ISO/IEC 14543-5-22

Edition 1.0 2010-02

INTERNATIONAL STANDARD

Information technology – Home electronic system (HES) architecture – Part 5-22: Intelligent grouping and resource sharing for HES Class 2 and Class 3 – Application profile – File profile

ISO/IEC 14543-5-22:2010-02(en)

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 It also gives information on projects, replaced and withdrawn publications.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 14543-5-22

Edition 1.0 2010-02

INTERNATIONAL STANDARD

Information technology – Home electronic system (HES) architecture – Part 5-22: Intelligent grouping and resource sharing for HES Class 2 and Class 3 – Application profile – File profile

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 35.240.67 ISBN 2-8318-1076-1

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INFORMATION TECHNOLOGY –
HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 5-22: Intelligent grouping and resource sharing for HES Class 2 and Class 3 – Application profile – File profile

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 14543-5-22 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 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 seven parts:

- **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.
- **Part 5-21: Application profile – AV profile**
  - Based on the IGRS Core Protocol.
  - Defines a device and service interaction mechanism, as well as application interfaces used in IGRS Basic Applications.
- **Part 5-22: Application profile – File profile**
  - Based on the IGRS Core Protocol.
  - Defines a device and service interaction mechanism, as well as application interfaces used in IGRS Basic Applications.
- **Part 5-3: Basic application**
  - 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.
- **Part 5-4: Device validation**
  - Defines a standard method to validate an IGRS-compliant device.
- **Part 5-5: Device types**
  - Defines IGRS Device types used in IGRS applications.
- **Part 5-6: Service types**
  - Defines basic service types used in IGRS applications.

INFORMATION TECHNOLOGY – HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 5-22: Intelligent grouping and resource sharing for HES Class 2 and Class 3 – Application profile – File profile

1 Scope

This part of ISO/IEC 14543 specifies the file data streaming application profile, device interaction flow model, the request and response messages in the device interaction process, and the service description format of the devices based on Intelligent Grouping and Resource Sharing (IGRS), ISO/IEC 14543-5-1.

This standard is applicable to resource sharing and service collaboration of file data stream 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 14543-5-1, *Information technology – Home electronic system (HES) architecture – Part 5-1: Intelligent grouping and resource sharing for Class 2 and Class 3 – Core protocol*

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

3 Abbreviations

For the purposes of this document the following acronyms and abbreviations apply.

FAMS	FileAccessManagement Service
FC	FileClient
FCMS	FileConnectionManagement Service
FS	FileServer
IGRS	Intelligent grouping and resource sharing

4 Conformance

In order to conform to this International Standard the following applies.

- The IGRS File profile interaction model shall conform to the system specification described in Clause 5.

- File service interaction flow models shall conform to the specifications described in Clause 6.
- The relevant File device (File Server, File Client) and service (FAMS, FCMS) shall conform to the specification defined in Clause 7 and Clause 8 respectively.
- The file invocation session setup procedure and message formats shall conform to specifications defined in Clause 9 and Clause 10.

5 System

5.1 Application scenario

In small office and home environment, there are many kinds of digital devices used, such as PC, Notebook, PDA, DC, DV, MP3, MP4, mobile phone, Set-Top-Box, etc. Typically, the user may need to browse, copy, delete, read and write data files stored in these devices over the network.

File profile is designed specifically for this objective. It sets up a middleware layer between the users and different devices. This layer hides the underlying networking protocols, devices, operating systems and file operation modes while providing a set of commonly shared file access APIs for the application program developers. The user can access the files in devices transparently in a network environment, without having to consider the devices and operating systems.

The file profile enables the client to access shared files on servers from different locations, and it includes the following functions:

a) retrieve client file access identity verification;

b) browse shared file/directory on server;

c) manage uploading/downloading of file/directory;

d) support simple management of file/directory in the shared directory of FileServer;

e) support file/directory update event subscription on FileServer;

f) support service update event subscription;

g) out-of-band transport negotiation when uploading/downloading file.

5.2 Design criteria

5.2.1 Relationship between file profile and IGRS

File profile includes two parts: one specifies the functional interfaces of IGRS FileAccessManagement service and FileConnectionManagement service; and the other specifies the interaction logical flows of these service functional interfaces.

5.2.2 Relationship between file profile and transport protocols

File profile is a higher layer application profile in the IGRS framework and does not rely on specific transport protocols. Therefore it shall support any transport protocols such as FTP, HTTP, etc.

5.2.3 Relationship between file profile and existing file sharing system

File profile defines a set of standardized file sharing access interface based on IGRS network. It does not put any restrictions on the backend implementation. File profile shall support either private systems developed by users, or existing file sharing systems such as ActiveDirectory, SMB, NFS, etc.

5.2.4 Relationship between file profile and file format

File profile is independent of any specific file format, which means it shall support interactions with files of any arbitrary format.

5.2.5 Device supported by file profile

Theoretically, file profile can run on any IGRS devices, such as PC, Notebook, PDA, DC, DV, MP3, MP4, mobile phone, Set-Top-Box, etc.

5.3 Interaction model

IGRS file profile is composed of two logical devices, FileServer and FileClient. FileServer is a device that provides content, and it includes two services: FileAccessManagement Service (FAMS) and FileConnectionManagement Service (FCMS). FileClient is a device that accesses content, and it includes FileClient Engine. The user can access and manage the contents published by FileServer from FileClient. All interaction functions shall be completed though the services and engine running on these devices.

The interaction model covers two services, FileAccessManagement Service and FileConnectionManagement Service, both of which run on FileServer. FileAccessManagement Service is responsible for a majority of functions in the file interaction framework, including file browsing management, file uploading/downloading management, client access rights authentication, etc. FileConnectionManagement Service is used in out-of-band transport protocol negotiation between FileServer and FileClient. FileClient Engine is a client application program responsible for interactions with FileServer. An IGRS file profile interaction model is shown in Figure 1.

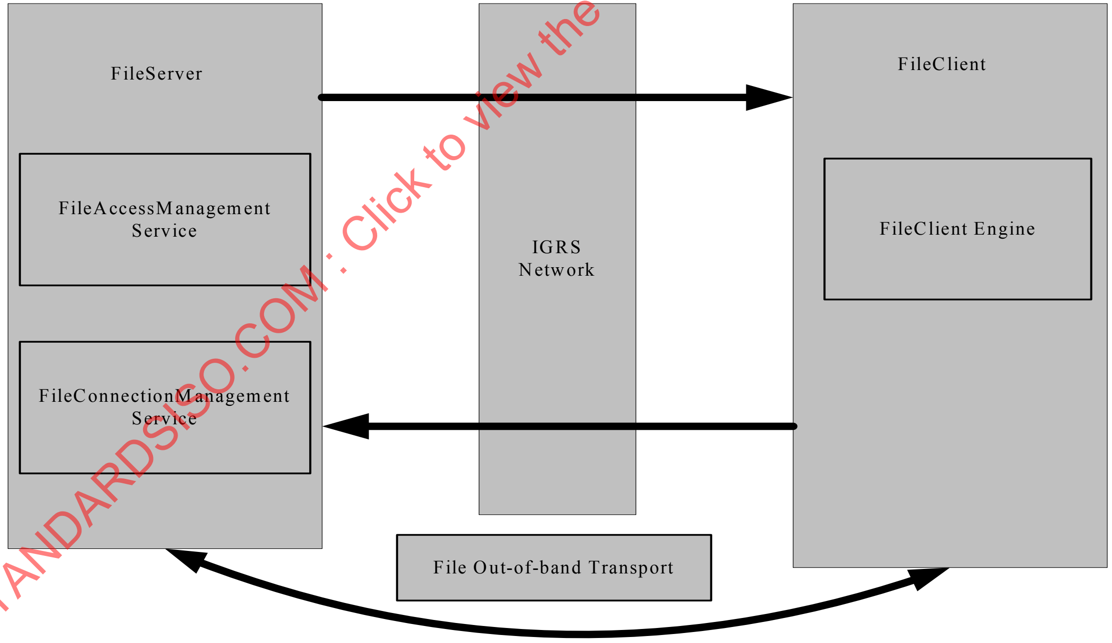
FileServer
FileAccessManagement Service
FileConnectionManagement Service
IGRS Network
FileClient
FileClient Engine
File Out-of-band Transport

Figure 1 – IGRS file profile interaction model

6 Interaction flow of IGRS file profile

6.1 Overview

Based on the basic rights setting of the shared files on FileServer, the interaction flow of IGRS file profile can be classified into two categories: simple interaction flow (see 6.2) and complete interaction flow (see 5.3). In simple interaction flow, only the required interfaces of

FileAccessManagement Service and FileConnectionManagement Service on FileServer are applicable in this process. FileClient can browse, download shared file and directory, or retrieve attributes of shared file and directory. In complete interaction flow, the applicable interfaces include all required interfaces of FileAccessManagement Service and FileConnectionManagement Service on FileServer and any optional interfaces except for those that relate to subscription. FileClient can perform management operations on shared files and directories, such as copy, move, delete, modify attributes, etc., or upload file to FileServer, in addition to browse, search, download shared file and directory, or retrieve attributes of shared file and directory. The detailed descriptions about services and interfaces in interaction flow can be found in Annex B.

6.2 Simple Interaction Flow

In simple interaction flow (see Figure 2), only the required interfaces of FileAccessManagement Service and FileConnectionManagement Service on FileServer are applicable. The FileClient shall access a shared file and directory in read-only mode, and is allowed to browse, download a shared file/directory or retrieve attributes of a shared file/directory. However, it shall not be allowed to modify a shared file/directory.

In a simple interaction flow, the invocation process of IGRS service interfaces between FileClient and FileServer is described as follows.

a) IGRS service discovery: through IGRS service discovery mechanism, FileClient discovers FileAccessManagement Service and FileConnectionManagement Service on FileServer.

b) Retrieve access right: a FileClient retrieves an authentication key to get an access right to a shared file on FileServer through invoking FAMS::GetAuthenticationKey() interface of FileAccessManagement Service on FileServer. FileClient can use any combination information of DeviceID/DeviceName, UserName/Password, and Third-party authentication mode of the local device, to retrieve authentication key from FileServer.

c) Browse a shared file/directory:

   1) Retrieve sorting capability of a FileServer: a FileClient can retrieve a shared file/directory sorting capability of a FileServer by invoking FAMS::GetSortCapability() interface of FileAccessManagement Service on the FileServer. For example, it can sort according to file names or the time of modification.

   2) Browse a shared file/directory on a FileServer: a FileClient can browse all the files and sub-directory information in any specified shared directory by invoking FAMS::Browse() interface of FileAccessManagement Service on FileServer.

   3) Retrieve attributes of a shared file/directory: a FileClient can retrieve attributes of any specified shared file/directory by invoking FAMS::GetAttribute() interface of FileAccessManagement Service on a FileServer.

d) Setup a connection with a FileServer:

   1) Retrieve transport protocols supported by a FileServer: a FileClient retrieves transport protocols supported by a FileServer, by invoking FCMS::GetProtocolInfo() interface of FileconnectionManagement Service on a FileServer.

   2) Select matching transport protocols: the use of retrieved transport protocols supported by a FileServer according to 1) in step d), a FileClient selects matching transport protocols supported by the device.

