

e-MS Exchange Protocol (e-MSEP)

Version 1.0 – 28 March 2005

Document identifier:

e-MSEP-1.0

Location:

<http://www.e-ms.ca/documentation.php>

Editors:

Brad Head, Sierra Systems Group, Inc. <BradHead@SierraSystems.com>

Karen Kuhn, Vancouver Island Health Authority <Karen.Kuhn@viha.ca>

Contributors:

James Angus, Vancouver Island Health Authority <James.Angus@viha.ca>

Kevin Routley, Vancouver Island Health Authority <Kevin.Routley@viha.ca>

Frieda Kaiser, Vancouver Island Health Authority <Frieda.Kaiser@viha.ca>

Jesse McDowell, Wolf Medical Systems <Jesse@wolfmedical.com>

Lorne Trudeau, Number 41 Media <Lorne@number41media.com>

James Roland, Sunaptic Solutions <James.Roland@sunaptic.com>

Simon Chester, Sunaptic Solutions <Simon.Chester@sunaptic.com>

Abstract:

The e-MS Exchange Protocol (e-MSEP) is a messaging format for supporting the exchange of electronic medical summaries between care providers. An electronic medical summary (e-MS) is a subset of patient data suitable for communication among primary health care practitioners and other health care providers for the purpose of sharing the care of a patient. The protocol allows a care provider to send and track the delivery of an e-MS document formatted in the XML standard: *HL7 CDA Release 2*.

The e-MSEP provides support for care provider query and the support for delegating access to messages. The protocol is intended to communicate with a messaging broker. Messages are delivered to a drop point (a message box) on the messaging broker. Messages are received by polling the broker periodically. The protocol is intended to be used by Electronic Medical Record (EMR) systems operated by care providers.

Status:

This document is a Specification of the British Columbia e-MS Project sponsored by the Ministry of Health Services and British Columbia Medical Association. It is anticipated that, after further testing and review by EMR vendors during an e-MS system trial, this protocol will be submitted for adoption as a provincial health care standard by the Health Information Standards Council of BC.

This document is updated periodically on no particular schedule. Send comments to the editor.

40 **Table of Contents**

41	1	Introduction	8
42	1.1	Purpose	8
43	1.2	Audience	8
44	1.3	Caveats and Assumptions	8
45	1.4	Terminology.....	8
46	1.5	Notational Conventions	8
47	1.6	Relation to Other Specifications.....	9
48	1.7	Conformance.....	9
49	2	System Overview (non-normative)	10
50	2.1	What the e-MS Exchange Service Does.....	10
51	2.2	Use Cases	11
52	2.3	Basic Messaging Operation	12
53	2.3.1	Retrieving Messages (Polling)	12
54	2.3.2	Sending Messages	13
55	2.4	Examples of Use Scenarios	15
56	2.4.1	Sending a Message	16
57	2.4.2	Receiving Messages.....	16
58	2.4.3	Delegating Access	16
59	2.4.4	Organization Administration.....	17
60	3	e-MS Exchange Service Architecture	18
61	3.1	Overview	18
62	3.2	Design Considerations	18
63	3.3	Request/Response Messaging	18
64	3.4	Command Message Design Pattern	19
65	3.5	Web Services	19
66	3.5.1	Security Considerations.....	20
67	3.5.2	Error Handling.....	22
68	3.5.3	Logical Data Model	22
69	3.6	Mutable and Immutable Key Identifiers.....	23
70	3.6.1	Provider and Organization Addresses	23
71	3.6.2	Unique Identifiers (Immutable).....	24
72	3.6.3	User Names	24
73	3.7	Lifecycle of an e-MS Message	24
74	3.7.1	New Status.....	25
75	3.7.2	Read Status	25
76	3.7.3	Received Status.....	26
77	3.7.4	Delivery Complete Status	26
78	3.7.5	Delivery Failure Status.....	26
79	3.7.6	Delivery Failure Notification	26
80	3.8	Notification Messages	26
81	4	Messaging Services Interface.....	28

