



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.380.8

(11/2011)

SERIES J: CABLE NETWORKS AND TRANSMISSION
OF TELEVISION, SOUND PROGRAMME AND OTHER
MULTIMEDIA SIGNALS

Digital transmission of television signals

Digital Program Insertion–Advertising Systems Interfaces: Part 8 – General Information Service

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU [had/had not] received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2011

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Recommendation ITU-T J.380.8 (ex J.dpi-asi-8)

Digital Program Insertion–Advertising Systems Interfaces: Part 8 – General Information Service

Summary

This document describes the Digital Program Insertion Advertising Systems Interfaces' General Information Service (GIS) messaging and data type specification using XML, XML Namespaces, and XML Schema.

Keywords

None

TABLE OF CONTENTS

1	Scope	11
2	References.....	11
3	Definitions	11
3.1	Terms defined elsewhere:.....	11
3.2	Terms defined in this Recommendation.....	11
4	Abbreviations and acronyms	13
5	Conventions	13
5.1	Notational Conventions	13
5.1.1	Normative XML Schema	13
5.1.2	Document Conventions	13
5.2	Processing Conventions.....	13
5.3	XML Namespaces	13
6	GIS Message TYPES.....	14
6.1	@version Attribute	15
6.2	Request Base Message Type	15
6.3	Response Base Message Type.....	15
6.4	Notification Base Message Type.....	15
6.5	Acknowledgement Base Message Type.....	16
6.6	Messages Requiring Notification Registration.....	16
6.7	Default Logical Service Channel End-Point	16
7	GIS Message Exchange	16
7.1	ListSupportedFeaturesRequest and ListSupportedFeaturesResponse Message Types	17
7.1.1	ListSupportedFeaturesRequest Message Type	17
7.1.2	ListSupportedFeaturesResponse Message Type	18
7.2	ListQualifiersRequest and ListQualifiersResponse Message Types.....	20
7.2.1	ListQualifiersRequest Message Type	20
7.2.2	ListQualifiersResponse Message Type	21
7.3	ListNotificationRegistrationRequest and ListNotificationRegistrationResponse Message Types.....	23
7.3.1	ListNotificationRegistrationRequest Message Types.....	23
7.3.2	ListNotificationRegistrationResponse Message Type	25
7.4	NotificationRegistrationRequest and NotificationRegistrationResponse Message Types	27
7.4.1	NotificationRegistrationRequest Message Type.....	27
7.4.2	NotificationRegistrationResponse Message Type	29
7.5	Notification and NotificationAcknowledgement Message Types.....	30

7.5.1	Notification Message Type	30
7.5.2	NotificationAcknowledgement Message Type	32
7.6	CreateCursorRequest and CreateCursorResponse Message Types.....	33
7.6.1	CreateCursorRequest Message Type	34
7.6.2	CreateCursorResponse Message Type	35
7.7	CancelCursorRequest and CancelCursorResponse Message Types	37
7.7.1	CancelCursorRequest Message Type.....	37
7.7.2	CancelCursorResponse Message Type	39
7.8	QueryRequest and QueryResponse Message Types	39
7.8.1	QueryRequest Message Type.....	40
7.8.2	QueryResponse Message Type	41
7.9	NotificationDeregisterRequest and NotificationDeregisterResponse Message Types	42
7.9.1	NotificationDeregisterRequest Message Type.....	43
7.9.2	NotificationDeregisterResponse Message Type	44
7.10	DeregistrationNotification and DeregistrationAcknowledgement Message Types	45
7.10.1	DeregistrationNotification Message Type	45
7.10.2	DeregistrationAcknowledgement Message Type	47
7.11	Service Check Messages	48
7.12	Service Status Messages.....	48
8	GIS Element Details	49
8.1	AdvancedFilterElement.....	50
8.2	AdvancedQueryFilter	51
8.3	AdvancedQueryLanguage	51
8.4	AdvancedQueryResult.....	52
8.5	AdvancedQueryResultData	53
8.6	BasicFilterElement	53
8.7	BasicQueryDataModelDescription.....	54
8.8	BasicQueryFilter.....	55
8.9	BasicQueryResult	56
8.10	Cursor	57
8.11	EnumerationValue	59
8.12	MaxFloat.....	59
8.13	MaxInteger	59
8.14	MaxLength	60
8.15	MinFloat	60
8.16	MinInteger	61
8.17	Qualifier.....	61
8.18	QualifierDeclaration	62
8.19	QualifierDescription	62

8.20	QualifierSet.....	64
8.21	Query	65
8.22	QueryResult.....	66
8.23	ServiceDataModel	71
8.24	ServiceDataModelProfile	71
8.25	UniqueQualifier.....	72
8.26	UniqueQualifierDeclaration	72
9	J.380.8 GIS attribute types	73
9.1	cursorIdAttrType Attribute Type	73
9.2	cursorIdRefAttrType Attribute Type.....	73
9.3	expandOutputAttrType Attribute Type	73
9.4	filterElementNameType Attribute Type.....	74
9.5	filterElementValueAttrType Attribute Type	74
9.6	MinMaxFloatType Type	74
9.7	MinMaxIntegerType Type	74
9.8	MaxLengthType Type	74
9.9	NotificationTypeEnumeration Type.....	74
9.10	qualifierNameAttrType Type	74
9.11	QualifierValueTypeEnumerationType	74
9.12	QueryFilterOpTypeEnumeration Attribute Type	75
9.13	queryIdAttrType Attribute Type	75
9.14	queryIdRefAttrType Attribute Type.....	75
9.15	queryLanguageAttrType Attribute Type.....	75
9.16	resultSetSizeOnlyAttrType Attribute Type.....	75
9.17	totalResultSetSizeAttrType Attribute Type	75
9.18	uniqueQualifierNameAttrType Attribute Type.....	76
10	Basic Queries and Regular Expressions	76
10.1	Regular Expressions and Wildcards.....	76
	Annex A Statuscode Element @Detail Attribute Values	78
	Appendix I Complex queries & expanded output examples.....	80
I.1	Multiple filter elements	80
I.2	Expanded Output	81
	Appendix II ADVANCED QUERIES.....	83
II.1	Advanced Query Examples	83
	Appendix III Cursor Examples	84
III.1	Creating Cursors.....	84
III.2	Traversing a Cursor	85
III.3	Canceling Existing Cursors	85

Appendix IV	Message Examples.....	87
IV.1	Listing Supported Features.....	87
IV.2	Listing Qualifiers.....	88
IV.3	Query Examples	89
IV.4	Notification Registration Examples	92
IV.5	Notification.....	93
Bibliography.....		94

LIST OF FIGURES

Figure 1 - General Information Service Interface.....	9
Figure 2 - Substitution Group Schema Convention.....	13
Figure 3 - GIS Message Exchange.....	17
Figure 4 - ListSupportedFeaturesRequestType XML Schema.....	18
Figure 5 - ListSupportedFeaturesResponseType XML Schema.....	19
Figure 6 - ListQualifiersRequestType XML Schema.....	21
Figure 7 - ListQualifiersResponseType XML Schema.....	22
Figure 8 - ListNotificationRegistrationRequestType XML Schema.....	24
Figure 9 - ListNotificationRegistrationResponseType XML Schema.....	26
Figure 10 - NotificationRegistrationRequestType XML Schema.....	28
Figure 11 - NotificationRegistrationResponseType XML Schema.....	30
Figure 12 - NotificationType XML Schema.....	32
Figure 13 - NotificationAcknowledgementType XML Schema.....	33
Figure 14 - CreateCursorRequestType XML Schema.....	34
Figure 15 - CreateCursorResponseType XML Schema.....	36
Figure 16 - CancelCursorRequestType XML Schema.....	38
Figure 17 - CancelCursorResponseType XML Schema.....	39
Figure 18 - QueryRequestType XML Schema.....	40
Figure 19 - QueryResponseType XML Schema.....	42
Figure 20 - NotificationDeregisterRequestType XML Schema.....	43
Figure 21 - NotificationDeregisterResponseType XML Schema.....	44
Figure 22 - DeregistrationNotificationType XML Schema.....	46
Figure 23 - DeregistrationAcknowledgementType XML Schema.....	47
Figure 24 - AdvancedFilterElement Element XML Schema.....	50
Figure 25 - AdvancedQueryFilter Element XML Schema.....	51
Figure 26 - AdvancedQueryLanguage Element XML Schema.....	52
Figure 27 - AdvancedQueryResult Element XML Schema.....	52
Figure 28 - AdvancedQueryResultData Element XML Schema.....	53
Figure 29 - BasicFilterElement Element XML Schema.....	54
Figure 30 - BasicQueryDataModelDescription Element XML Schema.....	55
Figure 31 - BasicQueryFilter Element XML Schema.....	56
Figure 32 - BasicQueryResult Element XML Schema.....	57
Figure 33 - Cursor Element XML Schema.....	58
Figure 34 - EnumerationValue Element XML Schema.....	59
Figure 35 - MaxFloat Element XML Schema.....	59
Figure 36 - MaxInteger Element Xml Schema.....	60
Figure 36 - MaxLength Element XML Schema.....	60
Figure 37 - MinFloat Element XML Schema.....	60
Figure 38 - MinInteger Element XML Schema.....	61
Figure 39 - Qualifier Element XML Schema.....	61
Figure 40 - QualifierDeclaration Element XML Schema.....	62
Figure 41 - QualifierDescription Element XML Schema.....	63
Figure 42 - QualiferSet Element XML Schema.....	64
Figure 43 - Query Element XML Schema.....	65