   3) Connection preparation: a FileClient notifies a FileServer to prepare for a connection setup, and retrieve the connection identifier for connection management to be used in a subsequent interaction process, by invoking FCMS::PrepareforConnection() interface of a FileConnectionManagement Service on a FileServer.

e) Download a shared file/directory from a FileServer: a FileClient can set up the downloading of a shared file/directory, and retrieve the URI list of a downloaded shared file/directory from a FileServer, by invoking FAMS::PrepareforDownload() interface of a FileAccessManagement Service on a FileServer.

f) File/directory transport: a FileClient and FileServer use out-of-band transport protocols and the URI retrieved in step e) to transport data. When the specified file/directory has been transported, it goes to back to step e) to transport another file/directory if required.

g) Close connection and release resource: when the file transport has been completed, the FileClient notifies the FileServer to close the connection between them and release the resource, by invoking FCMS::ReleaseConnection() interface of FileConnection-Management Service on FileServer.

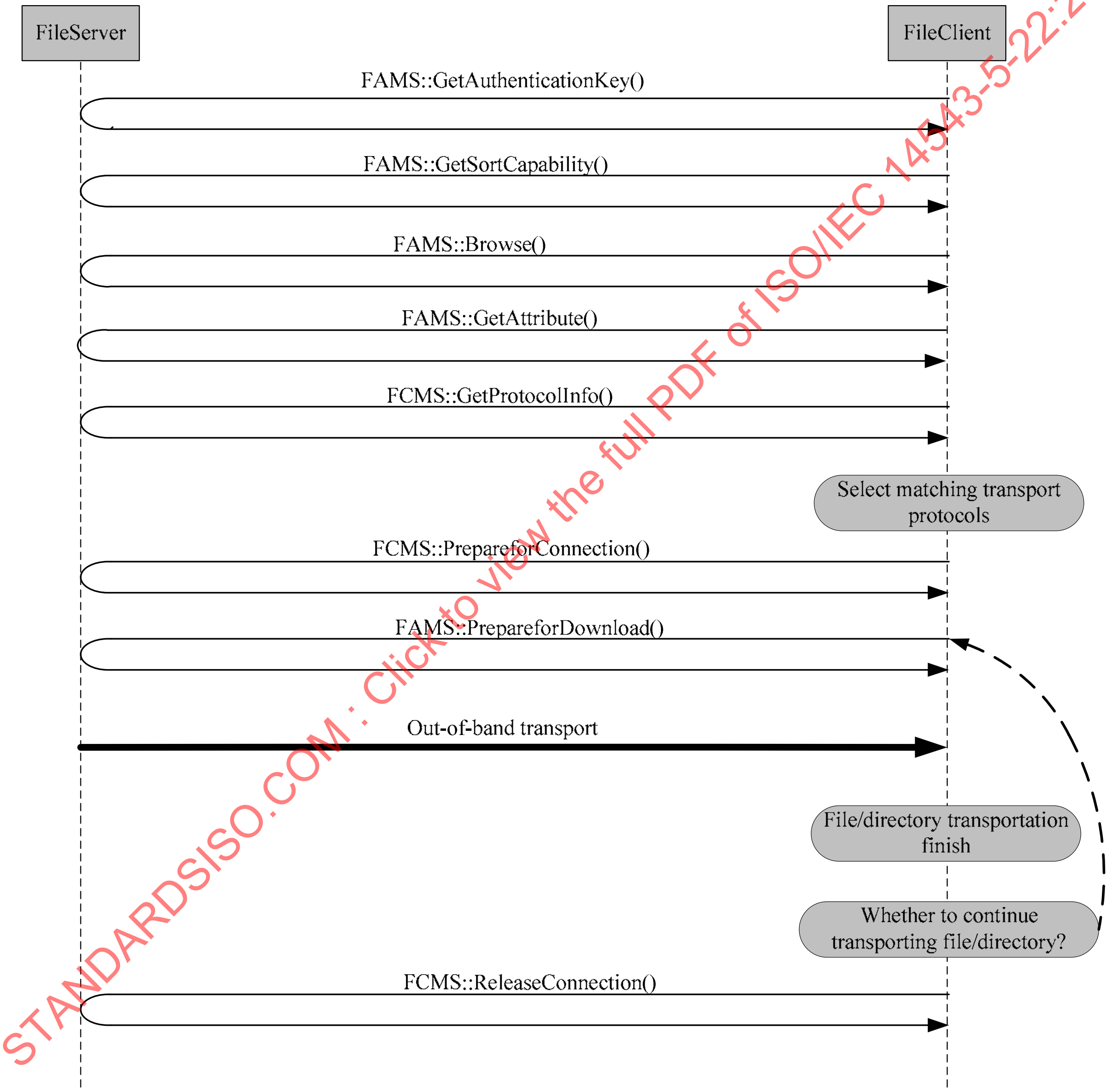
FileServer
FileClient
FAMS::GetAuthenticationKey()
FAMS::GetSortCapability()
FAMS::Browse()
FAMS::GetAttribute()
FCMS::GetProtocolInfo()
Select matching transport protocols
FCMS::PrepareforConnection()
FAMS::PrepareforDownload()
Out-of-band transport
File/directory transportation finish
Whether to continue transporting file/directory?
FCMS::ReleaseConnection()

Figure 2 – Simple interaction flow model

6.3 Complete interaction flow

In a complete interaction flow (see Figure 3), the applicable interfaces include all required interfaces of the FileAccessManagement Service and FileConnectionManagement Service on a FileServer and any optional interfaces except for those that relate to subscription. A FileClient can perform management operations on shared files and directories, such as copy, move, delete, modify attributes, etc., or upload a file to a FileServer, in addition to browse, search and download a shared file and directory, or retrieve attributes of a shared file and directory.

In a complete interaction flow, the invocation process of IGRS service interfaces between a FileClient and a FileServer is described as follows.

a) IGRS service discovery: through the IGRS service discovery mechanism, a FileClient discovers the FileAccessManagement Service and FileConnectionManagement Service on the FileServer.

b) Retrieve access right: a FileClient retrieves an authentication key to get an access right to a shared file on a FileServer by invoking FAMS::GetAuthenticationKey() interface of the FileAccessManagement Service on the FileServer. The FileClient can use any combination information of the DeviceID/DeviceName, UserName/Password, and the Third-party authentication mode of the local device, to retrieve the authentication key from the FileServer.

c) Browse and search a shared file/directory on a FileServer:

    1) Retrieve the sorting and searching capability of a FileServer: the FileClient can retrieve a shared file/directory sorting capability of a FileServer by invoking FAMS::GetSortCapability() interface of a FileAccessManagement Service on a FileServer. For example, it can sort according to file names or times of modification. It can also retrieve a shared file/directory searching capability of a FileServer through invoking FAMS::GetSearchCapability () interface of a FileAccessManagement Service on a FileServer. For example, it can search according to file names or the time of modification.
    2) Retrieve or set a browsing filter of a shared file/directory: the FileClient can retrieve or set a browsing file of a shared file/directory through invoking FAMS::GetBrowseFilter() or FAMS::SetBrowseFilter() interface of FileAccessManagement Service on FileServer.
    3) Browse or search a shared file/directory on a FileServer: the FileClient can browse all files and sub-directory information in any specified shared directory by invoking FAMS::Browse() interface of FileAccessManagement Service on FileServer. It can also search a shared file/directory according to designated criteria through invoking FAMS::Search() interface of a FileAccessManagement Service on a FileServer.
    4) Retrieve or set attributes of a shared file/directory: the FileClient can retrieve attributes of a specified shared file/directory by invoking FAMS::GetAttribute() interface of a FileAccessManagement Service on a FileServer. It can also set attributes of a specified shared file/directory through invoking FAMS::SetAttribute() interface of FileAccessManagement Service on FileServer.

d) Manage a shared file/directory: the FileClient can create a new shared file/directory in a specified shared directory on a FileServer by invoking FAMS::New() interface of a FileAccessManagement Service on a FileServer; or delete a shared file/directory on the FileServer, by invoking FAMS::Delete() interface of FileAccessManagement Service; or copy a shared file/directory on a FileServer to a specified shared directory through invoking FAMS::Copy() interface of a FileAccessManagement Service; or move a shared file/directory to a specified shared directory through invoking FAMS::Move() interface of a FileAccessManagement Service.

e) Set up a connection with a FileServer:

    1) Retrieve the transport protocols supported by a FileServer: the FileClient retrieves transport protocols supported by a FileServer, through invoking FCMS::GetProtocolInfo() interface of a FileConnectionManagement Service on a FileServer.
    2) Select matching transport protocols: the use of retrieved transport protocols supported by a FileServer according to 1) in step e), the FileClient selects matching transport protocols supported by the device.
    3) Connection preparation: the FileClient notifies a FileServer to prepare for connection setup, and retrieve connection identifier for the connection management to use in a subsequent interaction process by invoking

FCMS::PrepareforConnection() interface of a FileConnectionManagement Service in a FileServer.

f) Download a shared file/directory from a FileServer or upload a shared file/directory to a FileServer: the FileClient can set up the downloading of a shared file/directory, and retrieve the URI list of a shared file/directory to be downloaded through invoking FAMS::PrepareforDownload() interface of FileAccessManagement Service on FileServer; or set up a file/directory uploading of a shared directory, and retrieve the URI of the shared directory through invoking FAMS::PrepareforUpload() interface of FileAccess-Management Service.

g) File transport: a FileClient and FileServer use out-of-band transport protocols and the URI retrieved in step f) to download shared files from a FileServer or upload a client file to the designated shared directory on a FileServer. After the transport of a specified file has been completed, it returns to step f), if required.

h) Close connection and release resource: When the file transport has been completed, the FileClient notifies a FileServer to close the connection between them and release the resource through invoking FCMS::ReleaseConnection() interface of FileConnection-Management Service on FileServer.

