"/>
                <igrs:serviceAttribute      name="CurrentBrightness"      type="xsd:unsignedInt"
minOccurs="0"/>
                <igrs:serviceAttribute       name="CurrentContrast"       type="xsd:unsignedInt"
minOccurs="0" />
                <igrs:serviceAttribute         name="CurrentMute"         type="xsd:unsignedInt"
minOccurs="0" />

        </portType>
        <binding                                  name="RenderingManagementServiceIGRSPipe"
type="igrs:RenderingManagementPortType">
                <operation name="ListPresets">
                </operation>
                <operation name="SelectPresets">
                </operation>
                <operation name="GetBrightness">
                </operation>
                <operation name="SetBrightness">
                </operation>
                <operation name="GetContrast">
                </operation>
                <operation name="SetContrast">
                </operation>
                <operation name="GetSharpness">
                </operation>
                <operation name="SetSharpness">
                </operation>
                <operation name="GetVideoGain">
                </operation>
                <operation name="SetVideoGain">
                </operation>
                <operation name="GetVideoGreyLevel">
                </operation>
                <operation name="SetVideoGreyLevel">
                </operation>
                <operation name="GetColorTemperature">
                </operation>
                <operation name="SetColorTemperature">
                </operation>
                <operation name="GetMute">
                </operation>
                <operation name="SetMute">
                </operation>
                <operation name="GetVolume">
                </operation>
                <operation name="SetVolume">
                </operation>
                <operation name="GetLoudness">
                </operation>
                <operation name="SetLoudness">
                </operation>
                <operation name="QueryServiceAttribute">
                </operation>
                <operation name="GetAllDisplayWindowInfo">
                </operation>
                <operation name="GetPlayerDisplayAttribute">
                </operation>
                <operation name="GetDisplayWindowSize">
                </operation>
                <operation name="SetDisplayWindowSize">
```

```
                </operation>
                <operation name="GetDisplayWindowPosition">
                </operation>
                <operation name="SetDisplayWindowPosition">
                </operation>
        </binding>
        <service name="RenderingManagementService">
                <port                                   name="RenderingManagementService"
binding="igrs:RenderingManagementServiceIGRSPipe">
                </port>
        </service>

        <types>
                <element name="Type_ProtocolInfoList" type="PortocolInfoListType"/>
                <element name="Type_ProtocolInfo" type="ProtocolInfoType"/>
                <element name="Type_ConnectionId" type="xsd:unsignedInt"/>
                <element          name="Type_ServiceAttributeName"          type="xsd:string"
minOccurs="0"/>
                <element name="Type_SubscriptionId" type="xsd:string" minOccurs="0"/>
                <element name="Type_IPList" minOccurs="0">
                        <complexType>
                                <sequence>
                                        <element            name="IP"            type="xsd:string"
maxOccurs="unbounded"/>
                                </sequence>
                        </complexType>
                </element>
                <complexType name="PortocolInfoListType">
                        <sequence>
                                <element      name="ProtocolInfo"      type="ProtocolInfoType"
minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                </complexType>
                <complexType name="ProtocolInfoType">
                        <sequence>
                                <element name="TransportProtocol">
                                        <complexType>
                                                <sequence>
                                                        <element  name="Port"  type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                                                </sequence>
                                                <attribute name="Name" type="xsd:string"/>
                                        </complexType>
                                </element>
                                <element          name="ParameterList"          type="xsd:string"
minOccurs="0"/>
                                <element          name="AdditionalInfo"          type="xsd:string"
minOccurs="0"/>
                        </sequence>
                </complexType>
        </types>
        <message name="GetProtocolInfoRequest"/>
        <message name="GetProtocolInfoResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="ProtocolInfoList" ref="igrs:Type_ProtocolInfoList"/>
                <part name="IPList" ref="igrs:Type_IPList"/>
        </message>
        <message name="PrepareforConnectionRequest">
                <part name="RemoteProtocolInfo" type="Type_ProtocolInfo"/>
        </message>
        <message name="PrepareforConnectionResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
```

```
        <part name="ConnectionId" type="Type_ConnectionId"/>
    </message>
    <message name="ReleaseConnectionRequest">
        <part name="ConnectionId" type="Type_ConnectionId"/>
    </message>
    <message name="ReleaseConnectionResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
    </message>
    <message name="SubscribeServiceAttributeRequest">
        <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
    </message>
    <message name="SubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="SubscribeId" type="Type_SubscriptionId"/>
    </message>
    <message name="UnsubscribeServiceAttributeRequest">
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
    </message>
    <message name="UnsubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
    </message>

    <portType name="FileConnectionManagementPortType">
        <operation name="GetProtocolInfo">
            <input message="igrs:GetProtocolInfoRequest"/>
            <output message="igrs:GetProtocolInfoResponse"/>
        </operation>
        <operation name="PrepareforConnection">
            <input message="igrs:PrepareforConnectionRequest"/>
            <output message="igrs:PrepareforConnectionResponse"/>
        </operation>
        <operation name="ReleaseConnection">
            <input message="igrs:ReleaseConnectionRequest"/>
            <output message="igrs:ReleaseConnectionResponse"/>
        </operation>
        <operation name="SubscribeServiceAttribute">
            <input message="igrs:SubscribeServiceAttributeRequest"/>
            <output message="igrs:SubscribeServiceAttributeResponse"/>
        </operation>
        <operation name="UnsubscribeServiceAttribute">
            <input message="igrs:UnsubscribeServiceAttributeRequest"/>
            <output message="igrs:UnsubscribeServiceAttributeResponse"/>
        </operation>

        <igrs:serviceAttribute name="ProtocolInfoList" type="Type_ProtocolInfoList"
notifiable="true"/>
        <igrs:serviceAttribute name="IPList" type="Type_IPList" notifiable="true"/>

    </portType>

    <binding name="FileConnectionManagementServiceIGRSPipe"
type="igrs:FileConnectionManagementPortType">
        <operation name="GetProtocolInfo">
        </operation>
        <operation name="PrepareforConnection">
        </operation>
        <operation name="ReleaseConnection">
        </operation>
        <operation name="SubscribeServiceAttribute">
        </operation>
        <operation name="UnsubscribeServiceAttribute">
        </operation>
```

```
        </binding>
        <service name="FileConnectionManagementService">
                <port                                   name="FileConnectionManagementService"
binding="igrs:FileConnectionManagementServiceIGRSPipe">
                </port>
        </service>

        <types>
                <element name=" Type_AuthenticationKey" type="xsd:string"/>
                <element name=" Type_InstanceId" type="xsd:string"/>
                <element name=" Type_ObjectId" type="xsd:string"/>
                <element name=" Type_SubscriptionId" type="xsd:string"/>
                <element name=" Type_FilterRule" type="xsd:string"/>
                <element name=" Type_SortRule" type="xsd:string" />
                <element name=" Type_ServiceAttributeName" type="xsd:string"/>
                <element name=" Type_Count" type="xsd:int"/>
                <element name=" Type_SortCapability" type="xsd:string"/>
                <element name=" Type_SearchCapability" type="xsd:string"/>
                <element name=" Type_DeleteMode" type="xsd:string"/>
                <element                    name="                    Type_UserAuthenticationInfo"
type="UserAuthenticationInfoType"/>
                <element name=" Type_ObjectAttribute" type="ObjectAttributeType"/>
                <element name=" Type_ObjectURI" type="xsd:string"/>
                <element name=" Type_ObjectIdList">
                        <complexType>
                                <sequence>
                                        <element        name="ObjectId"         type="xsd:string"
minOccurs="1" maxOccurs="unbounded"/>
                                </sequence>
                        </complexType>
                </element>
                <element  name="Type_ObjectList">
                        <complexType>
                                <sequence>
                                        <element    name="Object"    type="ObjectAttributeType"
minOccurs="0"/>
                                </sequence>
                        </complexType>
                </element>
                <element name= "Type_ObjectURITreeList">
                        <complexType>
                                <sequence>
                                        <element                                name="ObjectURITree"
type="ObjectURITreeType" minOccurs="1" maxOccurs="unbounded"/>
                                </sequence>
                        </complexType>
                </element>
                <complexType name="ObjectURITreeType">
                      <attribute name="ObjectAttribute" type="ObjectAttributeType"/>
                      <attribute name= "ObjectURI" type ="xsd:string"/>
                                <sequence>
                                        <element                                name="ObjectURITree"
type="ObjectURITreeType" minOccurs="0" maxOccurs="unbounded"/>
                                </sequence>
                </complexType>
                <complexType name="UserAuthenticationInfoType">
                        <sequence>
                                <element name="UserInfo" minOccurs="0">
                                        <complexType>
                                                <sequence>
                                                        <element                name="UserName"
type="xsd:string"/>
```