Figure 44 - QueryResult Element XML Schema.....	67
Figure 45 - ServiceDataModel Element XML Schema	71
Figure 46 - ServiceDataModelProfile Element XML Schema	71
Figure 47 - UniqueQualifier Element XML Schema.....	72
Figure 48 - UniqueQualifier Element XML Schema.....	73

LIST OF TABLES

Table 1- XML Namespace Declarations.....	14
Table 2 - GIS Specific Message Types	14
Table 3 - ListSupportedFeaturesResponseType core:Callout @message values	20
Table 4- List ADS Registration Inquiry Granularity Control	24
Table 5- NotificationRegistrationRequest core:Callout @message values	29
Table 6 - Notification @noticeType Values	31
Table 7 - GIS Elementary Element Details.....	49
Table 8 - Advanced Query Languages.....	50
Table 9 - Allowed QualifierDescription Element @valueType Values	63
Table 10 - QueryResult Element Truth Table.....	69
Table 11 - QueryFilterOpTypeEnumeration Values.....	75
Table 12 - GIS Regular Expressions.....	77
Table 13 - Regular Expression Examples	77
Table 14 - StatusCode Details.....	78
Table 15 - Part 2 and Part 8 StatusCode Detail Usage.....	79

LIST OF EXAMPLES

Example 1- SISListSupportedFeaturesRequest Message Definition.....	15
Example 1-bis - BasicQueryFilter Element	76
Example 1-ter - BasicQueryFilter Element.....	76
Example 2 - BasicQueryFilter Element	80
Example 3 - BasicQueryFilter Element	80
Example 4 - BasicQueryFilter Element	81
Example 5 - Query Element.....	81
Example 6 - QueryResult Element.....	81
Example 7 - Query Element using UniqueQualifer	82
Example 8 - Query Element.....	82
Example 9 - QueryResult Element.....	82
Example 10 - AdvancedQueryFilter Element	83
Example 11 - QueryResult Element.....	83
Example 12 - CreateCursorRequest Message	84
Example 13 - CreateCursorResponse Message	84
Example 14 - QueryRequest Message	85
Example 15 - QueryRequest Message	85
Example 16 - CancelCursorRequest Message	85
Example 17 - CancelCursorResponse Message.....	86
Example 18 - ListSupportedFeaturesRequest Message	87
Example 19 - ListSupportedFeaturesResponse Message.....	87
Example 20 - ListSupportedFeaturesResponse Message.....	88
Example 21 - ListSupportedFeaturesResponse Message.....	88
Example 22 - ListQualifiersRequest Message	88
Example 23 - ListQualifiersResponse Message.....	89
Example 24 - QueryRequest Message	90
Example 25 - QueryResponse Message.....	90
Example 26 - QueryRequest Message	90
Example 27 - QueryResponse Message.....	91
Example 28 - QueryRequest Message	91
Example 29 - QueryResponse Message.....	92
Example 30 - NotificationRegistrationRequest Message	92
Example 31 - NotificationRegistrationRequest Message	93
Example 32 - Notification Message.....	93

Digital Program Insertion–Advertising Systems Interfaces: Part 8 – General Information Service

Introduction

Several logical services defined by the J.380 sub-series of Recommendations provide registration, query and notification services to service consumers.

Since the semantic elements of a logical service interface providing these functions are common to more than one logical service, the specification of these common semantic elements has been factored out into this separate General Information Service (GIS) specification.

For example, the Placement Opportunity Information Service (POIS) and the Subscriber Information Service (SIS) specifications may incorporate the registration, query and notification components specified herein.

Figure 1 provides an abstract representation of the GIS interface that may be implemented by a J.380 logical service.

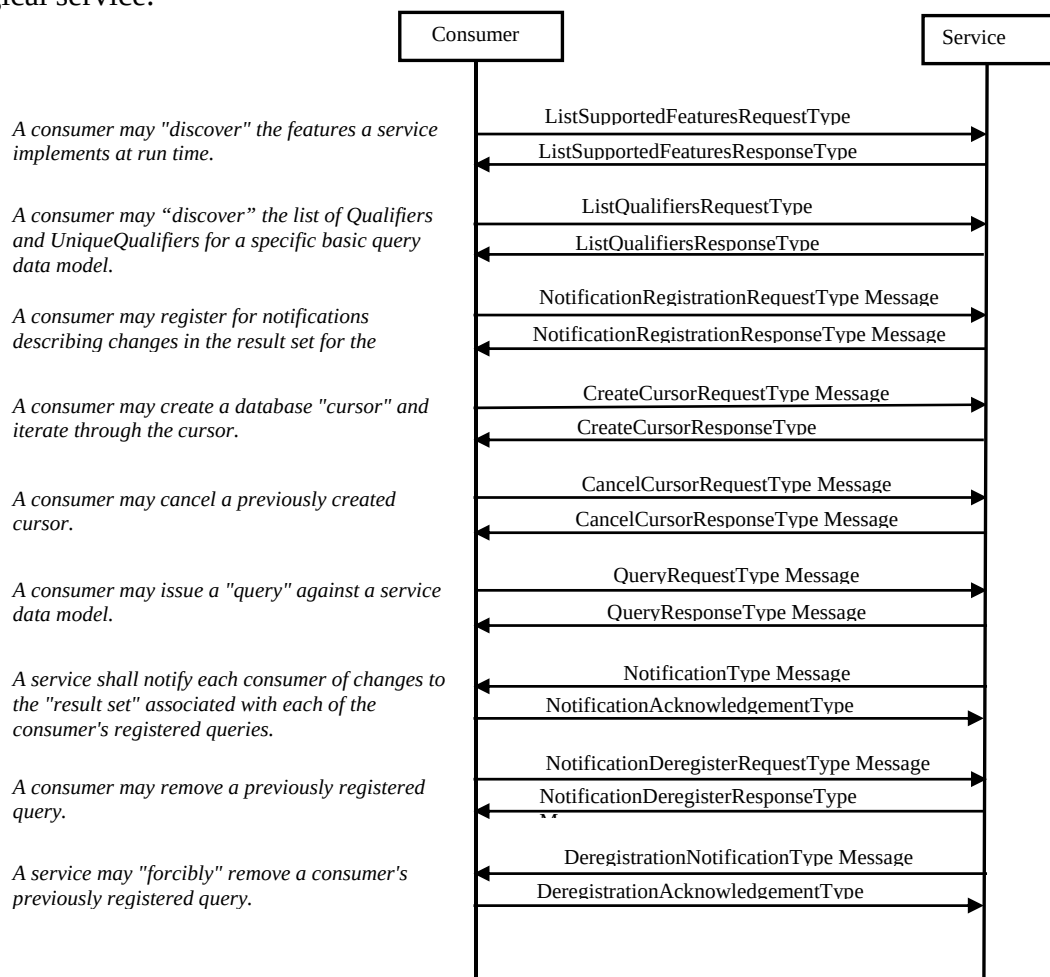


Figure 1 - General Information Service Interface

The GIS interface may be implemented by a logical service that provides data to other logical services. This specification describes the messaging that may be used by a consumer to retrieve data by querying one or more data models supported by a logical service.

Since a logical service's data model may be an abstract representation of a large data store, a consumer may optionally need to locally cache data for performance optimization. The GIS specification provides support for consumer local data caching by allowing the consumer to "register" notification queries with a logical service. In this case, the result set for each query shall be returned to the consumer via "notification" messages reflecting differences (add, delete, or modification). Changes in the logical service's underlying data store while the registered query is "active" shall result in notification messages whose content reflects the changes.

Two kinds of query interfaces may be used to discover information about a logical services' data – a "basic query" and an "advanced query".

Basic queries leverage a limited name/value regular expression grammar. No specialized knowledge of advanced query languages is needed when formulating a basic query. An advanced query is formulated using XQuery or XPath or any other query language.