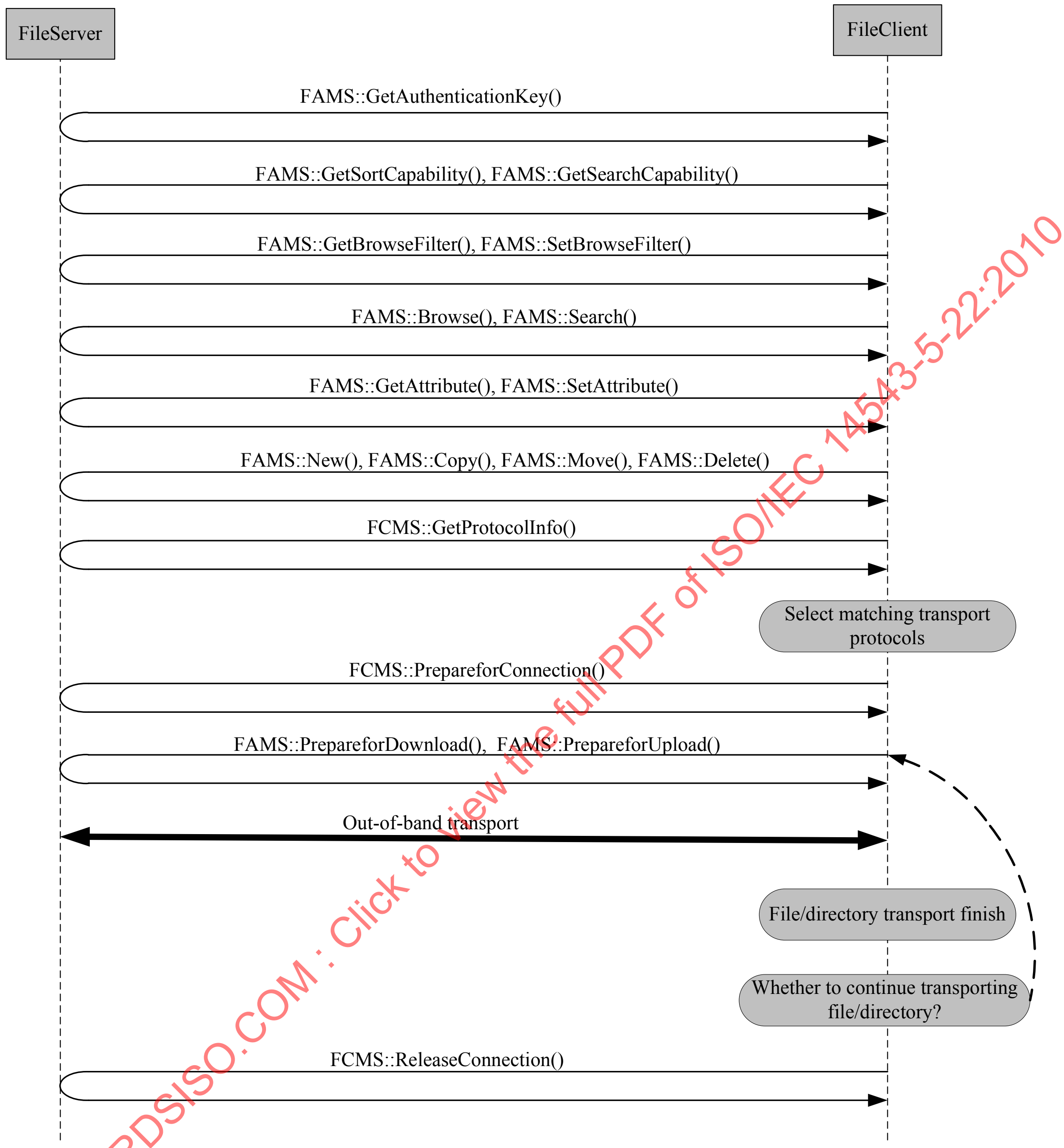
FileServer
FileClient
FAMS::GetAuthenticationKey()
FAMS::GetSortCapability(), FAMS::GetSearchCapability()
FAMS::GetBrowseFilter(), FAMS::SetBrowseFilter()
FAMS::Browse(), FAMS::Search()
FAMS::GetAttribute(), FAMS::SetAttribute()
FAMS::New(), FAMS::Copy(), FAMS::Move(), FAMS::Delete()
FCMS::GetProtocolInfo()
Select matching transport protocols
FCMS::PrepareforConnection()
FAMS::PrepareforDownload(), FAMS::PrepareforUpload()
Out-of-band transport
File/directory transport finish
Whether to continue transporting file/directory?
FCMS::ReleaseConnection()

Figure 3 – Complete Interaction Flow of Shared File

7 FileServer

7.1 General

FileServer is the device that provides the file content, and it includes the FileAccess-Management Service (FAMS) and FileConnectionManagement Service (FCMS).

The functions of a FileServer include the following.

a) Based on the authentication information provided by FileClient, a FileServer assigns a corresponding authentication key to a FileClient (recommended validity only during the session). This key holds FileClient access rights in such a way that it can be used as the identifier for subsequent file access by the FileClient.

b) It stores the actual file/directory and supports a FileClient to browse and search for the relevant content.

c) It supports simple management of a file/directory in a shared directory on a FileServer.

d) It manages the uploading/downloading of a file/directory.

e) It supports a FileServer file/directory update event subscription.

f) It supports a service update event subscription.

7.2 FileAccessManagement Service

7.2.1 General

The FileAccessManagement Service provides the following functions to the FileClient.

a) It provides authentication for the FileClient, and thus gives corresponding file access rights to the FileClient.

b) It allows the FileClient to retrieve a sorting/searching capability supported by a FileServer.

c) It allows the FileClient to browse the content directory in the network provided by the FileServer.

d) It allows the FileClient to search for a specified file/directory.

e) It allows the FileClient to retrieve and modify attributes of a file/directory.

f) It allows the FileClient to retrieve and set a browsing filter.

g) It allows the FileClient to subscribe to a file/directory object update event.

h) It allows the FileClient to subscribe to a FileAccessManagement Service update event.

i) It supports the FileClient to upload and download the specified file/directory.

The detailed description of this service is found in Annex C.

7.2.2 FileAccessManagement Service Type

FileAccessManagement Service Type is defined as:

urn:IGRS:Service:ServiceType:FileAccessManagement:1.

7.2.3 FileAccessManagement Service Attribute

Table 1 contains the File Access Management Service Attribute List.

Table 1 – FileAccessManagement Service Attribute List

Service attribute name	Data type	Field explanation
SortCaps	Type_SortCapability	File/directory object attribute sets sorted by searching or browsing results. An empty set denotes that the FileAccessManagement Service does not support any file/directory object attributes for sorting. If the returned set isn’t empty, then any element in the set can be used as the sorting rule provided by FileAccessManagement Service.
SearchCaps	Type_SearchCapability	File/directory object attribute sets that set searching rules. An empty set denotes that the FileAccessManagement Service does not support any file/directory object attributes for sorting. If the returned set is not empty, then any element in the set can be used as the searching rule provided by the FileAccessManagement Service.

7.2.4 Data Types of FileAccessManagement Service

Table 2 contains the File Access Management Service Data Type.

Table 2 – FileAccessManagement Service Data Type

Data type name	Data type	Field explanation
Type_UserAuthenticationInfo	String	Authentication information provided by the FileClient
Type_AuthenticationKey	String	Authentication key provided to the FileClient by the FileAccessManagement Service. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
Type_InstanceId	UnsignedInt	The session identifier between IGRS client and IGRS Service.
Type_ObjectId	String	Only Object identifier that identifies file/directory object. See detailed definition of Type_ObjectId in Annex A.1.
Type_SubscriptionId	String	Subscription identifier.
Type_ObjectIdList	String	Object identifier list.
Type_SortCapability	String	File/directory object attribute sets sorted by searching or browsing results. An empty set denotes that the FileAccessManagement Service does not support any file/directory object attributes for sorting. If the returned set is not empty, then any element in the set can be used as the sorting rule provided by the FileAccessManagement Service.
Type_SearchCapability	String	File/directory object attribute sets that set searching rules. An empty set denotes that the FileAccessManagement Service does not support any file/directory object attributes for sorting. If the returned set is not empty, then any element in the set can be used as the searching rule provided by the FileAccessManagement Service.
Type_FilterRule	String	Represents a searching rule when searching for a file/directory object. See detailed definition of Type_FilterRule in Clause A.3.
Type_SortRule	String	Represents a sorting rule when sorting a file/directory attribute list. See detailed definition of Type_SortRule in Clause A.4.
Type_ObjectAttribute	String	File/directory object attribute set, which includes ObjectType element to distinguish between file and directory object. See detailed definition of ObjectType in Clause A.2.
Type_ServiceAttributeName	String	Service attribute name.
Type_DeleteMode	String	Mode to delete file/directory (temporary/permanent delete).
Type_ObjectList	String	Result list of file/directory object.
Type_Count	Int	Object number
Type_ObjectURI	String	URI of Object on FileServer.
Type_ObjectURITreeList	String	Tree list of the file/directory object list on FileServer, which includes file/directory object URI and attribute. See detailed definition in the description of Annex B.

7.2.5 Invocation interface of FileAccessManagement Service

7.2.5.1 GetAuthenticationKey

Function description: Retrieve the assigned authentication key of the FileServer based on authentication information. The authentication key specifies the access rights of the FileClient. The authentication key is recommended to be only in effect during a session. During the subsequent service invocations by the FileClient, the authentication key is required to be used as the input parameter.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
InstanceId	Input	Type_InstanceId	Session identifier between IGRS client and IGRS service.
UserAuthenticationInfo	Input	Type_UserAuthenticationInfo	Authentication information provided by the FileClient. See detailed definition in Annex B.
AuthenticationKey	Output	Type_AuthenticationKey	FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.

Return Value: Success: 0

Failure: E1, E2, E3 (see Error Codes in 7.2.6).

7.2.5.2 GetSortCapability

Function description: Retrieve file/directory sorting capability supported by FileAccessManagement Service on FileServer such that FileClient can use the file/directory attribute set as the sorting rule, such as name, time of creation, etc.

Input/Output parameter:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	FileAccessManagement Service assigns an authentication key to FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
SortCaps	Output	Type_SortCapability	File/directory object attribute sets sorted by searching or browsing results. An empty set denotes that the FileAccessManagement Service does not support any file/directory object attributes for sorting. If the returned set is not empty, then any element in the set can be used as the sorting rule provided by the FileAccessManagement Service.

Return Value: Success: 0

Failure: E1, E2, E3, E9, E10 (see Error Codes in 7.2.6).

7.2.5.3 GetSearchCapability

Function description: Retrieve the file/directory searching capability supported by the FileAccessManagement Service on the FileServer so that the FileClient can use the file/directory attribute set as searching rule, such as name, time of creation, etc.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
SearchCaps	Output	Type_SearchCapability	The File/directory object attribute sets that set searching rule. An empty set denotes that FileAccessManagement Service does not support any file/directory object attributes for searching. If the returned set isn't empty, then any element in the set can be used as the searching rule provided by FileAccessManagement Service.

Return Value: Success: 0

Failure: E1, E2, E3, E9, E10 (see Error Codes in section 7.2.6)

7.2.5.4 Browse

Function description: Return to the lower level file/directory list of the specified directory on FileServer, and get back the basic attributes of each object.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectId	Input	Type_ObjectId	Only Object identifier to identify file/directory object.
BrowseFilter	Input	Type_FilterRule	Browsing rule of the file object in the current directory. This parameter value has two instances: If the BrowseFilter value is empty then the preset filtering rule is used during the browsing process. The setting of the preset filtering rule is carried out by SetBrowseFilter interface. If the BrowseFilter value is not empty then this value is used as the browsing filtering rule of the service invocation. This value will not affect the value of the preset filtering rule.
StartOffset	Input	Type_Count	Offset between the starting item of the returned result and the first item in all file/directory lists that complies with the browsing rule. This value shall start from 0.
RequestedCount	Input	Type_Count	Maximum number of browsing items returned. −1 denotes that all browsing results are returned.
SortRule	Input	Type_SortRule	Sorting rule of the browsed directory object. This attribute can be retrieved through GetSortCapability interface.
Result	Output	Type_ObjectList	Result list of browsed file/directory file object.
NumberReturned	Output	Type_Count	Optional. The number of directory and file objects returned from the browsing results.
NumberTotalMatched	Output	Type_Count	Total number of file and directory objects that complies with the browsing rule.

Return Value: Success: 0

Failure: E1, E2, E3, E6, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.5.5 GetAttribute

Function description: Retrieve attribute set of the specified file/directory object on FileServer.