```
                                        <element          name="UserPassword"
type="xsd:string"/>
                                    </sequence>
                                </complexType>
                            </element>
                            <element name="DeviceInfo" minOccurs="0">
                                <complexType>
                                    <sequence>
                                        <element                name="DeviceId"
type="xsd:string"/>
                                        <element            name="DeviceName"
type="xsd:string"/>
                                    </sequence>
                                </complexType>
                            </element>
                            <element name="3rdPartyAuthen" minOccurs="0">
                                <complexType>
                                    <sequence>
                                        <element    name="3rdPartyAuthenType"
type="xsd:string"/>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
                    </complexType>
                    <complexType name="ObjectAttributeType">
                        <sequence>
                            <element name="ObjectType" type="xsd:string"/>
                            <element name="ObjectId" type="xsd:string"/>
                            <element name="ObjectName" type="xsd:string"/>
                            <element name="ParentId" type="xsd:string" minOccurs="0"/>
                            <element name="DeviceId" type="xsd:string" minOccurs="0"/>
                            <element          name="DeviceName"          type="xsd:string"
minOccurs="0"/>
                            <element name="AccessRight">
                                <complexType>
                                    <sequence>
                                        <element name="Read" type="xsd:string"
minOccurs="0"/>
                                        <element name="Write" type="xsd:string"
minOccurs="0"/>
                                        <element name="Hide" type="xsd:string"
minOccurs="0"/>
                                    </sequence>
                                </complexType>
                            </element>
                            <element          name="CreateTime"          type="xsd:string"
minOccurs="0"/>
                            <element        name="LastAccessTime"        type="xsd:string"
minOccurs="0"/>
                            <element          name="LastWriteTime"         type="xsd:string"
minOccurs="0"/>
                            <element name="Size" type="xsd:string" minOccurs="0"/>
                            <element    name="Num_SubDirectories"    type="unsignedInt"
minOccurs="0"/>
                            <element        name="Num_SubFiles"        type="unsignedInt"
minOccurs="0"/>
                        </sequence>
                    </complexType>
                </types>
                <message name="GetAuthenticationKeyRequest">
                    <part name="InstanceId" type="Type_InstanceId"/>
```

```
        <part name="UserAuthenticationInfo" type="Type_UserAuthenticationInfo"/>
</message>
<message name="GetAuthenticationKeyResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="AuthenticationKey" type="xsd:string"/>
</message>
<message name="GetSortCapabilityRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
</message>
<message name="GetSortCapabilityResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="SortCaps" ref="igrs:Type_SortCapability"/>
</message>
<message name="GetSearchCapabilityRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
</message>
<message name="GetSearchCapabilityResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="SearchCaps" ref="igrs:Type_SearchCapability"/>
</message>
<message name="BrowseRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ObjectId" type="Type_ObjectId"/>
        <part name="BrowseFilter" type="Type_FilterRule"/>
        <part name="StartOffset" type="Type_Count"/>
        <part name="RequestedCount" type="Type_Count"/>
        <part name="SortRule" type="Type_SortRule"/>
</message>
<message name="BrowseResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="Result" type="Type_ObjectList"/>
        <part name="NumberReturned" type="Type_Count"/>
        <part name="NumberTotalMatched" type="Type_Count"/>
</message>
<message name="GetAttributeRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ObjectId" type="Type_ObjectId"/>
</message>
<message name="GetAttributeResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="ObjectAttribute" type="Type_ObjectAttribute"/>
</message>
<message name="SetAttributeRequest">
        <part name="AuthenticationKey" tpe="Type_AuthenticationKey"/>
        <part name="ObjectId" type="Type_ObjectId"/>
        <part name="ObjectAttribute" type="Type_ObjectAttribute"/>
</message>
<message name="SetAttributeResponse">
        <part name="ReturnCode" type="unsignedInt"/>
</message>
<message name="SearchRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ObjectIdList" type="Type_ObjectIdList"/>
        <part name="SearchRule" type="Type_FilterRule"/>
        <part name="StartOffset" type="Type_Count"/>
        <part name="RequestedCount" type="Type_Count"/>
        <part name="SortRule" ref="igrs:Type_SortRule"/>
</message>
<message name="SearchResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="Result" type="Type_ObjectList"/>
        <part name="NumberReturned" type="Type_Count"/>
```

```
            <part name="NumberTotalMatched" type="Type_Count"/>
        </message>
        <message name="GetBrowseFilterRequest">
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        </message>
        <message name="GetBrowseFilterResponse">
            <part name="ReturnCode" type="unsignedInt"/>
            <part name="BrowseFilter" type="Type_FilterRule"/>
        </message>
        <message name="SetBrowseFilterRequest">
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
            <part name="BrowseFilter" type="Type_FilterRule"/>
        </message>
        <message name="SetBrowseFilterResponse">
            <part name="ReturnCode" type="unsignedInt"/>
        </message>
        <message name="NewRequest">
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
            <part name="ParentId" type="Type_ObjectId"/>
            <part name="ObjectAttribute" type="Type_ObjectAttribute"/>
        </message>
    <message name="NewResponse">
          <part name="ReturnCode" type="unsignedInt"/>
            <part name="ObjectId" type="Type_ObjectId"/>
        </message>
        <message name="CopyRequest">]
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
            <part name="SourceObjectId" type="Type_ObjectId"/>
            <part name="DestParentId" type="Type_ObjectId"/>
    </message>
        <message name="CopyResponse">
            <part name="ReturnCode" type="unsignedInt"/>
            <part name="DestObjectId" type="Type_ObjectId"/>
            <part name="DestObjectAttribute" type="Type_ObjectAttribute"/>
        </message>
        <message name="MoveRequest">
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
            <part name="SourceObjectId" type="Type_ObjectId"/>
            <part name="DestParentId" type="Type_ObjectId"/>
        </message>
        <message name="MoveResponse">
            <part name="ReturnCode" type="unsignedInt"/>
            <part name="DestObjectId" type="Type_ObjectId"/>
            <part name="DestObjectAttribute" type="Type_ObjectAttribute"/>
        </message>
        <message name="DeleteRequest">
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
            <part name="ObjectId" type="Type_ObjectId"/>
            <part name="DeleteMode" type=" Type_DeleteMode"/>
        </message>
        <message name="DeleteResponse">
            <part name="ReturnCode" type="unsignedInt"/>
        </message>
        <message name="PrepareforDownloadRequest">
            <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
            <part name="SourceObjectIdList" type="Type_ObjectIdList"/>
        </message>
        <message name="PrepareforDownloadResponse">
            <part name="ReturnCode" type="unsignedInt"/>
            <part name="SourceObjectURITreeList" type="Type_ObjectURITreeList"/>
        </message>
        <message name="PrepareforUploadRequest">
```