A logical service may choose to implement either the basic query interface or the advanced query interface or both. See Clause 8.3 for additional information on advanced query language support and Clause 8 for additional information on basic and advanced query mechanisms.

Logical services implementing the interface described herein may support more than one service data model and each service data model may be queried with either the basic query mechanism or the advanced query mechanism or both. When more than one service data model is supported, the first service data model listed in the ListSupportedFeaturesResponse message type is referred to as the default data model (i.e., the first ServiceDataModelProfile element is distinguished as the default service data model). The default data model shall be used when no service data model is specified.

A service data model which may be queried using the basic query mechanism shall be constrained to describing a set of objects decorated with one or more name/value pairs – called "Qualifiers". An additional constraint on the basic query data model is each object in the data model shall be uniquely identifiable by at least one subset of its name/value pairs – called the "UniqueQualifier". The unique qualifier composition is described using the UniqueQualifierDeclaration element which provides the characteristic name identifier of the name/value pair using a sequence of QualifierDeclaration elements. See Clause 8.26 and Clause 8.18 for additional information on the UniqueQualifierDeclaration and QualifierDeclaration elements respectively.

As a concrete example of a service data model that could be easily queried using the basic query interface, consider a Subscriber Information Service (SIS) providing information to other logical services about subscribers. In this case, data representing each subscriber might be comprised of name/value pairs representing demographic information (such as age, income and preferences), location information (such as zip code, service group and street address), and equipment information (such as set-top type and MACAddress). The UniqueQualifier element for a subscriber might be comprised of one Qualifier element – the subscriber's MACAddress.

A service data model may have more than unique qualifier (i.e., more than one UniqueQualifierDeclaration element). When a service data model has more than one UniqueQualifierDeclaration element, each UniqueQualifierDeclaration element shall have a @uniqueQualifierName attribute and no two UniqueQualifiersDeclarations shall use the same string for the @uniqueQualifierName attribute. Furthermore, the first UniqueQualifierDeclaration element listed in the BasicQueryDataModelDescription element shall be referred to as the default unique qualifier declaration (or default unique qualifier). The default unique qualifier declaration shall be used when no specific unique qualifier declaration is referenced. See clauses 8.26 and 8.19 for additional information on the UniqueQualifierDeclaration and QualifierDeclaration elements respectively.

This specification also describes cursor based operations for both basic and advanced queries. Consumers may request the creation of temporary static, cursor based information with specified

lifetimes, and then iterate over the information until the cursor life cycle completes (i.e., expires). See clause 8.11 for additional information on cursors.

The GIS specification includes a notification model with associated notification management functions such as registration, deregistration and listing of active registrations. The GIS specification does not restrict the number of data stores that a logical service may use.

1 Scope

This document describes the Digital Program Insertion Advertising Systems Interfaces' General Information Service (GIS) messaging and data type specification using XML, XML Namespaces, and XML Schema.

2 References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T J.380.2]	ITU-T J.380.2 (2011), Digital Program Insertion-Advertising Systems Interfaces: Part 2 - Core Data Elements
[ITU-T J.380.7]	ITU-T J.380.7 (2011), Digital Program Insertion-Advertising Systems Interfaces: Part 7 - Message Transport
[W3C–XPath]	W3C Recommendation Xpath (1999), <i>XML Path Language (XPath) Version 1.0</i> .
[W3C–XQuery]	W3C Recommendation XQuery (2007), <i>An XML Query Language (XQuery) Version 1.0</i> .
[SCTE 130-8 Schema]	SCTE 130-8 Schema file

3 Definitions

Throughout this standard the terms below have specific meanings. Because some of the terms are defined in other J.380 documents having very specific technical meanings, the reader is referred to the original source for their definition. For terms defined by this standard, brief definitions are given below.

All [ITU-T J.380-2] definitions are included herein. See [ITU-T J.380.2] for additional information.

3.1 Terms defined elsewhere:

This Recommendation uses the following terms defined elsewhere: None

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1. Advanced Query: The "Advanced Query" interface defined by J.380.8 permits the consumer of a logical service implementation derived from J.380.8 to use an advanced query language to formulate queries against a logical service's data model.

3.2.2. AdvancedQueryLanguage: As used in this document, the term “Advanced Query Language” refers to any language used to formulate queries against a logical service’s data model using the advanced query interface. XPath and XQuery are each examples of an advanced query language.

3.2.3. AdvancedQueryFilter: An “Advanced Query Filter” is a collection of free form data items that individually represent complete query terms for a given query language. The individual terms are additively applied (ANDed) together during a query operation against a specific service data model, which results in the identification of a collection of objects contained within the data store.

3.2.4. Basic Query: To obtain information from a logical service implementation derived from J.380.8, a logical service consumer issues a "query" against the data. The "Basic Query" interface is based on an exchange of name/value pairs, referred to as qualifiers, and requires no specialized knowledge of advanced query languages such as XQuery.

3.2.5. BasicQueryFilter: A “Basic Query Filter” is a collection of name and value pairs additively applied (ANDed) together during a basic query operation against a specific service data model, which results in the identification of a collection of objects contained within the data store.

3.2.6. Character data (CDATA): XML data that is not parsed. CDATA carries markup examples that would otherwise be interpreted as XML because of the tags.

3.2.7. Cursor: A temporary construct containing static data. Consumers of logical services implementing the GIS interface may create and access cursor information using the standard query mechanisms described in this document.

3.2.8. Data Model: A Data Model is a formal view of the data items contained in an information store to which an information service implementing this standard will provide access and is specified for purposes of formulating and executing queries against the information store's data. This standard specifies one data model that may be used for querying a logical service's information store with this standard's "Basic Query" interface. More complex data models may be specified independently of this standard and may be queried with this standard's "Advanced Query" interface. In this latter case, the mechanisms by which a consumer incorporates a data model specification so that meaningful queries may be issued against it are outside the scope of this standard.

3.2.9. Default Data Model: The data model that is used when no specific data model is provided. The default data model typically occupies the first location in a data model sequence (i.e., a list of data models).

3.2.10. Qualifier: A “Qualifier” is a name/value pair used to describe one characteristic of an object in a logical service's basic query data model. For instance, <Qualifier name="Age" value="30to40"/> is an example of a qualifier where "Age" is the characteristic’s name and "30to40" is the characteristic’s value.

3.2.11. QualifierSet: A “QualifierSet” is a complete set of Qualifier elements that describe an object in a logical service's basic query data model.

3.2.12. UniqueQualifier: A “Unique Qualifier” is a set of one or more Qualifier elements that – taken together – uniquely identify an object in a logical service's basic query data model. To "uniquely identify" an object means that no other object in the data store has the same UniqueQualifier. However, an object may have more than one UniqueQualifier.

3.2.13. UniqueQualifierDeclaration: The “Unique Qualifier Declaration” defines the set of Qualifier element characteristic name identifiers comprising a unique qualifier. When each specified Qualifier element’s named characteristic identifier is paired with a value (i.e., a name/value pair which is a Qualifier element), the result is a service data model UniqueQualifier.

4 Abbreviations and acronyms

All [ITU-T J.380.2] abbreviations are included herein. See [ITU-T J.380.2] for additional information.

This Recommendation uses the following abbreviations and acronyms:

CDATA Character DATA

5 Conventions

5.1 Notational Conventions

5.1.1 Normative XML Schema

J.380.8 employs the same notational conventions as [ITU-T J.380.2]. Refer to [ITU-T J.380.2] for an explanation of notational conventions.

5.1.2 Document Conventions

J.380.8 employs the same document conventions as [ITU-T J.380.2]. Refer to [ITU-T J.380.2] for an explanation of document conventions. For example, the XML schema illustration is explained there.

This specification utilizes XML substitution groups for additional extensibility. XML substitution groups designate elements as substitutes for other element declarations without changing the original schema documents. Within this document, substitutable elements are graphically identified using the following illustrative technique.