Input/Output parameters:

Parameter	Input/Output	Data Type	Field Explanation
AuthenticationKey	Input	Type_AuthenticationKey	FileAccessManagement Service assigns authentication key to FileClient. The authentication key specifies the access rights of FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectId	Input	Type_ObjectId	Only Object identifier to identify file/directory object.
ObjectAttribute	Output	Type_ObjectAttribute	Attribute set of file/directory object. See detailed definition in Annex B.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10 (see Error Codes in section 7.2.6)

7.2.5.6 SetAttribute

Function description: Set attribute of the specified file/directory object on FileServer.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectId	Input	Type_ObjectId	The only Object identifier to identify file/directory object.
ObjectAttribute	Input	Type_ObjectAttribute	The attribute of an object to be set. Only the setting of an object name and initial access rights are required to be supported. See detailed definition in Annex B.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.5.7 Search

Function description: Search for the specified file/directory object on the FileServer. That is to search for all of the object information in a specified directory (the recursive search of the object information in a sub-directory is included), and to support searching in multiple parallel directories.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectIdList	Input	Type_ObjectIdList	Object identifier list within the search scope.
SearchRule	Input	Type_FilterRule	Searching rule of the file/directory object.
StartOffset	Input	Type_Count	Offset between the starting item of the returned result and the first item in all file/directory lists that complies with the searching rule. This value shall start from 0.
RequestedCount	Input	Type_Count	Maximum number of searching items returned. −1 denotes that all searching results are returned.
Result	Output	Type_ObjectList	Result of file/directory object list returned according to the searching rule.
NumberReturned	Output	Type_Count	The number of file/directory objects returned from searching results.
NumberTotalMatched	Output	Type_Count	Optional, total number of the file/directory objects that complies with the searching rule.

Return Value: Success: 0

Failure: E1, E2, E3, E6, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.5.8 GetBrowseFilter

Function description: Retrieve the current browsing filter rule. This is a preset browsing filter rule used by the FileAccessManagement Service.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	FileAccessManagement Service assigns the authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
BrowseFilter	Output	Type_FilterRule	Current browsing filter rule used.

Return Value: Success: 0

Failure: E1, E2, E3, E9, E10 (see Error Codes in 7.2.6).

7.2.5.9 SetBrowseFilter

Function description: Set the browsing filter rule. Only the SetBrowseFilter interface can change the preset browsing filter rule, and the preset browsing filter rule shall remain the same until the next invocation of the SetBrowseFilter interface.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	FileAccessManagement Service assigns authentication key to FileClient. The authentication key specifies the access rights of FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
BrowseFilter	Input	Type_FilterRule	Browsing filter rule to be set.

Return Value: Success: 0

Failure: E1, E2, E3, E9, E10 (see Error Codes in 7.2.6).

7.2.5.10 New

Function description: Create a new file/directory object in the directory object specified by FileServer. The attribute of the object is specified by parameter ObjectAttribute.

Input/Output parameters:

Parameter	Input/Output	Data Type	Field Explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns the authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ParentId	Input	Type_ObjectId	The object identifier of the parent directory object in which the file will be created.
ObjectAttribute	Input	Type_ObjectAttribute	The attribute of the new object, including object name and the initial access rights.
ObjectId	Output	Type_ObjectId	The only object identifier of the new object assigned by FileAccessManagement Service.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10, E11 (see Error Codes in 7.2.6).

7.2.5.11 Copy

Function description: Copy the specified file/directory object to the specified target path, while reserving the source object. The file directory structure of the new directory shall follow the tree structure of the source directory at the time of copying.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
SourceObjectId	Input	Type_ObjectId	The object identifier of the source object assigned by the FileAccessManagement Service.
DestParentId	Input	Type_ObjectId	The object identifier of the target directory.
DestObjectId	Output	Type_ObjectId	The object identifier of the new object assigned by FileAccessManagement Service.
DestObjectAttribute	Output	Type_ObjectAttribute	The object attribute of the new object.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10, E11 (see Error Codes in 7.2.6).

7.2.5.12 Move

Function description: Copy the specified file/directory to the specified target destination, and delete the source object. The tree structure of the directory shall remain the same at the time of moving.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
SourceObjectId	Input	Type_ObjectId	The object identifier of the source object assigned by the FileAccessManagement Service.
DestParentId	Input	Type_ObjectId	The object identifier of the target directory.
DestObjectId	Output	Type_ObjectId	The object identifier of the new object assigned by FileAccessManagement Service.
DestObjectAttribute	Output	Type_ObjectAttribute	The object attribute of the new object.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10, E11 (see Error Codes in 7.2.6).

7.2.5.13 Delete

Function description: Delete the specified file/directory object on FileServer, including the sub-directories and files of this directory.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectId	Input	Type_ObjectId	The object identifier of the object to be deleted.
DeleteMode	Input	Type_DeleteMode	The mode of object deletion (temporary/permanent delete).

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10, E11 (see Error Codes in 7.2.6).

7.2.5.14 PrepareforDownload

Function description: Prepare for file download, and pre-detect the feasibility of the file/directory download. If download is allowed by the FileServer, then the URI of the source to be transported shall be returned to the client, so that the FileClient application can download the file/directory through an out-of-band data transport; or else corresponding error codes shall be returned.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
SourceObjectIdList	Input	Type_ObjectIdList	The object identifier list of the file/directory object to be downloaded.
SourceObjectURITreeList	Output	Type_ObjectURITreeList	The tree list of the downloaded file/directory object list on FileServer, including the URI and the attributes of the file/directory object.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.5.15 PrepareforUpload

Function description: Verify the file/directory objects to be uploaded by the client, including whether the storage space of the server is sufficient and has an object with the same object name, etc. If uploading is allowed by the FileServer, the URI of the target object to be transported shall be returned to the client, so that the FileClient application can upload the file/directory through out-of-band data transport; or else corresponding error codes shall be returned.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectAttribute	Input	Type_ObjectAttribute	The attribute of the uploaded file/directory object.
DestParentId	Input	Type_ObjectId	The parent directory of the uploading object on FileServer.
DestParentURI	Output	Type_ObjectURI	The URI of the object on FileServer. This parameter denotes the URI of the destination directory of the uploaded objects on the FileServer.

Return Value: Success: 0

Failure: E1, E2, E3, E7, E8, E9, E10, E11 (aee Error Codes in 7.2.6).

7.2.5.16 SubscribeObjectChange

Function description: Subscribe to the update event of file/directory objects on FileServer. If an object is changed, then FileAccessManagement Service shall notify FileClient of the event.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectId	Input	Type_ObjectId	The object identifier of the shared object to be subscribed.

Return Value: Success: 0

Failure: E1, E2, E3, E4, E5, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.5.17 UnsubscribeObjectChange

Function description: Unsubscribe to the update event of file/directory objects on FileServer.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ObjectId	Input	Type_ObjectId	The object identifier of the shared object to be unsubscribed.

Return Value: Success: 0

Failure: E1, E2, E3, E4, E5, E7, E9, E10 (see Error Codes in section 7.2.6).

7.2.5.18 SubscribeServiceAttribute

Function description: Subscribe to the update event of service attributes. If an attribute changes, the FileAccessManagement Service shall notify the FileClient.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of the FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
ServiceAttributeName	Input	Type_ServiceAttributeName	The name of the service attributes to be subscribed.
SubscriptionId	Output	Type_SubscriptionId	The subscription identifier.

Return Value: Success: 0

Failure: E1, E2, E3, E4, E5, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.5.19 UnsubscribeServiceAttribute

Function description: Unsubscribe to the update event of the service attribute.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
AuthenticationKey	Input	Type_AuthenticationKey	The FileAccessManagement Service assigns an authentication key to the FileClient. The authentication key specifies the access rights of a FileClient. It is recommended that the key is only valid during this session. It is disabled when the session is over.
SubscriptionId	Input	Type_SubscriptionId	The subscription identifier to be unsubscribed.

Return Value: Success: 0

Failure: E1, E2, E3, E4, E5, E7, E9, E10 (see Error Codes in 7.2.6).

7.2.6 FileAccessManagement Service error codes definition

Error code No.	Error code name	Value	Field explanation
0	RETURN_SUCCESS	0	Success
E1	RETURN_FAILED	1	Failure
E2	RETURN_INVALIDPARA	2	Invalid parameter
E3	RETURN_ERRORFORMATPARA	3	Parameter format error
E4	RETURN_INVALIDSUBSCRIBEID	4	Invalid subscription identifier
E5	RETURN_NOTALLOWEDTOSUBSCRIBE	5	Subscription not allowed
E6	RETURN_OFFSETOVERFLOW	6	Offset overflow
E7	RETURN_OBJECTINEXISTENCE	7	Object not exist
E8	RETURN_SPACENOTENOUGH	10	Not enough space
E9	RETURN_INVALIDAUTHENTICATIONKEY	11	Invalid authentication key
E10	RETURN_RIGHTNOTMATCHED	12	Rights not matched
E11	RETURN_OBJECTNAMEEXISTED	13	Object name already exist
E12	RETURN_INTERFACEINEXISTENCE	14	Interface not exist

7.3 FileConnectionManagement Service

7.3.1 General

The FileConnectionManagement Service is used to create and manage the connection between the FileServer and the FileClient. The FileServer can support and manage multiple active connections at any one time by the FileConnectionManagement Service. The detailed service description is shown in Annex D.

7.3.2 FileConnectionManagement Service Type

FileConnectionManagement Service Type is defined as:

urn:IGRS:Service:ServiceType:FileConnectionManagement:1

7.3.3 FileConnectionManagement Service Attribute

Table 3 contains the File Connection Management Service Attribute list.

Table 3 – FileConnectionManagement Service Attribute

Service attribute name	Data type	Field explanation
ProtocolInfoList	Type_ProtocolInfoList	The transport protocol list supported by FileServer.
IPList	Type_IPList	The valid IP address list of FileServer.

7.3.4 FileConnectionManagement Service Data Type

Table 4 contains the File Connection Management Service Data Type.

Table 4 – FileConnectionManagement Service Data Type

Data type name	Data type	Field explanation
Type_ProtocolInfo	String	Transport protocol.
Type_ProtocolInfoList	String	Transport protocol list supported by FileServer.
Type_IPList	String	The valid IP address list of FileServer.
Type_ConnectionId	UnsignedInt	The connection identifier managed by FileConnectionManagement Service on FileServer.
Type_SubscriptionId	UnsignedInt	The subscription identifier.
Type_ServiceAttributeName	String	The name of the service attribute to be subscribed.

7.3.5 Invocation Interface of FileConnectionManagement Service

7.3.5.1 GetProtocolInfo

Function description: Retrieve the transport protocol supported by the FileServer, including the transport control protocol name, network port, the content format and other protocol information, as well as the valid IP address list of the FileServer.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
ProtocolInfoList	Output	Type_ProtocolInfoList	The transport protocol list supported by the FileServer.
IP List	Output	Type_IPList	The valid IP address list of the FileServer. The format is shown in Annex B.

Return Value: Success: 0

Failure: E1, E2, E3, E6, E8, E9 (see Error Codes in 7.3.6).

7.3.5.2 PrepareforConnection

Function description: Retrieve the ConnectionId used to send and receive data.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
RemoteProtocolInfo	Input	Type_ProtocolInfo	The transport protocol.
ConnectionId	Output	Type_ConnectionId	The connection identifier managed by the FileConnectionManagement service on the FileServer.

Return Value: Success: 0

Failure: E1, E2, E3, E6, E7, E8, E9 (see Error Codes in 7.3.6).