```
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ObjectAttribute" type="Type_ ObjectAttribute"/>
        <part name="DestParentId" type="Type_ObjectId"/>
</message>
<message name="PrepareforUploadResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="DestParentURI" type="Type_ObjectURI"/>
</message>
<message name="SubscribeObjectChangeRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ObjectId" type="Type_ObjectId"/>]
</message>
<message name="SubscribeObjectChangeResponse">
        <part name="ReturnCode" type="unsignedInt"/>
</message>
<message name="UnSubscribeObjectChangeRequest">
        <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ObjectId" type="Type_ObjectId"/>
</message>
<message name="UnSubscribeObjectChangeResponse">
        <part name="ReturnCode" type="unsignedInt"/>
</message>
<message name="SubscribeServiceAttributeRequest">
   <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
</message>
<message name="SubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="unsignedInt"/>
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnSubscribeServiceAttributedRequest">
   <part name="AuthenticationKey" type="Type_AuthenticationKey"/>
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnSubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="unsignedInt"/>
</message>
<portType name="FileAccessManagementPortType">
        <operation name="GetAuthenticationKey">
                <input message="igrs:GetAuthenticationKeyRequest"/>
                <output message="igrs:GetAuthenticationKeyResponse"/>
        </operation>
        <operation name="GetSortCapability">
                <input message="igrs:GetSortCapabilityRequest"/>
                <output message="igrs:GetSortCapabilityResponse"/>
        </operation>
        <operation name="GetSearchCapability">
                <input message="igrs:GetSearchCapabilityRequest"/>
                <output message="igrs:GetSearchCapabilityResponse"/>
        </operation>
        <operation name="Browse">
                <input message="igrs:BrowseRequest"/>
                <output message="igrs:BrowseResponse"/>
        </operation>
        <operation name="GetAttribute">
                <input message="igrs:GetAttributeRequest"/>
                <output message="igrs:GetAttributeResponse"/>
        </operation>
        <operation name="SetAttribute">
                <input message="igrs:SetAttributeRequest"/>
                <output message="igrs:SetAttributeResponse"/>
        </operation>
```

```
                <operation name="Search">
                        <input message="igrs:SearchRequest"/>
                        <output message="igrs:SearchResponse"/>
                </operation>
                <operation name="GetBrowseFilter">
                        <input message="igrs:GetBrowseFilterRequest"/>
                        <output message="igrs:GetBrowseFilterResponse"/>
                </operation>
                <operation name="SetBrowseFilter">
                        <input message="igrs: SetBrowseFilterRequest"/>
                        <output message="igrs: SetBrowseFilterResponse"/>
                </operation>
                <operation name="New">
                        <input message="igrs: NewRequest"/>
                        <output message="igrs: NewResponse"/>
                </operation>
                <operation name="Copy">
                        <input message="igrs:CopyRequest"/>
                        <output message="igrs:CopyResponse"/>
                </operation>
                <operation name="Move">
                        <input message="igrs:MoveRequest"/>
                        <output message="igrs:MoveResponse"/>
                </operation>
                <operation name="Delete">
                        <input message="igrs:DeleteRequest"/>
                        <output message="igrs:DeleteResponse"/>
                </operation>
                <operation name="PrepareforDownload">
                        <input message="igrs:PrepareforDownloadRequest"/>
                        <output message="igrs:PrepareforDownloadResponse"/>
                </operation>
                <operation name="PrepareforUpload">
                        <input message="igrs:PrepareforUploadRequest"/>
                        <output message="igrs:PrepareforUploadResponse"/>
                </operation>
                <operation name="SubscribeObjectChange">
                        <input message="igrs:SubscribeObjectChangeRequest"/>
                        <output message="igrs:SubscribeObjectChangeResponse"/>
                </operation>
                <operation name="UnsubscribeObjectChange">
                        <input message="igrs:UnsubscribeObjectChangeRequest"/>
                        <output message="igrs:UnsubscribeObjectChangeResponse"/>
                </operation>
                <operation name="SubscribeServiceAttribute">
                        <input message="igrs:SubscribeServiceAttributeRequest"/>
                        <output message="igrs:SubscribeServiceAttributeResponse"/>
                </operation>
                <operation name="UnSubscribeServiceAttribute">
                        <input message="igrs:UnSubscribeServiceAttributeRequest"/>
                        <output message="igrs:UnSubscribeServiceAttributeResponse"/>
                </operation>
                <igrs:serviceAttribute name="SortCaps" type="Type_SortCapability"
minOccurs="0" notifiable="true"/>
                <igrs:serviceAttribute name="SearchCaps" type="Type_SearchCapability"
minOccurs="0" notifiable="true"/>
        </portType>
        <binding name="FileAccessManagementServiceIGRSPipe"
type="igrs:FileAccessManagementPortType">
                <operation name="GetAuthenticationKey">
                </operation>
```

```
                <operation name="GetSortCapability">
                </operation>
                <operation name="GetSearchCapability">
                </operation>
                <operation name="Browse">
                </operation>
                <operation name="GetAttribute">
                </operation>
                <operation name="SetAttribute">
                </operation>
                <operation name="Search">
                </operation>
                <operation name="GetBrowseFilter">
                </operation>
                <operation name="SetBrowseFilter">
                </operation>
                <operation name="New">
                </operation>
                <operation name="Copy">
                </operation>
                <operation name="Move">
                </operation>
                <operation name="Delete">
                </operation>
                <operation name="PrepareforDownload">
                </operation>
                <operation name="PrepareforUpload">


                </operation>

                <operation name="SubscribeObjectChange">


                </operation>
                <operation name="UnsubscribeObjectChange">


                </operation>
                <operation name="SubscribeServiceAttribute">


                </operation>
                <operation name="UnsubscribeServiceAttribute">


                </operation>
        </binding>
        <service name="FileAccessManagementService">
                <port                                   name="FileAccessManagementService"
binding="igrs:FileAccessManagementServiceIGRSPipe">
                </port>
        </service>

        <types>
                <element name="Type_ObjectId" type="xsd:string"/>
                <element name="Type_ContentList" type="ContentListType"/>
                <element name="Type_ContentListScale" type="xsd:string"/>
                <element name="Type_FilterRule" type="xsd:string"/>
                <element name="Type_BrowseFlag" type="BrowseFlagType"/>
                <element name="Type_SortRule" type="xsd:string"/>
                <element name="Type_Count" type="xsd:Int" minOccurs="0"/>
```

```
            <element name="Type_TransferInstanceId" type="xsd:unsignedInt"
minOccurs="0"/>
            <element name="Type_TransferState" type="TransferStateType"
minOccurs="0"/>
            <element name="Type_Length" type="xsd:string" minOccurs="0"/>
            <element name="Type_TagList" type="xsd:string" minOccurs="0"/>
            <element name="Type_URI" type="xsd:string" minOccurs="0"/>
            <element name="Type_ContentUpdateId" type="xsd:unsignedInt"/>

            <element name="Type_MediaFormat" type="MediaFormatType"/>
            <element name="Type_ServiceAttributeName"
type="ServiceAttributeNameType"/>
            <element name="Type_SubscriptionId" type="xsd:unsignedInt"/>
            <element name="Type_SearchAttributeName" type="xsd:string"/>
            <element name="Type_AttributeValueList" type="xsd:string"/>
            <element name="Type_SearchCapabilityList">
                  <complexType>
                        <sequence>
                              <element name="SearchCapability" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                  </complexType>
            </element>
            <element name="Type_SortCapabilityList">
                  <complexType>
                        <sequence>
                              <element name="SortCapability" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                  </complexType>
            </element>
            <element name="Type_SearchAttributeValueList">
                  <complexType>
                        <sequence>
                              <element name="SearchAttributeValue"
type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                  </complexType>
            </element>