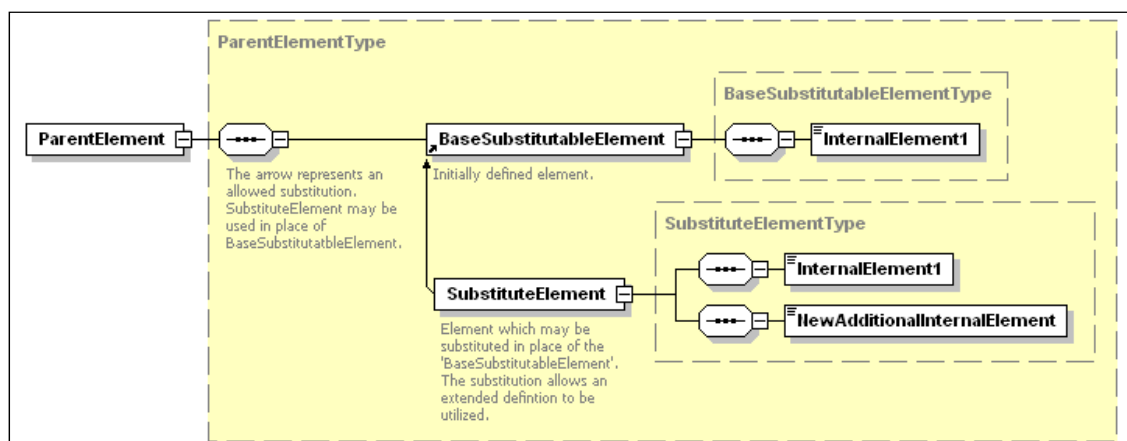


Figure 2 - Substitution Group Schema Convention

In Figure 2, the element referred to as “SubstituteElement” may be used in place of the element named “BaseSubstitutableElement” provided the XML namespace declarations are included in the document as per [W3C-XSD]. The diagram’s illustrative arrow signals the reader of the possible element substitution.

5.2 Processing Conventions

Unknown/Unrecognized/Unsupported XML Elements and Attributes. See [ITU-T J.380.2] for information.

5.3 XML Namespaces

This specification uses the ‘gis’ prefix, as described in Table 1, for the interface associated with the specific XML namespace URI that shall be used by all implementations. Table 1 lists the prefix, the corresponding namespace, and a description of the defining specification used herein.

Table 1- XML Namespace Declarations

Prefix	Namespace	Description
core	http://www.scte.org/schemas/130-2/2008a/core	See [ITU-T J.380.2]
gis	http://www.scte.org/schemas/130-8/2010a/gis	ITU-T J.380.8 (this Recommendation)
xsd	http://www.w3.org/2001/XMLSchema	XML foundation. See W3C.

Unless otherwise stated, all references to XML elements illustrated in this document are from the 'gis' namespace. Elements from other namespaces shall be prefixed with the name of the external namespace, e.g. <core:XXX>.

6 GIS Message TYPES

The following topics are covered by [ITU-T J.380.2] and this specification considers all aspects defined therein to be normative and applicable herein. See [ITU-T J.380.2] for additional information.

- Message format
- XML message carriage
- Transport mechanisms
- Message error handling

The GIS message interface shall include the messages defined in [ITU-T J.380.2] and logical services that incorporate the GIS message interface shall conform to the messaging transport mechanism described in [ITU-T J.380.7].

Table 2 identifies additional J.380.8 (GIS) specific message types.

Table 2 - GIS Specific Message Types

Message	Description
ListSupportedFeaturesRequestType	Request to retrieve a list of a logical service's supported features
ListSupportedFeaturesResponseType	Response to ListSupportedFeaturesRequest
ListQualifiersRequestType	Request to retrieve a list of names that may be used to construct basic queries using name/value pairs.
ListQualifiersResponseType	Response to ListQualifiersRequest
ListNotificationRegistrationRequestType	Request to list existing registrations
ListNotificationRegistrationResponseType	Response to ListNotificationRegistrationRequest
NotificationRegistrationRequestType	Registration request for notification
NotificationRegistrationResponseType	Response to NotificationRegistrationRequest
NotificationType	Notification message indicating a change to the result set of a registered query
NotificationAcknowledgementType	Response to Notification
CreateCursorRequestType	Request to create a cursor
CreateCursorResponseType	Response to CreateCursorRequest
CancelCursorRequestType	Request to cancel an existing cursor
CancelCursorResponseType	Response to CancelCursorRequest
QueryRequestType	Request to acquire records from the GIS

Message	Description
QueryResponseType	Response to QueryRequest
NotificationDeregisterRequestType	Request to de-register a previously accepted registration
NotificationDeregisterResponseType	Response to NotificationDeregisterRequest
DeregistrationNotificationType	Deregistration notification
DeregistrationAcknowledgementType	Deregistration notification acknowledgement

Because this specification defines an abstract messaging interface used by other logical service implementations, this specification defines the message types rather than message elements. Every logical service implementing the interface specified by J.380.8 shall define a set of message elements semantically equivalent to the message types defined in J.380.8 by sub-classing the message types listed in Table 2 and by assigning a new message element name to each message formed by prefixing a string representing the logical service type to the message type name.

For example, the Subscriber Information Service (SIS) schema definition might contain this definition for the ListSupportedFeaturesRequest message.

```
<xsd:element name="SISListSupportedFeaturesRequest" type="gis:ListSupportedFeaturesRequestType"/>
```

Example 1- SISListSupportedFeaturesRequest Message Definition

The following subclauses provide specific information on J.380.8's use of attributes and message types defined in [ITU-T J.380.2]. For detailed information on these attributes and message types see [ITU-T J.380.2].

6.1 @version Attribute

J.380.8 specifies message types with the intent that other logical services may create service specific concrete message definitions using the J.380.8 message types. Thus, J.380.8 shall not specify a value for the @version attribute. Logical services that derive messages from the J.380.8 message types shall set the @version attribute to a value that reflects the logical service's revision.

6.2 Request Base Message Type

All GIS top level **request** message types are derived from the core:Msg_RequestBaseType abstract base message type. See [ITU-T J.380.2] for details on the attributes and elements contained in this base message.

6.3 Response Base Message Type

All GIS top level **response** message types are derived from the core:Msg_ResponseBaseType abstract base message type. See [ITU-T J.380.2] for details on the attributes and elements contained in this base message.

6.4 Notification Base Message Type

All GIS top level **notification** message types are derived from the core:Msg_NotificationBaseType abstract base message type. See [ITU-T J.380.2] for details on the attributes and elements contained in this base message.

6.5 Acknowledgement Base Message Type

All GIS top level **acknowledgement** message types are derived from the core:Msg_AcknowledgementBaseType abstract base message type. See [ITU-T J.DPI-ASI 2] for details on the attributes and elements contained in this base message.

6.6 Messages Requiring Notification Registration

Consumers are required to register with any logical service implementing the GIS interface in order to receive Notification messages. The logical service shall not send a NotificationDeregisterRequest message type to any consumer when the consumer has no notification requests registered with the logical service. All other GIS message types do not require registration and may be sent at any time.

6.7 Default Logical Service Channel End-Point

A logical service incorporating the GIS interface shall support the receipt of the ListSupportedFeaturesRequestType message type. Typically, this message type is received on a default (or well known) logical service channel end-point address. Other GIS messages types may also be offered on the same logical service channel end-point or they may be optionally offered on additional logical service channel end-points. In order to discover all GIS provided logical service channel end-points, consumers should complete a ListSupportedFeaturesRequestType/ListSupportedFeatureResponseType message type exchange. See clause 7.1.2 for additional information on the ListSupportedFeaturesResponseType message type.

See [ITU-T J.380.2] for a complete description of the terms (message, end-point and logical service channel).

7 GIS Message Exchange

The following diagram illustrates a typical message exchange between a GIS consumer and the GIS.