7.3.5.3 ReleaseConnection

Function description: Notify the device to close the connection corresponding to the ConnectionId, and release resource.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
ConnectionId	Input	Type_ConnectionId	The connection identifier managed by FileConnectionManagement service on FileServer.

Return Value: Success: 0

Failure: E1, E2, E3, E6, E7 (see Error Codes in 7.3.6).

7.3.5.4 GetActiveConnectionIdList

Function description: Retrieve the current active connection identifier list on the FileServer.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
ConnectionIdList	Output	Type_ConnectionIdList	The current active connection identifier list on the current device.

Return Value: Success: 0

Failure: E1, E2, E3, E6, E7, E8, E9 (see Error Codes in 7.3.6).

7.3.5.5 GetCurrentConnectionInfo

Function description: Retrieve the connection information of the FileServer that corresponds to the connection identifier.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
ConnectionId	Input	Type_ConnectionId	The connection identifier on the current device.
ProtocolInfo	Output	Type_ProtocolInfo	The transport and control protocol used for file transport
ConnectionState	Output	Type_ConnectionState	The current connection state

Return Value: Success: 0

Failure: E1, E2, E3, E6, E7, E8, E9 (see Error Codes in 7.3.6).

7.3.5.6 SubscribeServiceAttribute

Function description: Subscribe to the update event of service attributes. If an attribute is changed, the FileConnectionManagement Service shall notify the FileClient.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
ServiceAttributeName	Input	Type_ServiceAttributeName	The name of the service attribute to be subscribed.
SubscriptionId	Output	Type_SubscriptionId	The subscription identifier

Return Value: Success: 0

Failure: E1, E2, E3, E4, E5, E6, E7, E8, E9 (see Error Codes in 7.3.6).

7.3.5.7 UnsubscribeServiceAttribute

Function description: Unsubscribe to the update event of file/directory objects on the FileServer.

Input/Output parameters:

Parameter	Input/Output	Data type	Field explanation
SubscriptionId	Input	Type_SubscriptionId	The subscription identifier of the object to be unsubscribed.

Return Value: Success: 0

Failure: E1, E3, E4, E5, E6, E7, E8, E9 (See Error Codes in section 7.3.6)

7.3.6 FileConnectionManagement Service error codes definition

Error code No.	Error code name	Value	Field explanation
0	RETURN_SUCCESS	0	Success
E1	RETURN_FAILED	1	Failure
E2	RETURN_INVALIDPARA	2	Invalid parameter
E3	RETURN_ERRORFORMATPARA	3	Parameter format error
E4	RETURN_INVALIDSUBSCRIBEID	4	Invalid subscription identifier
E5	RETURN_NOTALLOWEDTOSUBSCRIBE	5	Subscription not allowed
E6	RETURN_CONNECTIONDISABLED	8	Connection disabled
E7	RETURN_INVALIDCONNECTIONID	9	Invalid connection identifier
E8	RETURN_INVALIDAUTHENTICATIONKEY	11	Invalid Authentication Key
E9	RETURN_RIGHTNOTMATCHED	12	Rights not matched
E10	RETURN_INTERFACEINEXISTENCE	14	Interface not exist

8 FileClient

8.1 General

The FileClient can access the FileServer. After retrieving the authentication key from the FileServer, the FileClient can send service request messages to retrieve a FileAccessManagement Service function provided by the FileServer. The functions of the FileClient are as follows:

a) browse the shared file/directory provided by the FileServer;

b) manage the file/directory on the FileServer, including modify the file/directory attribute, create, copy and move file/directory, etc.;

c) upload and download files from the FileServer;

d) synchronize with the FileServer by subscribing to the service attributes update event on the FileServer.

8.2 FileClient Engine

The FileClient engine is an application program that is used to interact with the FileServer. It includes functions such as providing authentication information, user interface, invoking various interfaces on the FileServer to retrieve the functions provided by the FileServer etc.

9 Session

9.1 Session setup

9.1.1 General

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.

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.

9.1.2 Session setup condition

An IGRS Client can obtain the security mechanism description of this service through corresponding IGRS service advertisements or the detailed service description document. If an IGRS Client wants to invoke this service, it shall first request a session setup with this service according to the security mechanism description requirement.

9.1.3 Session setup process

A detailed session setup process is shown in 10.5 of ISO/IEC 14543-5-1.

9.2 Session termination

When an IGRS Client finishes invoking IGRS Services, the IGRS Client shall terminate the session with the IGRS Service (see 10.5 of ISO/IEC 14543-5-1).

10 Service invocation message format

10.1 General

The IGRS service invocation message format shall conform to the service invocation requirements specified in ISO/IEC 14543-5-1.

10.2 Service invocation request message

The Client can initiate “request/response” service invocation based on the session with the target service. The definition of the service invocation request message is shown in Table 5:

Table 5 – Service invocation request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP address:Port	Required field
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:InvokeServiceRequest	Required field, content shall be this
01-SourceDeviceId: requesting device identifier	Required field, and type is URI, defined in 8.1.2 of ISO/IEC 14543-5-1 Ed.1
01-TargetDeviceId: target device identifier	Required field, type is URI, defined in 8.1.2 of ISO/IEC 14543-5-1 Ed.1
Content-Length: Message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see B.1 of ISO/IEC 14543-5-1 Ed.1
MAN:”http://schemas.xmlsoap.org/soap/envelope/”;ns=02	Required field
02-SoapAction:”IGRS-InvokeService-Request”	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
<Session xmlns=”http://www.igrs.org/spec1.0”request=”NeedResponse”>	Required field, request attribute indicates that the invocation message needs response, see B.1 of ISO/IEC 14543-5-1 Ed.1
<SourceClientId> *Source client identifier* </SourceClientId>	Required field, type is 32-bit UnsignedInt
<TargetServiceId> *Target service identifier* </TargetServiceId>	Required field, type is 32-bit UnsignedInt
<SequenceId> *Invocation request sequence Id* </SequenceId>	Required field, type is 32-bit UnsignedInt
<!--Here is the start of the specific invocation request message-->	
<Invocation interface name>	Required field
<Input parameter name>*Input parameter value*</Input parameter name>	The service invocation interface that meets the requirements of ISO/IEC 14543-5-1.
……	
<Input parameter name>*Input parameter value*</Input parameter name>	The service invocation interface that meets the requirements of ISO/IEC 14543-5-1.
</Invocation interface name>	Required field
<!--Here is the end of the specific invocation request message-->	

Message	Field explanation
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field
NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.	

10.3 Service invocation response message

When an IGRS service receives an invocation request with response requirement, it should return an invocation response message. A service invocation response message is sent to the client who is invoking the service based on the same session. A message definition is shown in Table 6.

Table 6 – 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 B.1 of ISO/IEC 14543-5-1
01-IGRSVersion: IGRS/1.0	Required field, IGRS Version number
01-IGRSMessageType:InvokeServiceResponse	Required field, content shall be this
01-TargetDeviceId: target device identifier	Required field, type is URI
01-SourceDeviceId: requesting device identifier	Required field, type is URI
01-AcknowledgeId: Service pipe message response sequence Id	Required field, Type is 32 bit UnsignedInt
Content-Length: Message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:”http://schemas.xmlsoap.org/soap/envelope/”;ns=02	Required field
02-SoapAction:”IGRS-InvokeService-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/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns=”http://www.igrs.org/spec1.0”>	Required field, see B.1 of ISO/IEC 14543-5-1 Ed.1
<SourceServiceId>*Source service identifier* </SourceServiceId>	Required field, Type is 32 bit UnsignedInt
<TargetClientId>*Target client identifier* </TargetClientId>	Required field, Type is 32 bit UnsignedInt
<AcknowledgeId>*Invocation response message sequence Id* </AcknowledgeId>	Required field, Type is 32 bit UnsignedInt, It’s the same as the invocation request SequenceId of the request message.
<ReturnCode>*Invocation response status code*</ReturnCode>	Required field, defined in Clause 11 of ISO/IEC 14543-5-1
<!--Here is the specific invocation response message-->	
<FileReturnCode>*File universal error return code* </FileReturnCode>	Optional field, Indicate the error response message independent of the service invocation, such as the service invocation is not supported. There shall be one and only one return code returned between FileReturnCode and ReturnCode.
<Invocation interface name>	Required field

Message	Field explanation
<ReturnCode>*Service invocation response status code* </ReturnCode>	Optional field. Conform to service invocation return code definition. There shall be one and only one return code returned between FileReturnCode and ReturnCode.
< Output parameter>*Output parameter value* </ Output parameter>	Shall meet service interface definition requirements.
……	
< Output parameter>*Output parameter value* </ Output parameter>	Shall meet service interface definition requirements.
</ Invocation interface name>	Required field.
<!--Here is the specific invocation response message-->	
</Session>	Required field.
</SOAP-ENV:Body>	Required field.
</SOAP-ENV:Envelope>	Required field.

10.4 Content directory object update notification message

The FileAccessManagement service can notify a client based on a session setup with client when the content directory object changes. This message doesn't require any response. If content directory object update event is subscribed to a multi-directory structure, the update message should be sent only to the lowest directory object. A message definition is shown in Table 7.

Table 7 – Content directory object update notification message

Message	Field Explanation
M-NOTIFY /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host:Target host IP address:Port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see B.1 of ISO/IEC 14543-5-1
01-IGRSVersion: IGRS/1.0	Required field, IGRS Version number
01-IGRSMessageType: FamsServiceNotifyClient	Required field, content shall be this
01-SourceDeviceId: Source device identifier	Required field, type is URI
01-TargetDeviceId: Target device identifier	Required field, type is URI
01-SequenceId: Device pipe message sequence Id	Required field, type is 32 bit UnsignedInt (0 is reserved)
Content-Length:message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:"http://schemas.xmlsoap.org/soap/envelope/";ns=02	Required field
02-SoapAction:"IGRS-FamsServiceToClient-Notify"	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
<Session xmlns="http://www.igrs.org/spec1.0"request="NoResponse">	Required field, "request" attribute indicates that no response is needed, see B.1 of ISO/IEC 14543-5-1
<SourceServiceId> *Source Service identifier* </SourceServiceId>	Required field, Type is 32 bit UnsignedInt (0 is reserved)

Message	Field Explanation
<TargetClientId>*Target client identifier*</TargetClientId>	Required field, Type is 32 bit UnsignedInt (0 is reserved)
<SequenceId>*Invocation request sequence Id* </SequenceId>	Required field, Type is 32 bit UnsignedInt (0 is reserved)
<!--Here is the specific notification message-->	
<FamsUpdateNotification>	Required field
<SubscribeObjectId>The subscribed file/directory object identifier </SubscribeObjectId>	Required field, refer to the Type_ObjectId data type definition in Annex C
<ParentDirectoryId> *ID of parent directory in which the update event occurred* </ParentDirectoryId>	Required field, refer to Type_ObjectId data type definition in Annex C
<EventObjectIdList> *File/Directory object ID list in which update event occurred* </EventObjectIdList>	Optional field, refer to Type_ObjectIdList data type definition in Annex B, the updated object identifier list.
<EventType>*File/directory object update event type* </EventType>	Required field, Type is String, The options are: “ChildrenAdded”, “ChildrenDeleted”, “NameChanged”, "AccessRightChanged", "ContentChanged", "SelfDeleted"
</FamsUpdateNotification>	Required field
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

10.5 Service attribute update notification message

A Service that can notify the Client based on a session setup with this Client when the service attribute changes. This message does not require a response. A message definition is shown in Table 8.