            <complexType name="ContentListType">
                  <sequence>
                        <element name="Container" type="igrs:ContainerType"
minOccurs="0" maxOccurs="unbounded"/>
                        <element name="Item" type="igrs:ItemType" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
            </complexType>
            <complexType name="ContainerType">
                  <sequence>
                        <element name="ContainerAttribute"
type="igrs:ContainerAttributeType"/>
                  </sequence>
                  <attribute name="Num_containers" type="xsd:string"/>
                  <attribute name="Num_items" type="xsd:string"/>
            </complexType>
            <complexType name="ContainerAttributeType">
                  <sequence>
                        <element name="ObjectId" type="xsd:string"/>
                        <element name="ParentId" type="xsd:string" minOccurs="0"/>
                        <element name="DeviceId" type="xsd:string" minOccurs="0"/>
```

```
                        <element          name="DeviceName"          type="xsd:string"
minOccurs="0"/>
                        <element          name="ObjectPath"          type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectName" type="xsd:string"/>
                        <element name="ObjectStoreAttribute">
                                <complexType>
                                        <sequence>
                                                <element                name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                                                <element  name="Hide"  type="xsd:string"
minOccurs="0"/>
                                                <element                name="Document"
type="xsd:string" minOccurs="0"/>
                                                <element                 name="Content"
type="xsd:string" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element          name="CreateTime"          type="xsd:string"
minOccurs="0"/>
                        <element        name="LastAccessTime"        type="xsd:string"
minOccurs="0"/>
                        <element         name="LastWriteTime"        type="xsd:string"
minOccurs="0"/>
                        <element         name="AdditionalInfo"       type="xsd:string"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ItemType">
                <sequence>
                        <element name="ItemAttribute" type="igrs:ItemAttributeType"/>
                        <element name="ItemReferences" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element                name="KeyFrame"
minOccurs="0">
                                                        <attribute              name="Id"
type="igrs:Type_ObjectId"/>
                                                </element>
                                                <element        name="MicroPhotograph"
minOccurs="0">
                                                        <attribute              name="Id"
type="igrs:Type_ObjectId"/>
                                                </element>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
        </complexType>
        <complexType name="ItemAttributeType">
                <sequence>
                        <element name="Type_ObjectId" type="xsd:string"/>
                        <element name="ParentId" type="xsd:string" minOccurs="0"/>
                        <element name="DeviceId" type="xsd:string" minOccurs="0"/>
                        <element          name="DeviceName"          type="xsd:string"
minOccurs="0"/>
                        <element          name="ObjectPath"          type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectName" type="xsd:string"/>
                        <element name="ObjectStoreAttribute">
                                <complexType>
```

```
                                        <sequence>
                                                <element                name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                                                <element  name="Hide"  type="xsd:string"
minOccurs="0"/>
                                                <element                name="Document"
type="xsd:string" minOccurs="0"/>
                                                <element                 name="Content"
type="xsd:string" minOccurs="0"/>
                                        </sequence>
                                </complexType>
                        </element>
                        <element          name="CreateTime"          type="xsd:string"
minOccurs="0"/>
                        <element        name="LastAccessTime"        type="xsd:string"
minOccurs="0"/>
                        <element         name="LastWriteTime"        type="xsd:string"
minOccurs="0"/>
                        <element     name="ObjectType"     type="igrs:ObjectTypeType"
minOccurs="0"/>
                        <element name="Size" type="xsd:string" minOccurs="0"/>
                        <element name="Duration" type="xsd:string" minOccurs="0"/>
                        <element name="Encrypted" type="xsd:string" minOccurs="0"/>
                <element name="DRMContendId" type="xsd:string" minOccurs="0"/>
                        <element                                name="Type_MediaFormat"
type="MediaFormatType"/>
                        <element name="ObjectURI" type="xsd:string"/>
                        <element        name="Type_ConnectionManagementServiceId"
type="xsd:unsignedInt"/>
                        <element name="Choices" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element  name="Choice"  minOccurs="0"
maxOccurs="unbounded">
                                                        <complexType>
                                                                <sequence>
                                                                        <element
name="Selection" minOccurs="0" maxOccurs="unbounded">
        <complexType>
        <attribute name="Id" type="xsd:string"/>
        <attribute name="Name" type="xsd:string"/>
        </complexType>
                                                                        </element>
                                                                </sequence>
                                                                <attribute         name="Id"
type="xsd:string"/>
                                                                <attribute    name="Name"
type="xsd:string"/>
                                                                <attribute   name="Default"
type="xsd:string"/>
                                                                <attribute     name="Type"
type="xsd:string"/>
                                                        </complexType>
                                                </element>
                                        </sequence>
                                </complexType>
                        </element>
                </sequence>
```

```
		</complexType>
		<complexType name="ObjectTypeType">
			<choice>
				<element name="Audio" type="igrs:AudioType"/>
				<element name="Video" type="xsd:string"/>
				<element name="Photo" type="xsd:string"/>
				<element name="Doc" type="xsd:string"/>
				<element name="Camera" type="xsd:string"/>
				<element name="Screen" type="xsd:string"/>
			</choice>
		</complexType>
		<complexType name="AudioType">
			<sequence>
				<element name="Singer" type="xsd:string" minOccurs="0"/>
				<element name="Genre" type="xsd:string" minOccurs="0"/>
				<element name="MusicDisc" type="xsd:string" minOccurs="0"/>
				<element name="Author" type="xsd:string" minOccurs="0"/>
			</sequence>
		</complexType>

		<complexType name="BrowseFlagType">
			<choice>
				<element name="Constant_ContainerSelfInfo"
type="xsd:string"/>
				<element name="Constant_ContainerChildrenInfo"
type="xsd:string"/>
			</choice>
		</complexType>

		<complexType name="TransferStateType">
			<choice>
				<element name="IN_PROGRESS" type="xsd:string"/>
				<element name="STOPPED" type="xsd:string"/>
				<element name="ERROR" type="xsd:string"/>
				<element name="COMPLETED" type="xsd:string"/>
			</choice>
		</complexType>

		<complexType name="MediaFormatType">
			<sequence>
				<element name="AudioFormat" type="xsd:string"
minOccurs="0"/>
				<element name="VideoFormat" type="xsd:string"
minOccurs="0"/>
				<element name="PhotoFormat" type="xsd:string"
minOccurs="0"/>
				<element name="RemoteDesktop" type="RemoteDesktop"
minOccurs="0"/>
				<element name="VideoCamera" type="VideoCamera"
minOccurs="0"/>
			</sequence>
			<attribute name="Name" type="xsd:string"/>
			<attribute name="Type" type="xsd:string"/>
		</complexType>
		<complexType name="ServiceAttributeNameType">
			<choice>
				<element name="CONTENTUPDATEID" type="xsd:string"/>
				<element name="SORTCAPS" type="xsd:string"/>
				<element name="OBJECTID" type="xsd:string"/>
				<element name="TRANSFERSTATE" type="xsd:string"/>
			</choice>
		</complexType>
```