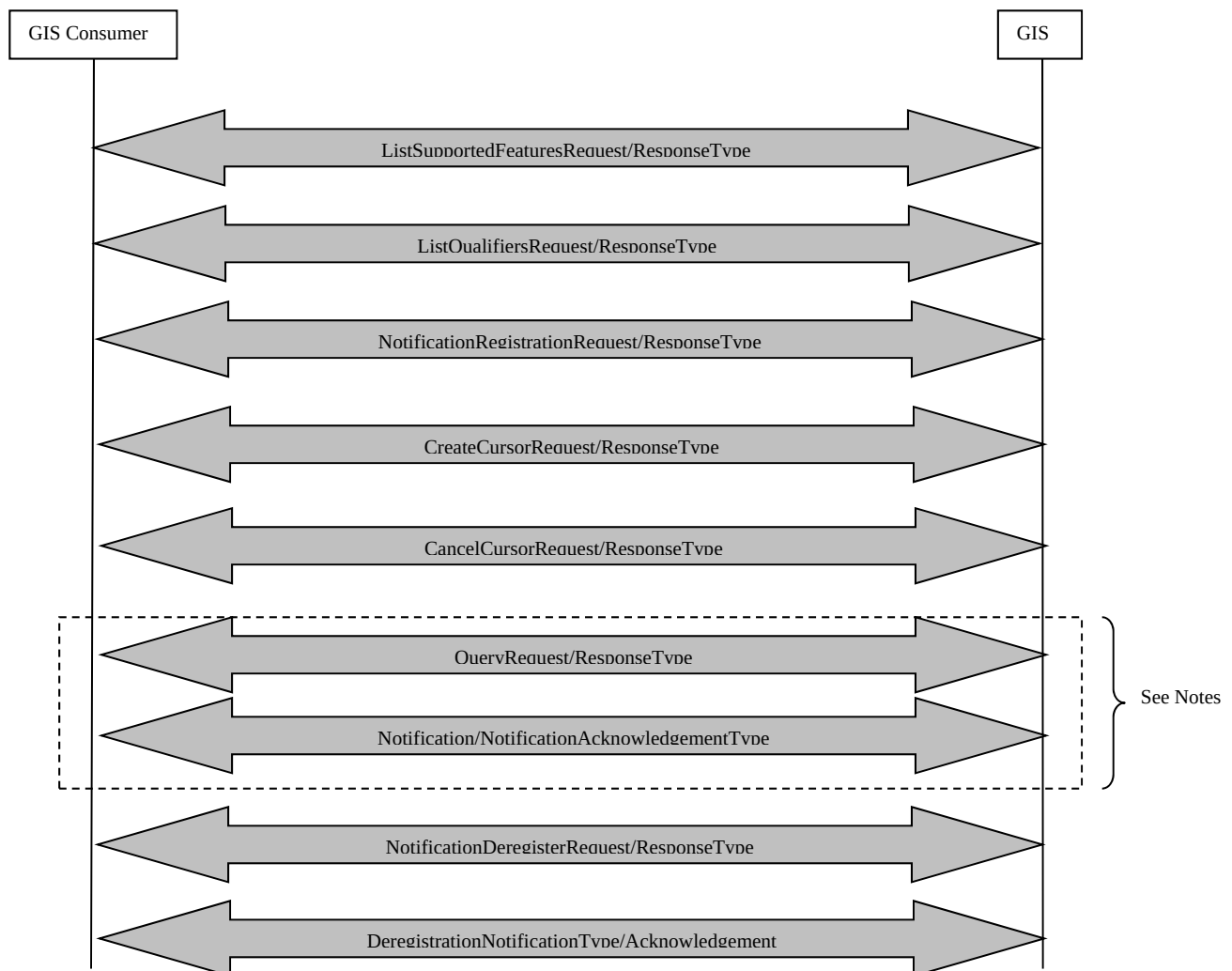


Figure 3 - GIS Message Exchange

Figure 3 illustrates all of the message exchanges that are specific to the GIS. The service check and service status message exchanges defined in [ITU-T J.380.2] are not depicted in this illustration.

7.1 ListSupportedFeaturesRequest and ListSupportedFeaturesResponse Message Types

The ListSupportedFeaturesRequest and ListSupportedFeaturesResponse message types allow consumers to inquire about the service data models and advanced query languages supported by a logical service.

7.1.1 ListSupportedFeaturesRequest Message Type

The ListSupportedFeaturesRequest message type allows the consumer of a logical service to inquire about the service data models and advanced query languages supported by the service.

The XML schema definition for this message is illustrated in Figure 4.

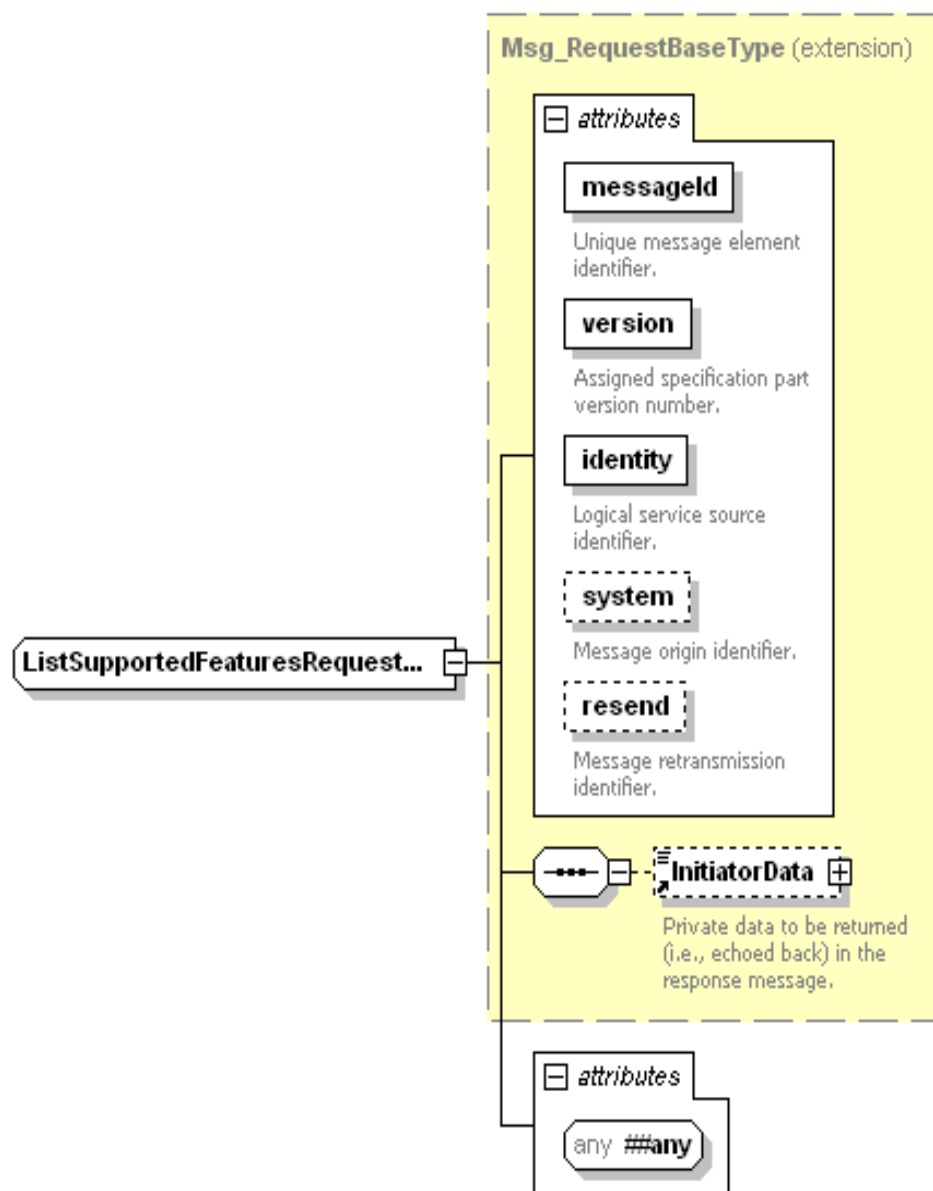


Figure 4 - ListSupportedFeaturesRequestType XML Schema

The ListSupportedFeaturesRequest message type defines no elements in addition to those that are already defined by core:Msg_RequestBaseType.

@##any[Optional] - The @##any attribute enables a logical service incorporating the J.380.8 interface to optionally extend concrete messages derived from J.380.8 message types with attributes not specified by the schema.

7.1.2 ListSupportedFeaturesResponse Message Type

If a ListSupportedFeaturesRequest type message is successful, the matching ListSupportedFeaturesResponse message type shall contain, at a minimum, a single core:Callout element containing one or more core:Address element(s).

If a ListSupportedFeaturesRequest type message is not successful, the matching ListSupportedFeaturesResponse message type shall not contain a core:Callout element.

See [ITU-T J.380.2] for additional information on the core:Callout element. The XML schema definition for this message is shown in Figure 5.