82	4.1 Overview	28
83	4.2 Messaging Request	28
84	4.2.1 MessageAck Request.....	28
85	4.2.2 MessageBoxStatus Request	29
86	4.2.3 MessageListing Request	30
87	4.2.4 MessageRead Request	31
88	4.2.5 MessageRecv Request.....	32
89	4.2.6 MessageReset Request	32
90	4.2.7 MessageSend Request	33
91	4.2.8 MessageTrack Request.....	36
92	4.3 Messaging Responses	37
93	4.3.1 MessageAckResponse	37
94	4.3.2 MessageBoxStatusResponse.....	38
95	4.3.3 MessageListingResponse.....	39
96	4.3.4 MessageReadResponse	41
97	4.3.5 MessageResetResponse.....	42
98	4.3.6 MessageRecvResponse	43
99	4.3.7 MessageSendResponse.....	46
100	4.3.8 MessageTrackResponse	47
101	4.4 Messaging Data Types XML Schema.....	49
102	5 Query Services Interface	55
103	5.1 Overview	55
104	5.2 Query Request Messages.....	55
105	5.2.1 Organization Query Request	55
106	5.2.2 Provider Query Request	56
107	5.2.3 User Query Request	58
108	5.3 Query Response Messages.....	59
109	5.3.1 Organization Query Response	59
110	5.3.2 Provider Query Response	61
111	5.3.3 User Query Response	62
112	5.3.4 Query Data Types XML Schema	64
113	6 Administrative Services Interface	67
114	6.1 Overview	67
115	6.1.1 Organization Administration.....	67
116	6.1.2 Provider Administration.....	67
117	6.1.3 User Administration.....	67
118	6.1.4 Preference Settings	67
119	6.2 Administrative Request Message.....	68
120	6.2.1 Organization Administrative Request	68
121	6.2.2 Provider Administrative Request	72
122	6.2.3 User Administrative Request	74
123	6.3 Administrative Response Message.....	75
124	6.3.1 Organization Administrative Response.....	76
125	6.3.2 Provider Administrative Response.....	79

126	6.3.3 User Administrative Response	81
127	6.4 Admin Data Types XML Schema	82
128	7 Error Messages	89
129	7.1 Overview	89
130	7.2 Error Codes	89
131	8 XML Schemas	91
132	8.1 Main eMSEP XML Schema – eMEP.xsd	91
133	8.2 Data Type Schemas – DataTypes.xsd	92
134	9 SOAP Bindings – Web Services	97
135	9.1 Web Service definitions for Exchange Services Protocol	97
136	9.1.1 e-MS Web Service Namespace	97
137	9.2 SOAP Request/Response Example	97
138	9.3 eMSEP Web Services (eMSEP)	98
139	9.3.1 Web Services URL	98
140	9.3.2 Web Service Methods	98
141	9.3.3 URL to view the WSDL	99
142	9.4 eMSEP Web Service Description (WSDL)	99
143	9.4.1 eMSEP.wsdl	99
144	10 References	164
145	10.1 Normative	164
146	10.2 Non-Normative	164
147	Appendix A. Acknowledgments	166
148	Appendix B. e-MS Clinical Document Architecture Implementation Guide	167
149	Appendix C. Revision History	168
150	Appendix D. Notices	171

Table of Figures

153	Figure 1 System Overview – Pilot implementation of e-MSEP using Microsoft products.	10
154	Figure 2 UML Sequence Diagram of polling activity for receiving messages	13
155	Figure 3 UML Sequence Diagram for Sending a Message	14
156	Figure 4 UML Sequence Diagram illustrating reliable Sending of a Message	15
157	Figure 5 Fictitious Scenario	16
158	Figure 6 Message Wrapping	20
159	Figure 7 Logical Data Model	23
160	Figure 8 Example of a valid UUID URN	24
161	Figure 9 Lifecycle of a Sent Message in the e-MS Server	25