```
        </types>

        <message name="GetSearchCapabilityListRequest"/>
        <message name="GetSearchCapabilityListResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="SearchCaps" ref="igrs:Type_SearchCapabilityList"/>
        </message>
        <message name="GetSearchAttributeCapabilityListRequest"/>
        <message name="GetSearchAttributeCapabilityListResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part                                          name="SearchAttributeCaps"
ref="igrs:Type_SearchAttributeCapabilityList"/>
        </message>
        <message name="GetSortCapabilityListRequest"/>
        <message name="GetSortCapabilityListResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="SortCaps" ref="igrs:Type_SortCapabilityList"/>
        </message>
        <message name="GetContentUpdateIdRequest"/>
        <message name="GetContentUpdateIdResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="ContentUpdateId" ref="igrs:Type_ContentUpdateId"/>
        </message>
        <message name="BrowseRequest">
                <part name="ObjectId" ref="igrs:Type_ObjectId"/>
                <part name="BrowseFlag" ref="igrs:Type_BrowseFlag"/>
                <part name="BrowseRule" ref="igrs:Type_FilterRule"/>
                <part name="Offset" ref="igrs:Type_Offset"/>
                <part name="RequestCount" ref="igrs:Type_Count"/>
                <part name="SortRule" ref="igrs:Type_SortRule"/>
           <part name="ResultScale" ref="igrs:Type_ContentListScale"/>
        </message>
        <message name="BrowseResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="Result" ref="igrs:Type_ContentList"/>
                <part name="NumberReturned" ref="igrs:Type_Count"/>
                <part name="ContainerNumberTotal" ref="igrs:Type_Count"/>
                <part name="ItemNumberTotal" ref="igrs:Type_Count"/>
        </message>
        <message name="SearchRequest">
                <part name="ObjectId" ref="igrs:Type_ObjectId"/>
                <part name="SearchRule" ref="igrs:Type_SearchRule"/>
                <part name="Offset" ref="igrs:Type_Offset"/>
                <part name="RequestCount" ref="igrs:Type_Count"/>
                <part name="SortRule" ref="igrs:Type_SortRule"/>
        </message>
        <message name="SearchResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="Result" ref="igrs:Type_ContentList"/>
                <part name="NumberReturned" ref="igrs:Type_Count"/>
                <part name="NumberTotalMatched" ref="igrs:Type_Count"/>
        </message>
        <message name="SearchAttributeValueRequest">
                <part name="ObjectId" ref="igrs:Type_ObjectId"/>
                <part name="SearchAttributeName" type="xsd:string"/>
                <part name="Offset" ref="igrs:Type_Offset"/>
                <part name="RequestCount" ref="igrs:Type_Count"/>
        </message>
        <message name="SearchAttributeValueResponse">
                <part name="ReturnCode" type="xsd:unsignedInt"/>
                <part name="AttributeValueList" type="Type_ AttributeValueList"/>
                <part name="NumberReturned" ref="igrs:Type_Count"/>
```

```
        <part name="NumberTotalMatched" ref="igrs:Type_Count"/>
</message>
<message name="ConvertMediaFormatRequest">
    <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="CurrentMediaFormat" ref="igrs:Type_MediaFormat"/>
        <part name="TargetMediaFormat" ref="igrs:Type_MediaFormat"/>
</message>
<message name="ConvertMediaFormatResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="CreateObjectRequest">
        <part name="ContainerId" ref="igrs:Type_ObjectId"/>
        <part name="Elements" ref="igrs:Type_ContentList"/>
</message>
<message name="CreateObjectResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="Result" ref="igrs:Type_ContentList"/>
</message>
<message name="DestroyObjectRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
</message>
<message name="DestroyObjectResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="UpdateObjectRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
        <part name="CurrentTag" ref="igrs:Type_TagList"/>
        <part name="NewTag" ref="igrs:Type_TagList"/>
</message>
<message name="UpdateObjectResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="ImportResourceRequest">
        <part name="SourceURI" ref="igrs:Type_URI"/>
        <part name="DestinationURI" ref="igrs:Type_URI"/>
</message>
<message name="ImportResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="ExportResourceRequest">
        <part name="SourceURI" ref="igrs:Type_URI"/>
        <part name="DestinationURI" ref="igrs:Type_URI"/>
</message>
<message name="ExportResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="DeleteResourceRequest">
        <part name="ResourceURI" ref="igrs:Type_URI"/>
</message>
<message name="DeleteResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="StopTransferResourceRequest">
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="StopTransferResourceResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="GetTransferProgressInfoRequest">
```

```
        <part name="TransferInstanceId" ref="igrs:Type_TransferInstanceId"/>
</message>
<message name="GetTransferProgressInfoResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="TransferStatus" ref="igrs:Type_TransferState"/>
        <part name="TransferLength" ref="igrs:Type_Length"/>
        <part name="TransferTotal" ref="igrs:Type_Total"/>
</message>
<message name="SubscribeContentUpdateEventRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
</message>
<message name="SubscribeContentUpdateEventResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="UnsubscribeContentUpdateEventRequest">
        <part name="ObjectId" ref="igrs:Type_ObjectId"/>
</message>
<message name="UnsubscribeContentUpdateEventResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>
<message name="SubscribeServiceAttributeRequest">
        <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
</message>
<message name="SubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnsubscribeServiceAttributeRequest">
        <part name="SubscriptionId" type="Type_SubscriptionId"/>
</message>
<message name="UnsubscribeServiceAttributeResponse">
        <part name="ReturnCode" type="xsd:unsignedInt"/>
</message>

<portType name="ContentIndexPortType">
        <operation name="GetSearchCapabilityList">
                <input message="igrs:GetSearchCapabilityListRequest"/>
                <output message="igrs:GetSearchCapabilityListResponse"/>
        </operation>
        <operation name="GetSearchAttributeCapabilityList">
                <input message="igrs:GetSearchAttributeCapabilityListRequest"/>
                <output message="igrs:GetSearchAttributeCapabilityListResponse"/>
        </operation>
        <operation name="GetSortCapabilityList">
                <input message="igrs:GetSortCapabilityListRequest"/>
                <output message="igrs:GetSortCapabilityListResponse"/>
        </operation>
        <operation name="GetContentUpdateId">
                <input message="igrs:GetContentUpdateIdRequest"/>
                <output message="igrs:GetContentUpdateIdResponse"/>
        </operation>
        <operation name="Browse">
                <input message="igrs:BrowseRequest"/>
                <output message="igrs:BrowseResponse"/>
        </operation>
        <operation name="Search">
                <input message="igrs:SearchRequest"/>
                <output message="igrs:SearchResponse"/>
        </operation>
        <operation name="SearchAttributeValue">
                <input message="igrs:SearchAttributeValueRequest"/>
                <output message="igrs:SearchAttributeValueResponse"/>
```

```
            </operation>
            <operation name="ConvertMediaFormat">
                  <input message="igrs:ConvertMediaFormatRequest"/>
                  <output message="igrs:ConvertMediaFormatResponse"/>
            </operation>
            <operation name="CreateObject">
                  <input message="igrs:CreateObjectRequest"/>
                  <output message="igrs:CreateObjectResponse"/>
            </operation>
            <operation name="DestroyObject">
                  <input message="igrs:DestroyObjectRequest"/>
                  <output message="igrs:DestroyObjectResponse"/>
            </operation>
            <operation name="UpdateObject">
                  <input message="igrs:UpdateObjectRequest"/>
                  <output message="igrs:UpdateObjectResponse"/>
            </operation>
            <operation name="ImportResource">
                  <input message="igrs:ImportResourceRequest"/>
                  <output message="igrs:ImportResourceResponse"/>
            </operation>
            <operation name="ExportResource">
                  <input message="igrs:ExportResourceRequest"/>
                  <output message="igrs:ExportResourceResponse"/>
            </operation>
            <operation name="DeleteResource">
                  <input message="igrs:DeleteResourceRequest"/>
                  <output message="igrs:DeleteResourceResponse"/>
            </operation>
            <operation name="StopTransferResource">
                  <input message="igrs:StopTransferResourceRequest"/>
                  <output message="igrs:StopTransferResourceResponse"/>
            </operation>
            <operation name="GetTransferProcessInfo">
                  <input message="igrs:GetTransferProcessRequest"/>
                  <output message="igrs:GetTransferProcessResponse"/>
            </operation>
            <operation name="SubscribeContentUpdateEvent">
                  <input message="igrs:SubscribeContentUpdateEventRequest"/>
                  <output message="igrs:SubscribeContentUpdateEventResponse"/>
            </operation>
            <operation name="UnsubscribeContentUpdateEvent">
                  <input message="igrs:UnsubscribeContentUpdateEventRequest"/>
                  <output message="igrs:UnsubscribeContentUpdateEventResponse"/>
            </operation>
            <operation name="SubscribeServiceAttribute">
                  <input message="igrs:SubscribeServiceAttributeRequest"/>
                  <output message="igrs:SubscribeServiceAttributeResponse"/>
            </operation>
            <operation name="UnsubscribeServiceAttribute">
                  <input message="igrs:UnsubscribeServiceAttributeRequest"/>
                  <output message="igrs:UnsubscribeServiceAttributeResponse"/>
            </operation>