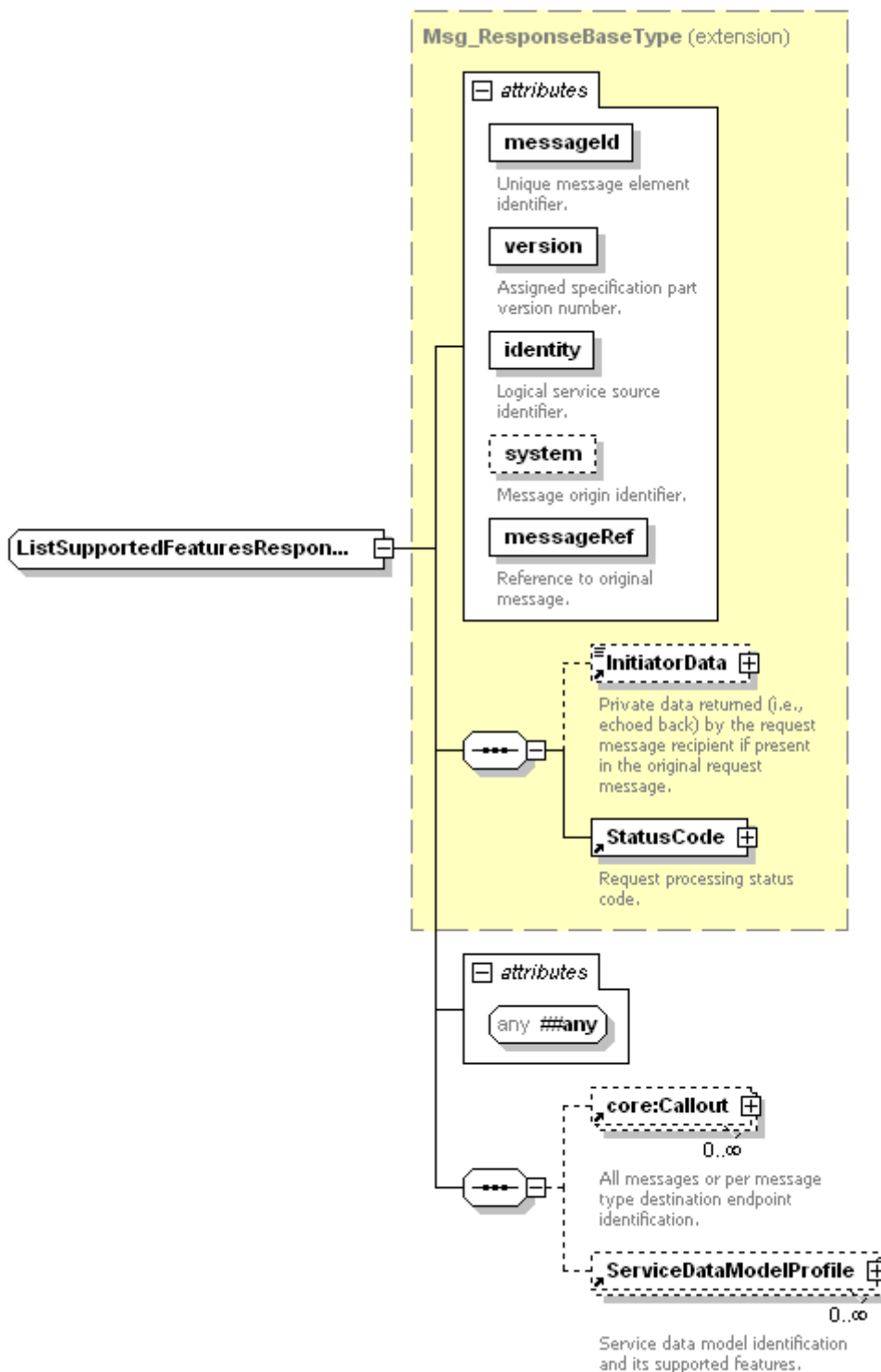


Figure 5 - ListSupportedFeaturesResponseType XML Schema

The ListSupportedFeaturesResponse message type is derived from the core namespace base type core:Msg_ResponseBaseType and defines the following attributes and elements in addition to those already defined by core:Msg_ResponseBaseType.

@##any[Optional] - The @##any attribute enables a logical service incorporating the J.380.8 interface to optionally extend concrete messages derived from J.380.8 message types with attributes not specified by the schema.

core:Callout [Optional] – Zero or more core:Callout elements specifying the service channel message end-point(s). See [ITU-T J.380.2] for additional information on the core:Callout element.

Table 3 contains the values for the @message attribute of the core:Callout element. Values for the @message attribute should be used exactly as defined in this table where "xxx" is a string representing the logical service as shown in Example 1.

Table 3 - ListSupportedFeaturesResponseType core:Callout @message values

@message Attribute Value	Description
xxxNotificationRegistrationRequest	Destination end-point for message of type NotificationRegistrationRequestType
xxxNotificationDeregisterRequest	Destination end-point for message of type NotificationDeregisterRequestType
xxxListNotificationRegistrationRequest	Destination end-point for message of type ListNotificationRegistrationRequestType
xxxListQualifiersRequest	Destination end-point for message of type ListQualifiersRequestType
xxxCreateCursorRequest	Destination end-point for message of type CreateCursorRequestType
xxxCancelCursorRequest	Destination end-point for message of type CancelCursorRequestType
xxxQueryRequest	Destination end-point for message of type QueryRequestType
ServiceStatusNotification	Destination end-point for message of type core:ServiceStatusNotificationType
...	User defined address endpoint outside of the scope of this specification. The string shall be prefixed with the text "private:".

All message values listed in Table 3 and not present in the ListSupportedFeaturesResponseType message's core:Callout XML element sequence shall be available through the default endpoint if present. (The default endpoint is identified by a core:Callout element not having the @message attribute.) See [ITU-T J.380.2] for additional information.

ServiceDataModelProfile[Optional] – A list of zero or more ServiceDataModelProfile elements – one for each service data model supported by the logical service. The first ServiceDataModelProfile element is referred to as the default service data model and shall be used when no service data model is specified. For more information on the ServiceDataModelProfile element see clause 8.24.

7.2 ListQualifiersRequest and ListQualifiersResponse Message Types

The ListQualifiersRequest and ListQualifiersResponse message types allow consumers to discover the qualifiers associated with service data models that may be queried using the basic query interface. If a specific service data model does not support the basic query interface, the ListQualifiersResponse message type shall return an error (specified later in this clause).

7.2.1 ListQualifiersRequest Message Type

The ListQualifiersRequest message type allows the consumer of a logical service that implements the GIS interface to inquire about the unique qualifiers, the qualifier names and the qualifier descriptions used by a logical service's data model.

The XML schema definition for this message is shown in Figure 6.

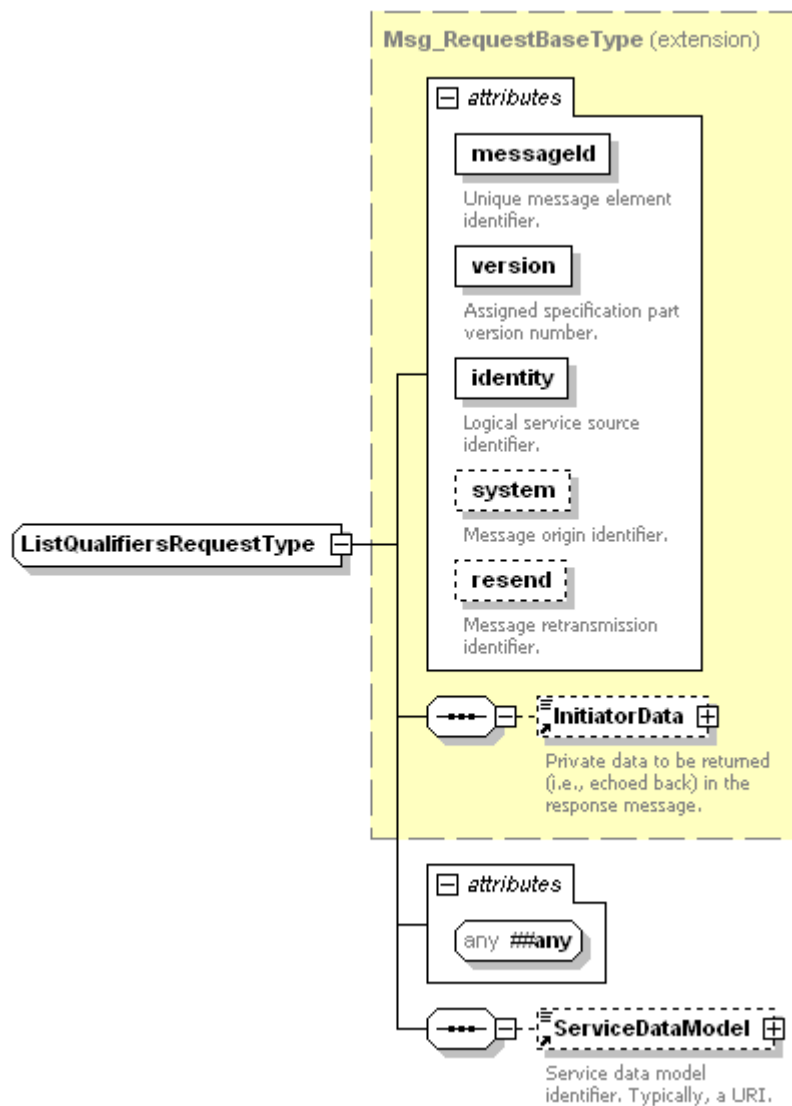


Figure 6 - ListQualifiersRequestType XML Schema

The ListQualifiersRequest message type defines the following attributes and elements in addition to those that are already defined on the abstract base message core:Msg_RequestBaseType.

@##any[Optional] - The @##any attribute enables a logical service incorporating the J.380.8 interface to optionally extend concrete messages derived from J.380.8 message types with attributes not specified by the schema.