Table 8 – Service attribute update notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host:Target host IP address:Port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see B.1 of ISO/IEC 14543-5-1
01-IGRSVersion: IGRS/1.0	Required field, IGRS Version number
01-IGRSMessageType:ServiceNotifyClient	Required field, content shall be this
01-SourceDeviceId: Source Device identifier	Required field, type is URI
01-TargetDeviceId: Target Device Identifier	Required field, type is URI
01-SequenceId: Device pipe message sequence Id	Required field, type is 32 bit UnsignedInt (0 is reserved)
Content-Length:message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:"http://schemas.xmlsoap.org/soap/envelope/";ns=02	Required field
02-SoapAction:"IGRS-ServiceToClient-Notify"	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
<Session xmlns="http://www.igrs.org/spec1.0"request="NoResponse">	Required field, "request" attribute indicates that no response is needed, see B.1 of ISO/IEC 14543-5-1
<SourceServiceId> *Source Service Identifier*</SourceServiceId>	Required field, type is 32 bit UnsignedInt (0 is reserved)
<TargetClientId> *Target Client Identifier* </TargetClientId>	Required field, type is 32 bit UnsignedInt (0 is reserved)
<SequenceId> *Invocation request sequence Id* </SequenceId>	Required field, type is 32 bit UnsignedInt (0 is reserved)
<!--Here is the specific notification message-->	
<ServiceAttributeUpdateNotification>	Required field
<SubscriptionId> *Subscription Identifier*</SubscriptionId>	Required field, String, Subscription Identifier
<ServiceAttributeName>*The updating Service attribute name* </ServiceAttributeName>	Required field, String, the updating Service attribute name
<ServiceAttributeValue > *Value of service attribute in which update event occurred* </ ServiceAttributeValue >	Required field. String
</ServiceAttributeUpdateNotification>	Required field
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

Annex A
(normative)

Description of data type generation rules

A.1 Type_ObjectId

A.1.1 General

File/directory object identifier Type_ObjectId is a global unique ID. Each file/directory object identifier may uniquely correspond to a directory or file.

A.1.2 IGRS File/Directory object identifier format

<IGRSObjectIdURN> ::= "urn:" <IGRSDeviceNS> ":" <IGRSObjectNS> ":" <IGRSObjectIdVAL>

NOTE 1 <IGRSDeviceNS>::= Device ID (GUId)

NOTE 2 <IGRSObjectNS>::= <"Directory">|<"File">

NOTE 3 <IGRSObjectIdVAL>::= Object path name, such as "/a/b/c", and is case sensitive

A.2 ObjectType

Identified Object Type. The legal values: FILE|DIRECTORY

NOTE 1 If the object is file, the value of ObjectType shall be FILE.

NOTE 2 If the object is directory, the value of ObjectType shall be DIRECTORY.

A.3 Type_FilterRule

A.3.1 Syntactic definition

< FilterRule > ::= <NILSTRING> | <LogicalExpression>

<LogicalExpression> ::= <RelationExpression> | '(' <LogicalExpression> ')' |'not' <LogicalExpression> |<LogicalExpression>

<BinaryLogicalOperator> <LogicalExpression>

<BinaryLogicalOperator> ::= 'and' | 'or'

<RelationExpression> ::= <ATTRIBUTENAME> <RelationOperator> <CONSTANTVALUE>

<RelationOperator> ::= '=' | '<' | '>' | '<=' | '>=' | '<>' | 'like' | 'not like'

A.3.2 Semantic description

The following applies.

a) <NILSTRING> denotes it is a null string. Any Item shall satisfy the searching condition indicated by <NILSTRING>.

b) <ATTRIBUTENAME> is the attribute name included in the return result of GetSearchCapability.

c) <CONSTANTVALUE> is a constant, its type shall match the type of <ATTRIBUTRNAME>;

d) 'like'|'not like' is a relation operator of mode matching. It denotes 'matching'|'not matching' respectively;

e) The wild card that appears in matching mode is '%'. For example, 'abc%' means a string started with abc. '%abc' means a string ended with abc. '%abc%' means a string contained abc. 'abc%def' means a string started with abc and ended with def, etc.

f) '='|'<'|'>'|'<='|'>='|'<>' are comparison operators. They denote "equal'|"less than"|"greater than"|"greater than or equal"|"less than or equal"|"not equal".

g) 'and'|'or' are binary logic operators.

h) 'not' is a unitary logic operator.

A.4 Type_SortRule

A.4.1 Syntactic definition

<SortRule>::= <NILSTRING> | <SortExpression>

<SortExpression>::= <ATTRIBUTENAME> <SortSwitch> |<ATTRIBUTENAME> <SortSwitch> ',' <SortExpression>

<SortSwitch>::= 'ASC' | 'DESC'

A.4.2 Semantic description

The following applies.

a) <NILSTRING> denotes a null string. It indicates that the result can be processed without sorting.

b) <ATTRIBUTENAME> is the name of attribute that is contained in the return result of GetSortCapability.

c) 'ASC'|'DESC' are sorting options. They mean "ascending sort"|"descending sort".

d) The priority of the sorting indicators separated by ',' is from left to right.

A.5 Content directory object update notification message

```
    <element name="FamsUpdateNotification">
        <complexType>
            <sequence>
                <element name="SubscribeObjectId" type="xsd:string"/>
                <element name="ParentDirectoryId" type="xsd:string"/>
                <element name="EventObjectIdList" minOccurs="0">
                    <complexType>
                        <sequence>
                            <element              name="EventObjectId"
type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="EventType">
                    <complexType>
                        <choice>
                            <element name="ChildrenAdded/>
                            <element name="ChildrenDeleted"/>
                            <element name="NameChanged"/>
                            <element name="AccessRightChanged"/>
```

```
<element name="ContentChanged"/>
                                                            <element name="SelfDeleted"/>
                                                  </choice>
                                        </complexType>
                              </element>
                    </sequence>
          </complexType>
</element>
```

A.6 Service attribute update notification message

```
<element name="ServiceAttributeUpdateNotification">
          <complexType>
                    <sequence>
                              <element name="SubscriptionId" type="xsd:string"/>
                              <element name="ServiceAttributeName" type="xsd:string"/>
                              <element name="ServiceAttributeValue" type="xsd:string"/>
                    </sequence>
          </complexType>
</element>
```

Annex B
(normative)

Information device – IGRS – File profile message format

B.1 General

IGRS File Profile messages adopt XML Schema description. Service invocation message formats of IGRS File Profile shall conform to the service invocation message format specified in ISO/IEC 14543-5-1.

B.1.1 Service invocation request message

The Client can initiate “request/response” service invocation based on the session with the target service. The definition of the service invocation request message is shown in Table B.1.

Table B.1 – IGRS file profile service invocation request message

Message	Field explanation
M-POST /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host: Target Host IP address:Port	Required field
01-IGRSVersion: IGRS/1.0	Required field, IGRS version number
01-IGRSMessageType:InvokeServiceRequest	Required field, content shall be this
01-SourceDeviceId: requesting device identifier	Required field, and type is URI, defined in 8.1.2 of ISO/IEC 14543-5-1 ED.1
01-TargetDeviceId: target device identifier	Required field, type is URI, defined in 8.1.2 of ISO/IEC 14543-5-1 Ed.1
Content-Length: Message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:”http://www.igrs.org/spec1.0”;ns=01	Required field, see B.1 of ISO/IEC 14543-5-1
MAN:”http://schemas.xmlsoap.org/soap/envelope/”;ns=02	Required field
02-SoapAction:”IGRS-InvokeService-Request”	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
<Session xmlns=”http://www.igrs.org/spec1.0”request=”NeedResponse”>	Required field, request attribute indicates that the invocation message needs response, see B.1 of ISO/IEC 14543-5-1 Ed.1
<SourceClientId> *Source client identifier* </SourceClientId>	Required field, type is 32-bit UnsignedInt
<TargetServiceId> *Target service identifier* </TargetServiceId>	Required field, type is 32-bit UnsignedInt
<SequenceId> *Invocation request sequence Id* </SequenceId>	Required field, type is 32-bit UnsignedInt
<!--Here is the start of the specific invocation request message-->	
<Invocation interface name>	Required field
<Input parameter name>*Input parameter value*</Input parameter name>	The service invocation interface that meets ISO/IEC 14543-5-2 series
……	

Message	Field explanation
<Input parameter name>*Input parameter value*</Input parameter name>	The service invocation interface that meets the ISO/IEC 14543-5-2X series.
</Invocation interface name>	Required field
<!--Here is the end of the specific invocation request message-->	
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

NOTE Italics indicate where content is to be inserted; all other text in message definitions is fixed.

B.1.2 Service invocation response message

When an IGRS service receives an invocation request with a response requirement, it should return an invocation response message. A service invocation response message is sent to the client who is invoking the service based on the same session. A message definition is shown in Table B.2.

Table B.2 – 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 B.1 of ISO/IEC 14543-5-1 Ed.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS Version number
01-IGRSMessageType:InvokeServiceResponse	Required field, content shall be this
01-TargetDeviceId: target device identifier	Required field, type is URI
01-SourceDeviceId: requesting device identifier	Required field, type is URI
01-AcknowledgeId: Service pipe message response sequence Id	Required field, Type is 32 bit UnsignedInt
Content-Length: Message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:”http://schemas.xmlsoap.org/soap/envelope/”;ns=02	Required field
02-SoapAction:”IGRS-InvokeService-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/">	Required field
<SOAP-ENV:Body>	Required field
<Session xmlns=”http://www.igrs.org/spec1.0”>	Required field, see B.1 of ISO/IEC 14543-5-1 Ed.1
<SourceServiceId>*Source service identifier* </SourceServiceId>	Required field, Type is 32 bit UnsignedInt
<TargetClientId>*Target client identifier* </TargetClientId>	Required field, Type is 32 bit UnsignedInt
<AcknowledgeId>*Invocation response message sequence Id* </AcknowledgeId>	Required field, Type is 32 bit UnsignedInt, It’s the same as the invocation request SequenceId of the request message.
<ReturnCode>*Invocation response status code*</ReturnCode>	Required field, defined in Clause 11 of ISO/IEC 14543-5-1 Ed.1.
<!--Here is the specific invocation response message-->	

Message	Field explanation
<FileReturnCode>*File universal error return code* </FileReturnCode>	Optional field, Indicate the error response message independent of the service invocation, such as the service invocation is not supported. There shall be one and only one return code returned between the FileReturnCode and ReturnCode.
<Invocation interface name>	Required field
<ReturnCode>*Service invocation response status code* </ReturnCode>	Optional field. Conforms to service invocation return code definition. There shall be one and only one return code returned between the FileReturnCode and ReturnCode.
< Output parameter>*Output parameter value* </ Output parameter>	Shall meet service interface definition requirements.
……	
< Output parameter>*Output parameter value* </ Output parameter>	Shall meet service interface definition requirements.
</ Invocation interface name>	Required field
<!--Here is the specific invocation response message-->	
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

B.1.3 File/Directory object update notification message

The FileAccessManagement service can notify a client based on a session setup with a client when the file/directory object changes. This message does not require any response. If a file/directory object update event is subscribed to a multi-directory structure, the update message should be sent only to the lowest directory object. A message definition is shown in Table B.3.