            <igrs:serviceAttribute name="ContentUpdateId" type="xsd:unsignedInt"
notifiable="true"/>
            <igrs:serviceAttribute name="SortCaps" type="Type_SortCapabilityList"
notifiable="true"/>
            <igrs:serviceAttribute name="ObjectId" type="Type_ObjectId"
notifiable="true"/>
            <igrs:serviceAttribute name="TransferState" type="Type_TransferState"
minOccurs="0" notifiable="true"/>
```

```
</portType>
<binding name="ContentIndexServiceIGRSPipe" type="igrs:ContentIndexPortType">
        <operation name="GetSearchCapabilityList">


        </operation>
        <operation name="GetSearchAttributeCapabilityList">


        </operation>
        <operation name="GetSortCapabilityList">


        </operation>
        <operation name="GetContentUpdateId">


        </operation>
        <operation name="Browse">


        </operation>
        <operation name="Search">


        </operation>
        <operation name="SearchAttributeValue">


        </operation>
        <operation name="ConvertMediaFormat">


        </operation>
        <operation name="CreateObject">


        </operation>
        <operation name="DestroyObject">


        </operation>
        <operation name="UpdateObject">


        </operation>
        <operation name="ImportResource">


        </operation>
        <operation name="ExportResource">


        </operation>
        <operation name="DeleteResource">


        </operation>
        <operation name="StopTransferResource">
```

```
                </operation>
                <operation name="GetTransferProcessInfo">


                </operation>
                <operation name="SubscribeContentUpdateEvent">


                </operation>
                <operation name="UnsubscribeContentUpdateEvent">


                </operation>
                <operation name="SubscribeServiceAttribute">


                </operation>
                <operation name="UnsubscribeServiceAttribute">


                </operation>
        </binding>
        <service name="ContentIndexService">
                <port                                                    name="ContentIndexService"
binding="igrs:ContentIndexServiceIGRSPipe">
                </port>
        </service>


        <types>
                <element name="Type_ProtocolInfoList" type="ProtocolInfoListType"/>
                <element name="Type_ProtocolInfo" type="ProtocolInfoType"/>
                <element name="Type_ConnectionManagementServiceId" type="xsd:string"/>
                <element name="Type_ConnectionId" type="xsd:unsignedInt"/>
                <element       name="Type_TransportInstanceId"       type="xsd:unsignedInt"
minOccurs="0"/>
                <element name="Type_ConnectionIdList" type="ConnectionIdListType"/>
                <element      name="Type_ConnectionState"      type="ConnectionStateType"
minOccurs="0"/>
                <element name="Type_IPList" type="IPListType" minOccurs="0"/>
                <element name="Type_MediaFormatList" type="MediaFormatListType"/>
                <element                                          name="Type_ServiceAttributeName"
type="ServiceAttributeNameType" minOccurs="0"/>
                <element name="Type_SubscriptionId" type="xsd:string" minOccurs="0"/>
                <element                          name="Type_RenderingManagementInstanceID"
type="xsd:unsignedInt"/>
                <element                                           name="Type_ConnectionRoleFlag"
type="ConnectionRoleFlagType"/>
                <complexType name="ProtocolInfoListType">
                        <sequence>
                                <element   name="Type_ProtocolInfo"   type="ProtocolInfoType"
maxOccurs="unbounded"/>
                        </sequence>
                </complexType>
                <complexType name="ProtocolInfoType">
                        <sequence>
                                <element name="TransportProtocol">
                                        <complexType>
                                                <sequence>
                                                        <element   name="Port"   type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                                                </sequence>
```

```
                                        <attribute name="Name" type="xsd:string"/>
                                </complexType>
                        </element>
                        <element name="ControlProtocol" minOccurs="0">
                                <complexType>
                                        <sequence>
                                                <element  name="Port"  type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                                        </sequence>
                                        <attribute name="Name" type="xsd:string"/>
                                </complexType>
                        </element>
                        <element         name="ParameterList"         type="xsd:string"
minOccurs="0"/>
                        <element         name="AdditionalInfo"         type="xsd:string"
minOccurs="0"/>
                </sequence>
        </complexType>
        <complexType name="ConnectionIdListType">
                <sequence>
                        <element         name="ConnectionId"         type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
        </complexType>
        <complexType name="ConnectionStateType">
                <choice>
                        <element name="OK" type="xsd:string"/>
                        <element name="DISCONNECTED" type="xsd:string"/>
                        <element                name="CONTENTFORMATMISMATCH"
type="xsd:string"/>
                        <element                    name="INSUFFICIENTBANDWIDTH"
type="xsd:string"/>
                        <element name="UNRELIABLECHANNEL" type="xsd:string"/>
                        <element name="UNKNOWN" type="xsd:string"/>
                </choice>
        </complexType>
        <complexType name="IPListType">
                <sequence>
                        <element                name="IP"                type="xsd:string"
maxOccurs="unbounded"/>
                </sequence>
        </complexType>
        <complexType name="MediaFormatListType">
                <sequence>
                        <element     name="MediaFormat"     type="MediaFormatType"
maxOccurs="unbounded"/>
                </sequence>
        </complexType>
        <complexType name="MediaFormatType">
                <sequence>
                        <element         name="AudioFormat"         type="xsd:string"
minOccurs="0"/>
                        <element         name="VideoFormat"         type="xsd:string"
minOccurs="0"/>
                        <element         name="PhotoFormat"         type="xsd:string"
minOccurs="0"/>
                        <element     name="ReomteDesktop"     type="RemoteDesktop"
minOccurs="0"/>
                        <element       name="VideoCamera"       type="VideoCamera"
minOccurs="0"/>
                </sequence>
                <attribute name="Name" type="xsd:string"/>
```

```
                <attribute name="Type" type="xsd:string"/>
            </complexType>
            <complexType name="ServiceAttributeNameType">
                <choice>
                    <element name="CONNECTIONSTATE" type="xsd:string"/>
                    <element name="CONNECTIONIDLIST" type="xsd:string"/>
                    <element name="PROTOCOLINFOLIST" type="xsd:string"/>
                    <element name="MEDIAFORMATLIST" type="xsd:string"/>
                    <element name="IPLIST" type="xsd:string"/>
                </choice>
            </complexType>
            <complexType name="ConnectionRoleFlagType">
                <choice>
                    <element name="ASSERVER" type="xsd:string"/>
                    <element name="ASCLIENT" type="xsd:string"/>
                </choice>
            </complexType>
        </types>

        <message name="GetProtocolInfoRequest"/>
        <message name="GetProtocolInfoResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ProtocolInfoList" type="Type_ProtocolInfoList"/>
            <part name="MediaFormatList" type="Type_MediaFormatList"/>
            <part name="IPList" type="Type_IPList"/>
        </message>
        <message name="PrepareForConnectionRequest">
            <part name="RemoteProtocolInfo" type="Type_ProtocolInfo"/>
            <part    name="PeerCMSId"    type="Type_ConnectionManagementServiceId"
miniOccurs="0"/>
            <part name="PeerConnectionId" type="Type_ConnectionId" miniOccurs="0"/>
            <part name="ConnectionRoleFlag" type="Type_ConnectionRoleFlag"/>
            <part name="PeerIPList" type="Type_IPList" minOccurs="0"/>
        </message>
        <message name="PrepareForConnectionResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ConnectionId" type="Type_ConnectionId"/>
            <part       name="TransportInstanceId"       type="Type_TransportInstanceId"
miniOccurs="0"/>
            <part      name="RmsId"       type="Type_RenderingManagementInstanceId"
minOccurs="0"/>
            <part name="UsableIPList" type="Type_IPList" miniOccurs="0"/>
        </message>
        <message name="ReleaseConnectionRequest">
            <part name="ConnectionId" type="Type_ConnectionId"/>
        </message>
        <message name="ReleaseConnectionResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <message name="GetActiveConnectionIdListRequest"/>
        <message name="GetActiveConnectionIdlistResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="ConnectionIdList" type="Type_ConnectionIdList"/>
        </message>
        <message name="GetCurrentConnectionInfoRequest">
            <part name="ConnectionId" type="Type_ ConnectionId"/>
        </message>
        <message name="GetCurrentConnectionInfoResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="TransportInstanceId" type="Type_TransportInstanceId"/>
            <part name="ProtocolInfo" type="Type_ ProtocolInfo"/>
            <part name="PeerCMSId" type="Type_ConnectionManagementServiceId"/>
```