ServiceDataModel [Optional] – The ServiceDataModel element contains a unique identifier (usually a URI) for a service data model that may be accessed using the basic query interface. If the ServiceDataModel element is not present, the default service data model shall be selected. For additional information on the ServiceDataModel element, see clause 8.23

7.2.2 ListQualifiersResponse Message Type

If the service data model specified by the ServiceDataModel element in the ListQualifiersRequest message type is known to the logical service and is a basic query data model, the matching ListQualifiersResponse message type shall contain a single BasicQueryDataModelDescription element describing the service data model. If the service data model is known but does not support a basic query interface (i.e., it only supports advanced queries), the ListQualifiersResponse message type shall return an error where the core:StatusCode element shall contain a @detailCode attribute set to the value core:NotSupported.

If the service data model specified by the ServiceDataModel element in the ListQualifiersRequest message is not known to the logical service, the matching ListQualifiersResponse message type shall not contain a BasicQueryDataModelDescription element and the core:StatusCode element shall contain a @detailCode attribute set to the value core:ResourceNotFound.

The XML schema definition for this message is illustrated in Figure 7.

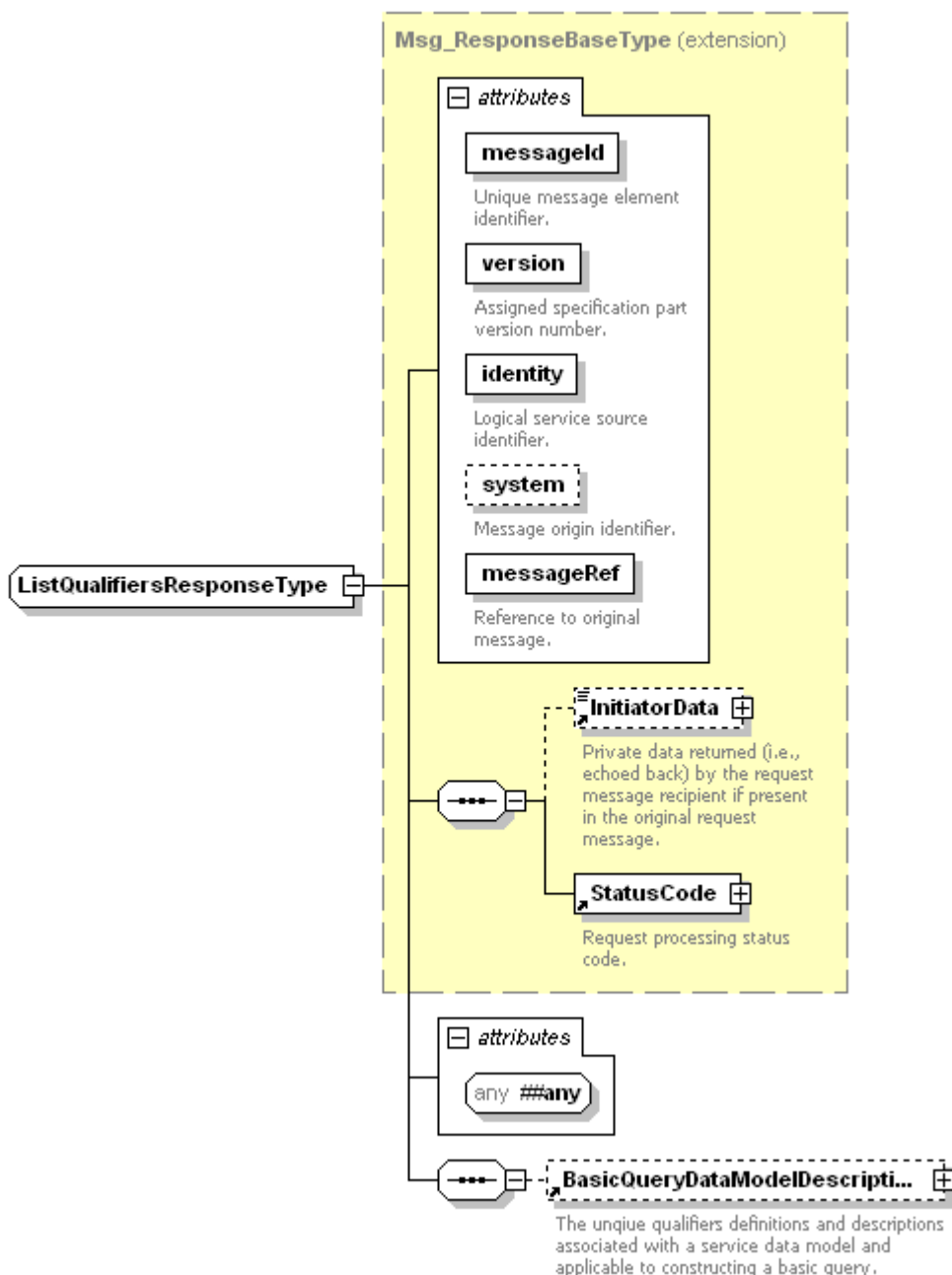


Figure 7 - ListQualifiersResponseType XML Schema

The ListQualifiersResponse message type is derived from the core namespace base type core:Msg_ResponseBaseType and defines the following attributes and elements in addition to those that are already defined on the abstract base message core:Msg_ResponseBaseType.

@##any[Optional] - The @##any attribute enables a logical service incorporating the J.380.8 interface to optionally extend concrete messages derived from J.380.8 message types with attributes not specified by the schema.

BasicQueryDataModelDescription [Optional] – The BasicQueryDataModelDescription element defines the service data model’s unique qualifiers and provides descriptions for a group of qualifiers. For more information on the BasicQueryDataModelDescription element see clause 8.7.

7.3 ListNotificationRegistrationRequest and ListNotificationRegistrationResponse Message Types

A GIS consumer may inquire about current notification registrations by using the ListNotificationRegistrationRequest message type. The logical service shall respond to the ListNotificationRegistrationRequest type message with a ListNotificationRegistrationResponse type message. This message exchange allows a GIS consumer to discover the active queries previously installed by one or more NotificationRegistrationRequest type messages.

There are two list registration granularity levels:

- Per logical service (i.e., via the @identity attribute)
- Per registration message

A logical service may restrict access to registration information by returning a status code equating to “not authorized” for the list registration request on a per logical service basis. The per registration message granularity level shall be supported by all GIS compliant logical services. The per logical service message granularity level may be supported by any GIS compliant logical service.

7.3.1 ListNotificationRegistrationRequest Message Types

The ListNotificationRegistrationRequest message may be issued to a logical service to retrieve information about active notification registrations.

The XML schema definition for this message is illustrated in Figure 8 .