Table B.3 – File/Directory object update notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host:Target host IP address:Port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see B.1 of ISO/IEC 14543-5-1 Ed.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS Version number
01-IGRSMessageType: FamsServiceNotifyClient	Required field, content shall be this
01-SourceDeviceId: Source device identifier	Required field, type is URI
01-TargetDeviceId: Target device identifier	Required field, type is URI
01-SequenceId: Device pipe message sequence Id	Required field, type is 32 bit UnsignedInt (0 is reserved)
Content-Length:message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:"http://schemas.xmlsoap.org/soap/envelope/";ns=02	Required field
02-SoapAction:"IGRS-FamsServiceToClient-Notify"	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

Message	Field explanation
<Session xmlns="http://www.igrs.org/spec1.0"request="NoResponse">	Required field, "request" attribute indicates that no response is needed, see B.1 of ISO/IEC 14543-5-1
<SourceServiceId> *Source Service identifier* </SourceServiceId>	Required field, Type is 32 bit UnsignedInt (0 is reserved)
<TargetClientId>*Target client identifier*</TargetClientId>	Required field, Type is 32 bit UnsignedInt (0 is reserved)
<SequenceId>*Invocation request sequence Id* </SequenceId>	Required field, Type is 32 bit UnsignedInt (0 is reserved)
<!--Here is the specific notification message-->	
<FamsUpdateNotification>	Required field
<SubscribeObjectId>The subscribed file/directory object identifier </SubscribeObjectId>	Required field, refer to the Type_ObjectId data type definition in Annex C.
<ParentDirectoryId> *ID of parent directory in which the update event occurred* </ParentDirectoryId>	Required field, refer to Type_ObjectId data type definition in Annex C.
<EventObjectIdList> *File/Directory object ID list in which update event occurred* </EventObjectIdList>	Optional field, refer to Type_ObjectIdList data type definition in Annex B, the updated object identifier list.
<EventType>*File/directory object update event type* </EventType>	Required field, type is string. The options are: "ChildrenAdded", "ChildrenDeleted", "NameChanged", "AccessRightChanged", "ContentChanged", "SelfDeleted"
</FamsUpdateNotification>	Required field
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

NOTE The XML schema of a specific notification message in a file/directory object update notification message is defined as:

```
<element name="FamsUpdateNotification">
       <complexType>
           <sequence>
               <element name="SubscribeObjectId" type="xsd:string"/>
               <element name="ParentDirectoryId" type="xsd:string"/>
               <element name="EventObjectIdList" minOccurs="0">
                   <complexType>
                       <sequence>
                           <element name="EventObjectId"
type="xsd:string"
                  minOccurs="1" maxOccurs="unbounded"/>
                       </sequence>
                   </complexType>
               </element>
               <element name="EventType">
                   <complexType>
                       <choice>
                           <element name="ChildrenAdded/>
                           <element name="ChildrenDeleted"/>
                           <element name="NameChanged"/>
                           <element
name="AccessRightChanged"/>
           <element name="ContentChanged"/>
                           <element name="SelfDeleted"/>
                       </choice>
                   </complexType>
               </element>
           </sequence>
       </complexType>
   </element>
```

B.1.4 Service attribute update notification message

This service can notify the Client based on a session setup with a client when the service attribute changes. This message does not require a response. A message definition is shown in Table B.4.

Table B.4 – Service attribute update notification message

Message	Field explanation
M-NOTIFY /IGRS HTTP/1.1	HTTP EXTENDED COMMAND LINE
Host:Target host IP address:Port	Required field
MAN:"http://www.igrs.org/spec1.0";ns=01	Required field, see B.1 of ISO/IEC 14543-5-1 Ed.1
01-IGRSVersion: IGRS/1.0	Required field, IGRS Version number
01-IGRSMessageType:ServiceNotifyClient	Required field, content shall be this
01-SourceDeviceId: Source Device identifier	Required field, type is URI
01-TargetDeviceId: Target Device Identifier	Required field, type is URI
01-SequenceId: Device pipe message sequence Id	Required field, type is 32 bit UnsignedInt (0 is reserved)
Content-Length:message body length	Required field
Content-type:text/xml; charset=utf-8	Required field
MAN:"http://schemas.xmlsoap.org/soap/envelope/";ns=02	Required field
02-SoapAction:"IGRS-ServiceToClient-Notify"	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
<Session xmlns="http://www.igrs.org/spec1.0"request="NoResponse">	Required field, "request" attribute indicates that no response is needed, see B.1 of ISO/IEC 14543-5-1 Ed.1
<SourceServiceId> *Source Service Identifier*</SourceServiceId>	Required field, type is 32 bit UnsignedInt (0 is reserved)
<TargetClientId> *Target Client Identifier* </TargetClientId>	Required field, type is 32 bit UnsignedInt (0 is reserved)
<SequenceId> *Invocation request sequence Id* </SequenceId>	Required field, type is 32 bit UnsignedInt (0 is reserved)
<!--Here is the specific notification message-->	
<ServiceAttributeUpdateNotification>	Required field
<SubscriptionId> *Subscription Identifier*</SubscriptionId>	Required field, String, Subscription Identifier
<ServiceAttributeName>*The updating Service attribute name* </ServiceAttributeName>	Required field, String, the updating Service attribute name
<ServiceAttributeValue > *Value of service attribute in which update event occurred* </ ServiceAttributeValue >	Required field. String.
</ServiceAttributeUpdateNotification>	Required field
</Session>	Required field
</SOAP-ENV:Body>	Required field
</SOAP-ENV:Envelope>	Required field

NOTE The XML schema of a specific notification message in a service attribute update notification message is defined as:

```
    <element name="ServiceAttributeUpdateNotification">
        <complexType>
            <sequence>
                <element name="SubscriptionId" Type="xsd:string"/>
                <element name="ServiceAttributeName" Type="xsd:string"/>
                <element name="ServiceAttributeValue" Type="xsd:string"/>
            </sequence>
        </complexType>
    </element>
```

B.2 FileAccessManagement Service

B.2.1 Service data type

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”

targetNamespace =“http://www.igrs.org/spec1.0”>            <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="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>
</schema>
```

B.2.2 Service invocation message format

B.2.2.1 GetAuthenticationKey

B.2.2.1.1 GetAuthenticationKeyRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”GetAuthenticationKeyRequest”>
        <complexType>
            <sequence>
                <element name=”InstanceId” Type=”string”/>
                <element name=”UserAuthenticationInfo”
            type=”Type_UserAuthenticationInfo”/>
            </sequence>
        </complexType>
    </element>
```

B.2.2.1.2 GetAuthenticationKeyResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”GetAuthenticationKeyResponse”>
        <complexType>
            <sequence>
                            <element name=”ReturnCode” type=”unsignedInt”/>
                <element name=”AuthenticationKey” type=”string”/>
            </sequence>
        </complexType>
    </element>
</schema>
```

B.2.2.2 GetSortCapability

B.2.2.2.1 GetSortCapabilityRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”GetSortCapabilityRequest”/>
        <complexType>
            <sequence>
                <element name=” AuthenticationKey” type=”string”/>
            </sequence>
        </complexType>
    </element>
```

B.2.2.2.2 GetSortCapabilityResponse

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

xmlns:igrs = "http://www.igrs.org/spec1.0"
targetNamespace ="http://www.igrs.org/spec1.0">
    <element name="GetSortCapabilityResponse>
        <complexType>
            <sequence>
                            <element name="ReturnCode" type="unsignedInt"/>
                            <element name="SortCaps" ref="igrs: Type_SortCapability"/>
            </sequence>
        </complexType>
    </element>
</schema>
```

B.2.2.3 GetSearchCapability

B.2.2.3.1 GetSearchCapabilityRequest

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

xmlns:igrs = "http://www.igrs.org/spec1.0"
targetNamespace ="http://www.igrs.org/spec1.0">
    <element name="GetSearchCapabilityRequest">
        <complexType>
            <sequence>
                <element name=" AuthenticationKey" type="string"/>
            </sequence>
        </complexType>
    </element>
```

B.2.2.3.2 GetSearchCapabilityResponse

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

xmlns:igrs = "http://www.igrs.org/spec1.0"
    targetNamespace ="http://www.igrs.org/spec1.0"><element
name="GetType_SearchCapabilityResponse">
        <complexType>
            <sequence>
                            <element name="ReturnCode" type="unsignedInt"/>
                <element name="SearchCaps" ref="igrs: Type_SearchCapability"/>
            </sequence>
        </complexType>
    </element>
```

B.2.2.4 Browse

B.2.2.4.1 BrowseRequest

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

xmlns:igrs = "http://www.igrs.org/spec1.0"
targetNamespace ="http://www.igrs.org/spec1.0">
    <element name="BrowseRequest">
        <complexType>
            <sequence>
                <element name="AuthenticationKey" type="string"/>
                <element name="ObjectId" type="Type_ObjectId"/>
                <element name="BrowseFilter" type="Type_FilterRule"/>
                            <element name="StartOffset" type="Type_Count"/>
```

```
                    <element name=”RequestedCount” type=”Type_Count”/>
                                <element name=”SortRule” type=”Type_SortRule”/>
                </sequence>
                  </complexType>
      </element>
```

B.2.2.4.2 BrowseResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”BrowseResponse”>
            <complexType>
                <sequence>
                                <element name=”ReturnCode” type=”unsignedInt”/>
                     <element name=”Result” type=”Type_ObjectList”/>
                     <element name=”NumberReturned” type=”Type_Count”/>
                    <element name=”NumberTotalMatched” type=”Type_Count”/>
                </sequence>
            </complexType>
      </element>
```

B.2.2.5 GetAttribute

B.2.2.5.1 GetAttributeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”GetAttributeRequest”>
            <complexType>
                <sequence>
                    <element name=”AuthenticationKey” type=”string”/>
                    <element name=”ObjectId” type=” Type_ObjectId”/>
                </sequence>
                  </complexType>
      </element>
```

B.2.2.5.2 GetAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”GetFileAttributeResponse”>
            <complexType>
                <sequence>
                                <element name=”ReturnCode” type=”unsignedInt”/>
                    <element name=”ObjectAttribute” type=” Type_ObjectAttribute”/>
                </sequence>
            </complexType>
      </element>
```

B.2.2.6 SetAttribute

B.2.2.6.1 SetAttribute request

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”SetAttributeRequest”>
            <complexType>
            <sequence>
                <element name=” AuthenticationKey” type=”string”/>
                <element name=”ObjectId” type=” Type_ObjectId”/>
                <element name=”ObjectAttribute” type=” Type_ObjectAttribute”/>
            </sequence>
              </complexType>
    </element>
```

B.2.2.6.2 SetAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” SetAttributeResponse”>
        <complexType>
            <sequence>
                            <element name=”ReturnCode” type=”unsignedInt”/>
            </sequence>
        </complexType>
    </element>
```

B.2.2.7 Search

B.2.2.7.1 SearchRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”SearchRequest”>
        <complexType>
            <sequence>
                <element name=” AuthenticationKey” type=”string”/>
                <element name=”ObjectIdList” type=” Type_ObjectIdList”/>
                            <element name=”SearchRule” type=” Type_FilterRule”/>
                     <element name=”StartOffset” type=” Type_Count”/>
                <element name=”RequestedCount” type=” Type_Count”/>
                            <element name=”SortRule” type=” Type_SortRule”/>
            </sequence>
              </complexType>
    </element>
```