```
            <part name="PeerConnectionId" type="Type_ConnectionId"/>
            <part name="ConnectionState" type="Type_ConnectionState"/>
            <part name="RcsId" type="Type_RenderingManagementInstanceId"
miniOccurs="0"/>
        </message>
        <message name="SubscribeServiceAttributeRequest">
            <part name="ServiceAttributeName" type="Type_ServiceAttributeName"/>
        </message>
        <message name="SubscribeServiceAttributeResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="SubscriptionId" type="Type_SubscriptionId"/>
        </message>
        <message name="UnsubscribeServiceAttributeRequest">
            <part name="SubscriptionId" type="Type_SubscriptionId"/>
        </message>
        <message name="UnsubscribeServiceAttributeResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
        </message>
        <portType name="ConnectionManagementPortType">
            <operation name="GetProtocolInfo">
                <input message="igrs:GetProtocolInfoRequest"/>
                <output message="igrs:GetProtocolInfoResponse"/>
            </operation>
            <operation name="PrepareForConnection">
                <input message="igrs:PrepareForConnectionRequest"/>
                <output message="igrs:PrepareForConnectionResponse"/>
            </operation>
            <operation name="ReleaseConnection">
                <input message="igrs:ReleaseConnectionRequest"/>
                <output message="igrs:ReleaseConnectionResponse"/>
            </operation>
            <operation name="GetActiveConnectionIdList">
                <input message="igrs:GetActiveConnectionIdListRequest"/>
                <output message="igrs:GetActiveConnectionIdListResponse"/>
            </operation>
            <operation name="GetCurrentConnectionInfo">
                <input message="igrs:GetCurrentConnectionInfoRequest"/>
                <output message="igrs:GetCurrentConnectionInfoResponse"/>
            </operation>
            <operation name="SubscribeServiceAttribute">
                <input message="igrs:SubscribeServiceAttributeRequest"/>
                <output message="igrs:SubscribeServiceAttributeResponse"/>
            </operation>
            <operation name="UnsubscribeServiceAttribute">
                <input message="igrs:UnsubscribeServiceAttributeRequest"/>
                <output message="igrs:UnsubscribeServiceAttributeResponse"/>
            </operation>

            <igrs:serviceAttribute name="ConnectionState" type="Type_ConnectionState"
minOccurs="0" notifiable="true"/>
            <igrs:serviceAttribute name="ConnectionIdList" type="Type_ConnectionIdList"
notifiable="true"/>
            <igrs:serviceAttribute name="ProtocolInfoList" type="Type_ProtocolInfoList"
notifiable="true"/>
            <igrs:serviceAttribute name="MediaFormatList" type="Type_MediaFormatList"
notifiable="true"/>
            <igrs:serviceAttribute name="IPList" type="Type_IPList" notifiable="true"/>

        </portType>
        <binding name="ConnectionManagementServiceIGRSPipe"
type="igrs:ConnectionManagementPortType">
            <operation name="GetProtocolInfo">
```

```
                </operation>
                <operation name="PrepareForConnection">


                </operation>
                <operation name="ReleaseConnection">


                </operation>
                <operation name="GetActiveConnectionIdList">


                </operation>
                <operation name="GetCurrentConnectionInfo">


                </operation>
                <operation name="SubscribeServiceAttribute">


                </operation>
                <operation name="UnsubscribeServiceAttribute">


                </operation>
        </binding>
        <service name="ConnectionManagementService">
                <port                                name="ConnectionManagementService"
binding="igrs:ConnectionManagementServiceIGRSPipe">
                </port>
        </service>

        <types>
                <element name="Type_TransportState" type="TransprotStateType"/>
                <element name="Type_StorageMediaName" type="StorageMediaNameType"/>
                <element name="Type_PlayMode" type="PlayModeType"/>
                <element name="Type_PlaySpeed" type="PlaySpeedType"/>
                <element name="Type_TrackNumber" type="xsd:unsignedInt"/>
                <element name="Type_MediaTimeLength" type="xsd:string"/>
                <element name="Type_TransportURIList" type="TransportURIListType"/>
                <element name="Type_TransportURI" type="xsd:string"/>
                <element name="Type_ControlURI" type="xsd:string" minOccurs="0"/>
                <element name="Type_Item" type="xsd:string" minOccurs="0"/>
                <element name="Type_ItemList" type="ItemListType" minOccurs="0"/>
                <element name="Type_SeekMode" type="SeekModeType"/>
                <element name="Type_SeekTargetPosition" type="xsd:string"/>
                <element name="Type_TransportInstanceId" type="xsd:unsignedInt"/>
                <element                                    name="Type_ServiceAttributeName"
type="ServiceAttributeNameType" minOccurs="0"/>
                <element name="Type_SubscriptionId" type="xsd:string" minOccurs="0"/>
                <element name="Type_Count" type="xsd:unsignedInt" minOccurs="0"/>
                <element name="Type_TimeLength" type="xsd:unsignedInt" minOccurs="0"/>
            <element          name="Type_TimeShiftSwitch"          type="TimeShiftSwitchType"
minOccurs="0"/>

      <complexType name="TimeShiftSwitchType">
                        <choice>
                                <element name="OPEN" type="xsd:string"/>
                                <element name="CLOSE" type="xsd:string"/>
                        </choice>
```

```
</complexType>
<complexType name="TransprotStateType">
	<choice>
		<element name="PLAYING" type="xsd:string"/>
		<element name="PAUSED_PLAYBACK" type="xsd:string"/>
		<element name="PAUSED_RECORDING" type="xsd:string"/>
		<element name="STOPPED" type="xsd:string"/>
		<element name="RECORDING" type="xsd:string"/>
		<element name="TRANSITIONING" type="xsd:string"/>
		<element name="NO_MEDIA_PRESENT" type="xsd:string"/>
		<element name="COMPLETE" type="xsd:string"/>
		<element name="ERROR_OCCURRED" type="xsd:string"/>

	</choice>
</complexType>
<complexType name="StorageMediaNameType">
	<choice>
		<element name="UNKNOWN" type="xsd:string"/>
		<element name="DV" type="xsd:string"/>
		<element name="MINI-DV" type="xsd:string"/>
		<element name="VHS" type="xsd:string"/>
		<element name="W-VHS" type="xsd:string"/>
		<element name="S-VHS" type="xsd:string"/>
		<element name="D-VHS" type="xsd:string"/>
		<element name="VHSC" type="xsd:string"/>
		<element name="VIDEO8" type="xsd:string"/>