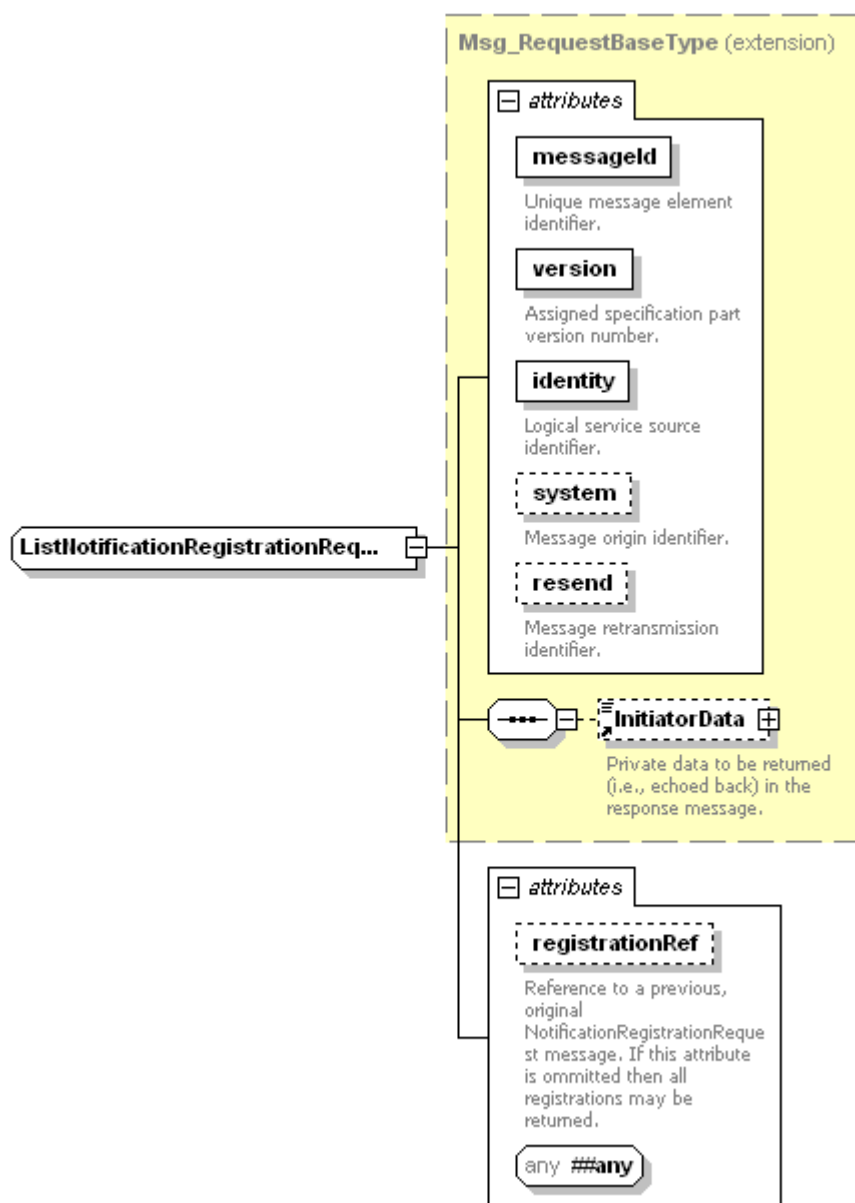


Figure 8 - ListNotificationRegistrationRequestType XML Schema

The ListNotificationRegistrationRequest message returns all active registrations associated with a specific logical service source as identified by the @identity attribute when the @registrationRef attribute is omitted. If only a specific registration is to be returned, both the @identity and the @registrationRef attributes shall be included. Table 4 defines the inclusion relationships.

Table 4- List ADS Registration Inquiry Granularity Control

@identity	@registrationRef	Description	Access Restrictions Permitted
Included	Omitted	All registrations matching the supplied @identity value.	Yes
Included	Included	A specific registration matching the @identity and @registrationRef values.	No

The access restrictions permitted column indicates whether the status code equivalent to `core:NotAuthorized` may optionally be returned.

The `ListNotificationRegistrationRequest` message type is derived from the core namespace base type `core:Msg_RequestBaseType` and defines the following attributes and elements in addition to those that are already defined on the abstract base message `core:Msg_RequestBaseType`.

@registrationRef [Optional, core:registrationRefAttrType] – A reference to an original `NotificationRegistrationRequest` type message. This attribute instructs the logical service to list only the notification registration request identified by this reference. The value shall be the `NotificationRegistrationRequest` message's original `@messageId` attribute value. If this attribute is not present, then the matching `ListNotificationRegistrationResponse` type message shall contain information about all notifications registered for the consumer as identified by the `@identity` attribute.

@##any[Optional] - The `@##any` attribute enables a logical service incorporating the J.380.8 interface to optionally extend concrete messages derived from J.380.8 message types with attributes not specified by the schema.

7.3.2 ListNotificationRegistrationResponse Message Type

The `ListNotificationRegistrationResponse` message type is the response pair to the previously defined `ListNotificationRegistrationRequest` message type.

The XML schema definition for this message is shown in Figure 9 .

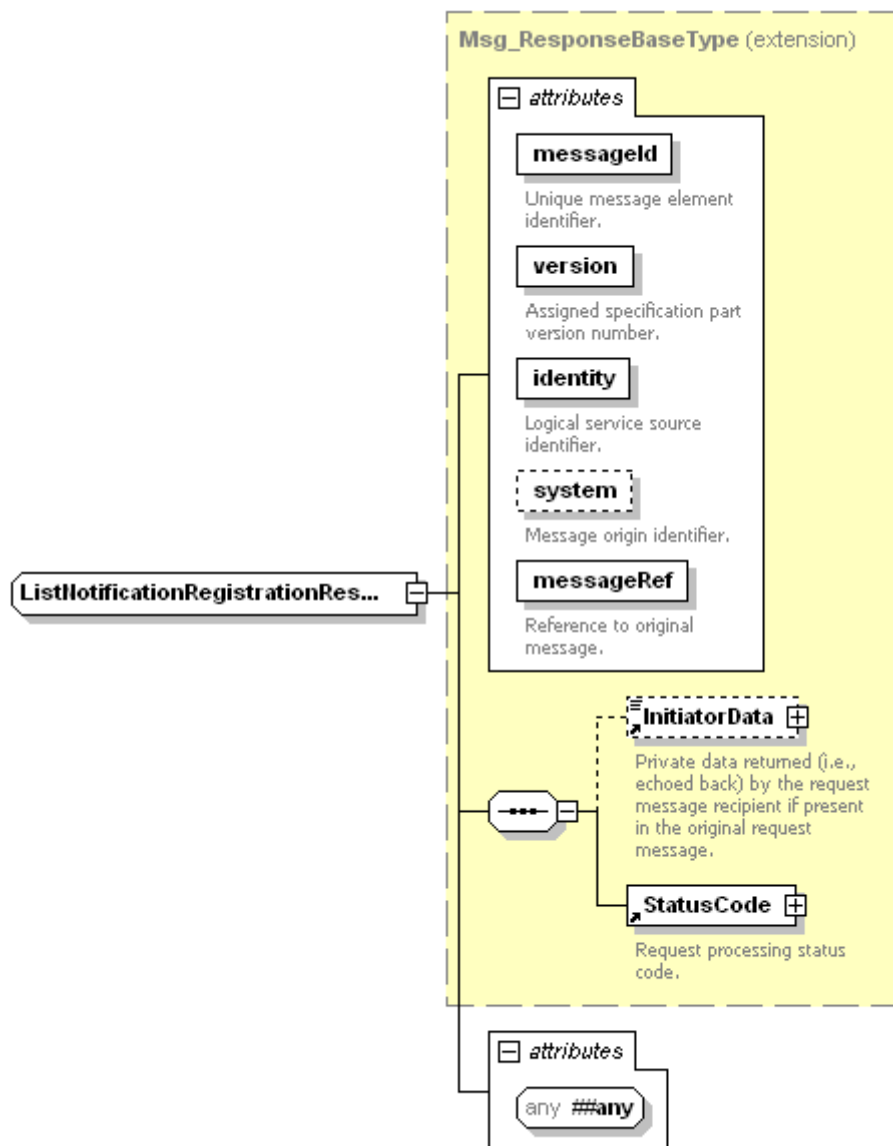


Figure 9 - ListNotificationRegistrationResponseType XML Schema

The ListNotificationRegistrationResponse message type is derived from the core namespace base type core:Msg_ResponseBaseType and defines the following attributes and elements in addition to those that are already defined on the abstract base message core:Msg_ResponseBaseType.

@##any[Optional] - The @##any attribute enables a logical service incorporating the J.380.8 interface to optionally extend concrete messages derived from J.380.8 message types with attributes not specified by the schema.

A logical service incorporating the GIS NotificationRegistrationRequest message type shall also include the GIS ListNotificationRegistrationRequest and GIS ListNotificationRegistrationResponse message types (typically defined as the xxxListNotificationRegistrationRequest and xxxListNotificationRegistrationResponse messages by the logical service). The xxxListNotificationRegistrationResponse message, which shall extend the GIS ListNotificationRegistrationResponse message type, shall include the logical service's xxxNotificationRegistrationRequest message as part of the element type definition and the inclusion should be specified as a sequence of zero or more instances of the xxxNotificationRegistrationRequest message.

The xxxListNotificationRegistrationResponse message type's xxxNotificationRegistrationRequest shall be a recoded copy of the accepted registration message. The message element order does not

convey any information (e.g., element order does not reflect registration order). See [ITU-T J.380.2] for compliance requirements.

If the sub-classed ListNotificationRegistrationResponse message type is unable to locate the specific registration as specified by the ListNotificationRegistrationRequest message type's @registrationRef attribute or by the @identity attribute, the response message type's core:StatusCode element shall contain the value core:ResourceNotFound (i.e., a response of zero located registrations shall return an error).

7.4 NotificationRegistrationRequest and NotificationRegistrationResponse Message Types

A logical service that implements the GIS interface shall support registration for notification message delivery. The NotificationRegistrationRequest message type allows a consumer to specify notification interests relative to a basic or an advanced query.

On receipt of an update, addition or deletion event from its underlying data store affecting the result set of a registered query, the logical service shall send a Notification type message to each matching registered consumer.

7.4.1 NotificationRegistrationRequest Message Type

The NotificationRegistrationRequest message type allows a consumer to specify a set of notification interests by registering a query against a logical service's data model. These registered queries shall be examined by the logical service relative to changes in any data relevant to the query. If any change to the data causes a change to the query result for a registered query, a notification containing the new result shall be sent to the consumer in the form of a Notification message type.

The XML schema representation of the NotificationRegistrationRequest message type is illustrated in Figure 10 .

Figure 10: XML schema representation of the NotificationRegistrationRequest message type