B.2.2.7.2 SearchResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”SearchResponse”>
        <complexType>
            <sequence>
                            <element name=”ReturnCode” type=”unsignedInt”/>
                <element name=”Result” type=” Type_ObjectList”/>
                <element name=”NumberReturned” type=” Type_Count”/>
                <element name=”NumberTotalMatched” type=” Type_Count”/>
            </sequence>
        </complexType>
    </element>
```

B.2.2.8 GetBrowseFilter

B.2.2.8.1 GetBrowseFilterRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” GetBrowseFilterRequest”>
        <complexType>
            <sequence>
                            <element name=” AuthenticationKey” type=”string”/>
            </sequence>
              </complexType>
    </element>
```

B.2.2.8.2 GetBrowseFilterResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”GetBrowseFilterResponse”>
        <complexType>
            <sequence>
                            <element name=”ReturnCode” type=”unsignedInt”/>
                <element name=”BrowseFilter” type=” Type_FilterRule”/>
            </sequence>
        </complexType>
       </element>
</schema>
```

B.2.2.9 SetBrowseFilter

B.2.2.9.1 SetBrowseFilterRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” SetBrowseFilterRequest”>
        <complexType>
            <sequence>
                <element name=”AuthenticationKey” type=”string”/>
                <element name=”BrowseFilter” type=” Type_FilterRule”/>
            </sequence>
              </complexType>
    </element>
```

B.2.2.9.2 SetBrowseFilterResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”SetBrowseFilterResponse”>
        <complexType>
            <sequence>
                            <element name=”ReturnCode” type=”unsignedInt”/>
            </sequence>
        </complexType>
       </element>
</schema>
```

B.2.2.10 New

B.2.2.10.1 NewRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” NewRequest”>
        <complexType>
                     <sequence>
                <element name=” AuthenticationKey” type=”string”/>
                            <element name=”ParentId” type=” Type_ObjectId”/>
                            <element name=”ObjectAttribute” type=”
Type_ObjectAttribute”/>
                     </sequence>
              </complexType>
    </element>
```

B.2.2.10.2 NewResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” NewResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
                            <element name=”ObjectId” type=” Type_ObjectId”/>
            </sequence>
        </complexType>
       </element>
```

B.2.2.11 Copy

B.2.2.11.1 CopyRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” CopyRequest”>
        <complexType>
                     <sequence>
                <element name=” AuthenticationKey” type=”string”/>
                            <element name=”SourceObjectId” type=”Type_ObjectId”/>
                            <element name=”DestParentId” type=”Type_ObjectId”/>
                     </sequence>
        </complexType>
    </element>
```

B.2.2.11.2 CopyResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”CopyResponse”>
            <complexType>
                  <sequence>
                        <element name=”ReturnCode” type=”unsignedInt”/>
                        <element name=”DestObjectId” type=”Type_ObjectId”/>
                        <element name=”DestObjectAttribute” type=”Type_ObjectAttribute”/>
                  </sequence>
            </complexType>
          </element>
```

B.2.2.12 Move

B.2.2.12.1 MoveRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”MoveRequest”>
            <complexType>
                              <sequence>
                        <element name=”AuthenticationKey” type=”string”/>
                                    <element name=”SourceObjectId” type=”Type_ObjectId”/>
                                    <element name=”DestParentId” type=”Type_ObjectId”/>
                              </sequence>
            </complexType>
          </element>
```

B.2.2.12.2 MoveResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”MoveResponse”>
            <complexType>
                  <sequence>
                        <element name=”ReturnCode” type=”unsignedInt”/>
                        <element name=”DestObjectId” type=”Type_ObjectId”/>
                        <element name=”DestObjectAttribute” type=”Type_ObjectAttribute”/>
                  </sequence>
            </complexType>
          </element>
```

B.2.2.13 Delete

B.2.2.13.1 DeleteRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”DeleteRequest”>
        <complexType>
                    <sequence>
                <element name=”AuthenticationKey” type=”string”/>
                    <element name=”ObjectId” type=”Type_ObjectId”/>
                                 <element name=”DeleteMode” type=”
Type_DeleteMode”/>
                           </sequence>
        </complexType>
       </element>
```

B.2.2.13.2 DeleteResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”DeleteResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
            </sequence>
        </complexType>
       </element>
```

B.2.2.14 PrepareforDownload

B.2.2.14.1 PrepareforDownloadRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” PrepareforDownloadRequest”>
        <complexType>
                    <sequence>
                <element name=”AuthenticationKey” type=”string”>
                           <element name=”SourceObjectIdList” type=”
Type_ObjectIdList”/>
                    </sequence>
        </complexType>
       </element>
```

B.2.2.14.2 PrepareforDownloadResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” PrepareforDownloadResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
                    <element name=”SourceObjectURITreeList”
           type=”Type_ObjectURITreeList”/>
            </sequence>
        </complexType>
       </element>
```

B.2.2.15 PrepareforUpload

B.2.2.15.1 PrepareforUploadRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” PrepareforUploadRequest”>
        <complexType>
                    <sequence>
                <element name=”AuthenticationKey” type=”string”/>
                <element name=”ObjectAttribute” type=”Type_ObjectAttribute”/>
                <element name=”DestParentId” type=”Type_ObjectId”/>
                    </sequence>
        </complexType>
       </element>
```

B.2.2.15.2 PrepareforUploadResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” PrepareforUploadResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
                <element name=”DestParentURI” type=”Type_ObjectURI”/>
            </sequence>
        </complexType>
       </element>
```

B.2.2.16 SubscribeObjectChange

B.2.2.16.1 SubscribeObjectChangeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=” SubscribeObjectChangeRequest”>
        <complexType>
                    <sequence>
                <element name=”AuthenticationKey” type=”string”/>
                           <element name=”ObjectId” type=" Type_ObjectId”/>
                    </sequence>
        </complexType>
       </element>
```

B.2.2.16.2 SubscribeObjectChangeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=” SubscribeObjectChangeResponse”>
            <complexType>
                              <sequence>
                                        <element name=”ReturnCode” type=”unsignedInt”/>
                              </sequence>
            </complexType>
          </element>
```

B.2.2.17 UnSubscribeServiceAttribute

B.2.2.17.1 UnSubscribeServiceAttributeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”UnSubscribeServiceAttributeRequest”>
            <complexType>
                              <sequence>
                        <element name=” AuthenticationKey” type=”string”/>
                                        <element name=”ObjectId” type=" Type_ObjectId"/>
                              </sequence>
            </complextType>
          </element>
```

B.2.2.17.2 UnSubscribeServiceAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”UnSubscribeServiceAttributeResponse”>
            <complexType>
                              <sequence>
                                        <element name=”ReturnCode” type=”unsignedInt”/>
                              </sequence>
            </complexType>
          </element>
```

B.2.2.18 SubscribeServiceAttribute

B.2.2.18.1 SubscribeServiceAttributeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
      <element name=”SubscribeServiceAttributeRequest”>
            <complexType>
                              <sequence>
                        <element name=” AuthenticationKey” type=”string”/>
                                        <element name=”ServiceAttributeName” type=”string”/>
                              </sequence>
            </complextType>
          </element>
```

B.2.2.18.2 SubscribeServiceAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”SubscribeServiceAttributeResponse”>
        <complexType>
            <sequence>
                           <element name=”ReturnCode” type=”unsignedInt”/>
                           <element name=”SubscriptionId” type=”string”/>
                    </sequence>
        </complexType>
       </element>
```

B.2.2.19 UnsubscribeServiceAttribute

B.2.2.19.1 UnsubscribeServiceAttributeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”UnsubscribeServiceAttributeRequest”>
        <complexType>
                    <sequence>
                <element name=” AuthenticationKey” type=”string”/>
                           <element name=”SubscriptionId” type=”string”/>
                    </sequence>
        </complextType>
       </element>
```

B.2.2.19.2 UnsubscribeServiceAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”UnsubscribeServiceAttributeResponse”>
        <complexType>
                    <sequence>
                           <element name=”ReturnCode” type=”unsignedInt”/>
                    </sequence>
        </complexType>
       </element>
```

B.3 FileConnectionManagement Service

B.3.1 Service data type

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>

              <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>
```

B.3.2 Service Invocation Message Format

B.3.2.1 GetProtocolInfo

B.3.2.1.1 GetProtocolInfoRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”GetProtocolInfoRequest”>
```

B.3.2.1.2 GetProtocolInfoResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”GetProtocolInfoResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
                <element name=”ProtocolInfoList” ref=”igrs: Type_ProtocolInfoList”/>
                <element name=”IPList” ref=”igrs: Type_IPList”/>
            </sequence>
        </complexType>
       </element>
```

B.3.2.2 PrepareforConnection

B.3.2.2.1 PrepareforConnectionRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”PrepareforConnectionRequest”>
        <complexType>
            <sequence>
                <element name=”RemoteProtocolInfo” type=”Type_ProtocolInfo”/>
            </sequence>
        </complexType>
    </element>
```

B.3.2.2.2 PrepareforConnectionResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”PrepareforConnectionResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
                <element name=”ConnectionId” type=” Type_ConnectionId”/>
            </sequence>
        </complexType>
       </element>
```

B.3.2.3 ReleaseConnection

B.3.2.3.1 ReleaseConnectionRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”ReleaseConnectionRequest”.
        <complexType>
              <sequence>
                            <element name=”ConnectionId” type=” Type_ConnectionId”/>
            </sequence>
        </complexType>
    </element>
```

B.3.2.3.2 ReleaseConnectionResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”ReleaseConnectionResponse”>
        <complexType>
            <sequence>
                <element name=”ReturnCode” type=”unsignedInt”/>
            </sequence>
        </complexType>
       </element>
```

B.3.2.4 SubscribeServiceAttribute

B.3.2.4.1 SubscribeServiceAttributeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”SubscribeServiceAttributeRequest”>
        <complexType>
                    <sequence>
                          <element name=”ServiceAttributeName” Type=”string”/>
                    </sequence>
        </complexType>
       </element>
```

B.3.2.4.2 SubscribeServiceAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”SubscribeServiceAttributeResponse”>
        <complexType>
                    <sequence>
                          <element name=”ReturnCode” type=”unsignedInt”/>
                          <element name=”SubscriptionId” type=”string”/>
                    </sequence>
        </complexType>
       </element>
```

B.3.2.5 UnsubscribeServiceAttribute

B.3.2.5.1 UnsubscribeServiceAttributeRequest

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”UnsubscribeServiceAttributeRequest”>
        <complexType>
                    <sequence>
                          <element name=”SubscriptionId” type=”string”/>
                    </sequence>
        </complexType>
       </element>
```

B.3.2.5.2 UnsubscribeServiceAttributeResponse

```
<schema xmlns = “http://www.w3.org/2001/XMLSchema”

xmlns:igrs = “http://www.igrs.org/spec1.0”
targetNamespace =“http://www.igrs.org/spec1.0”>
    <element name=”UnsubscribeServiceAttributeResponse”>
        <complexType>
                    <sequence>
                          <element name=”ReturnCode” type=”unsignedInt”/>
                    </sequence>
        </complexType>
       </element>
</schema>
```