		<element name="HI8" type="xsd:string"/>
		<element name="CD-ROM" type="xsd:string"/>
		<element name="CD-DA" type="xsd:string"/>
		<element name="CD-R" type="xsd:string"/>
		<element name="CD-RW" type="xsd:string"/>
		<element name="VIDEO-CD" type="xsd:string"/>
		<element name="SACD" type="xsd:string"/>
		<element name="MD-AUDIO" type="xsd:string"/>
		<element name="MD-PICTURE" type="xsd:string"/>
		<element name="DVD-ROM" type="xsd:string"/>
		<element name="DVD-VIDEO" type="xsd:string"/>
		<element name="DVD-R" type="xsd:string"/>
		<element name="DVD+RW" type="xsd:string"/>
		<element name="DVD-RW" type="xsd:string"/>
	</choice>
</complexType>
<complexType name="PlayModeType">
	<choice>
		<element name="NORMAL" type="xsd:string"/>
		<element name="SHUFFLE" type="xsd:string"/>
		<element name="REPEAT_ONE" type="xsd:string"/>
		<element name="REPEAT_ALL" type="xsd:string"/>
		<element name="RANDOM" type="xsd:string"/>
		<element name="DIRECT_1" type="xsd:string"/>
		<element name="INTRO" type="xsd:string"/>
	</choice>
</complexType>
<complexType name="PlaySpeedType">
	<choice>
		<element name="NORMAL" type="xsd:string"/>
		<element name="FASTFORWARD" type="xsd:string"/>
		<element name="SLOWFORWARD" type="xsd:string"/>
		<element name="FASTBACKWARD" type="xsd:string"/>
	</choice>
</complexType>
```

```
            <complextype name="TransportURIListType">
               <sequence>
                  <element name="TransportURI" type="xsd:string" minOccurs="1"
maxOccurs="unbounded"/>
               </sequence>
            </complextype>
            <complextype name="ItemListType">
               <sequence>
                  <element name="Item" type="ItemType" minOccurs="1"/>
               </sequence>
            </complextype>
            <complexType name="ItemType">
                  <sequence>
                        <element name="ItemAttribute" type="ItemAttributeType"/>
                        <element name="ItemReferences" minOccurs="0">
                              <complexType>
                                    <sequence>
                                          <element name="KeyFrame"
minOccurs="0">
                                                <attribute name="Id"
type="igrs:Type_ObjectId"/>
                                          </element>
                                          <element name="MicroPhotograph"
minOccurs="0">
                                                <attribute name="Id"
type="igrs:Type_ObjectId"/>
                                          </element>
                                    </sequence>
                              </complexType>
                        </element>
                  </sequence>
            </complexType>
            <complexType name="ItemAttributeType">
                  <sequence>
                        <element name="ObjectId" type="xsd:string"/>
                        <element name="ParentId" type="xsd:string" minOccurs="0"/>
                        <element name="DeviceId" type="xsd:string" minOccurs="0"/>
                        <element name="DeviceName" type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectPath" type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectName" type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectStoreAttribute">
                              <complexType>
                                    <sequence>
                                          <element name="ReadOnly"
type="xsd:string" minOccurs="0"/>
                                          <element name="Hide" type="xsd:string"
minOccurs="0"/>
                                          <element name="Document"
type="xsd:string" minOccurs="0"/>
                                          <element name="System"
type="xsd:string" minOccurs="0"/>
                                    </sequence>
                              </complexType>
                        </element>
                        <element name="CreateTime" type="xsd:string"
minOccurs="0"/>
                        <element name="LastAccessTime" type="xsd:string"
minOccurs="0"/>
```

```
                        <element name="LastWriteTime" type="xsd:string"
minOccurs="0"/>
                        <element name="ObjectType" type="igrs:ObjectTypeType"
minOccurs="0"/>
                        <element name="Size" type="xsd:string" minOccurs="0"/>
                        <element name="Duration" type="xsd:string" minOccurs="0"/>
                        <element name="Encrypted" type="xsd:string" minOccurs="0"/>
                    <element name="DRMContendId" type="xsd:string" minOccurs="0"/>
                        <element name="Type_MediaFormat">
                            <complexType>
                                <sequence>
                                    <element name="AudioFormat"
type="xsd:string" minOccurs="0"/>
                                    <element name="VideoFormat"
type="xsd:string" minOccurs="0"/>
                                    <element name="PhotoFormat"
type="xsd:string" minOccurs="0"/>
                                    <element name="RemoteDesktop"
type="RemoteDesktop" minOccurs="0"/>
                                    <element name="VideoCamera"
type="VideoCamera" minOccurs="0"/>
                                </sequence>
                                <attribute name="Name" type="xsd:string"/>
                                <attribute name="Type" type="xsd:string"/>
                            </complexType>
                        </element>
                        <element name="ObjectURI" type="xsd:string"/>
                        <element name="Type_ConnectionMangementServiceId"
type="xsd:unsignedInt"/>
                        <element name="Choices" minOccurs="0">
                            <complexType>
                                <sequence>
                                    <element name="Choice" minOccurs="0"
maxOccurs="unbounded">
                                        <complexType>
                                            <sequence>
                                                <element
name="Selection" minOccurs="0" maxOccurs="unbounded">
        <complexType>
        <attribute name="Id" type="xsd:string"/>
        <attribute name="Name" type="xsd:string"/>
        </complexType>
                                                </element>
                                            </sequence>
                                            <attribute name="Id"
type="xsd:string"/>
                                            <attribute name="Name"
type="xsd:string"/>
                                            <attribute name="Default"
type="xsd:string"/>
                                            <attribute name="Type"
type="xsd:string"/>
                                        </complexType>
                                    </element>
                                </sequence>
                            </complexType>
                        </element>
                    </sequence>
```

```
            </complexType>
            <complexType name="ObjectTypeType">
                  <choice>
                        <element name="Audio" type="igrs:AudioType"
minOccurs="0"/>
                        <element name="Video" type="xsd:string" minOccurs="0"/>
                        <element name="Photo" type="xsd:string" minOccurs="0"/>
                        <element name="Doc" type="xsd:string" minOccurs="0"/>
                        <element name="Camera" type="xsd:string" minOccurs="0"/>
                        <element name="Screen" type="xsd:string" minOccurs="0"/>
                  </choice>
            </complexType>
            <complexType name="AudioType">
                  <sequence>
                        <element name="Singer" type="xsd:string" minOccurs="0"/>
                        <element name="Genre" type="xsd:string" minOccurs="0"/>
                        <element name="MusicDisc" type="xsd:string" minOccurs="0"/>
                        <element name="Author" type="xsd:string" minOccurs="0"/>
                  </sequence>
            </complexType>
            <complexType name="SeekModeType">
                  <choice>
                        <element name="TRACK_NR" type="xsd:string"/>
                        <element name="TAPE_INDEX" type="xsd:string"/>
                        <element name="ABS_COUNT" type="xsd:string"/>
                        <element name="REL_COUNT" type="xsd:string"/>
                        <element name="ABS_TIME" type="xsd:string"/>
                        <element name="REL_TIME" type="xsd:string"/>
                        <element name="FRAME" type="xsd:string"/>
                  </choice>
            </complexType>
            <complexType name="ServiceAttributeNameType">
                  <choice>
                        <element name="TRANSPORTSTATE" type="xsd:string"/>
                        <element name="CURRENTSPEED" type="xsd:string"/>
                        <element name="TRANSPROTURILIST" type="xsd:string"/>
                        <element name="PLAYMODE" type="xsd:string"/>
                  </choice>
            </complexType>
      </types>

      <message name="SetTransportURIListRequest">
            <part name="InstanceId" type="Type_TransportInstanceId"/>
            <part name="TransportURIList" type="igrs:Type_TransportURIList"/>
            <part name="ItemList" type="Type_ItemList"/>
      </message>
      <message name="SetTransportURIListResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
      </message>
      <message name="GetTransportInfoRequest">
            <part name="InstanceId" type="Type_TransportInstanceId"/>
      </message>
      <message name="GetTransportInfoResponse">
            <part name="ReturnCode" type="xsd:unsignedInt"/>
            <part name="CurrentTransportState" type="Type_TransportState"/>
            <part name="CurrentSpeed" type="Type_PlaySpeed"/>
      </message>
      <message name="PlayRequest">
            <part name="InstanceId" type="Type_TransportInstanceId"/>
            <part name="Speed" type="Type_PlaySpeed"/>
            <part name="Offset" type="Type_Count"/>
      </message>
```