Table of Examples

164	Example 1 Received Request with required Message ID	29
-----	---	----

165	Example 2 Message Box Status Request	30
166	Example 3 Message Listing Request	30
167	Example 4 Message Listing Request with detail header information	30
168	Example 5 Message Listing Request with request to include received messages in the listing...	31
169	Example 6 MessageRead Request with required Message ID	31
170	Example 7 MessageRecv Request with required Message ID	32
171	Example 8 Reset Request.....	33
172	Example 9 Reset Request with optional Message ID	33
173	Example 10 Reset Request with optional date and time.....	33
174	Example 11 Send Request with only a ClinicalDocument	35
175	Example 12 Send Request containing a ClinicalDocument and Attachment.....	35
176	Example 13 Track Request with required Message ID parameter.....	37
177	Example 14 Received Response acknowledging the request	38
178	Example 15 Received Response indicating a No Such Message Error	38
179	Example 16 Received Response indicating a Message Already Received Error	38
180	Example 17 Status Response indicating 3 new messages are available for retrieval.	39
181	Example 18 Message Box Status Response indicating a Permission Error	39
182	Example 19 Message Box Status Response indicating a No Message Boxes Error	39
183	Example 20 Message Listing Response with Header Info included	40
184	Example 21 Message Listing Response – simple response	41
185	Example 22 Message Read Response acknowledging the request	41
186	Example 23 Message Read Response indicating a No Such Message Error	42
187	Example 24 Message Read Response indicating a Message Already Read Error	42
188	Example 25 Message Reset Response indicating 3 messages were reset based on a date and	
189	time requested.....	43
190	Example 26 Message Reset Response indicating 6 messages were reset.....	43
191	Example 27 Reset Response indicating that the specified message was reset	43
192	Example 28 Reset Response indicating a No Such Message Error	43
193	Example 29 Reset Response indicating No Data Found – reset didn't reset any messages	43
194	Example 30 MessageRecv Response returning a Read Confirmation Notification	45
195	Example 31 Message Receive Response returning a Clinical Document with Attachment	45
196	Example 32 Send Response with message ID returned.....	47
197	Example 33 Message Send Response returning an Error indicating No Such Address	47
198	Example 34 Track Response indicating the message not yet been received or read	48
199	Example 35 Track Response indicating the message has been received but not read	48
200	Example 36 Track Response indicating the message has been read but not received*	49
201	Example 37 Track Response indicating the message has been received and read	49
202	Example 38 Query Request for Organization details passing messaging address	56
203	Example 39 Query Request for Organization passing search parameters	56
204	Example 40 Query Request for Provider details passing the Provider's unique messaging	
205	address.....	57

206	Example 41 Query Request for Provider passing search parameters	57
207	Example 42 Query Request for Provider Listing for all records modified since a date/time	57
208	Example 43 Query Request for User details passing the Username (unique key)	58
209	Example 44 Query Request for User passing search parameters.....	59
210	Example 45 Query Response showing summarized Organization search results	60
211	Example 46 Query Response showing detail of a single Organization.....	60
212	Example 47 Query Response showing zero results returned	61
213	Example 48 Provider Query Response showing summarized Provider search results	62
214	Example 49 Query Response showing summarized User search results	63
215	Example 50 Query Response showing User Response Error – No Such Key Identifier	64
216	Example 51 Add user access.....	69
217	Example 52 Remove user access.....	69
218	Example 53 List user access.....	69
219	Example 54 Affiliate a user with an organization.....	69
220	Example 55 Remove a user from a role within an organization.....	69
221	Example 56 Remove a user from an organization	70
222	Example 57 List users affiliated with an organization.....	70
223	Example 58 List only Organization Administrator users affiliated with an organization	70
224	Example 59 List provider membership	70
225	Example 60 Add a provider to an organization	71
226	Example 61 Remove a provider from an organization (by address)	71
227	Example 62 Remove a provider from an organization (by provider ID)	71
228	Example 63 Update organization preferences	71
229	Example 64 View organization preferences	72
230	Example 65 Update organization details – modify city.....	72
231	Example 66 Add user access rights to Provider Membership.....	73
232	Example 67 Remove user access to Provider membership	73
233	Example 68 List user access to Provider membership	73
234	Example 69 Update Provider’s messaging address.....	74
235	Example 70 Update Provider’s note	74
236	Example 71 Update Provider’s preferences	74
237	Example 72 View Provider’s preferences.....	74
238	Example 73 Request User Access Listing	75
239	Example 74 User access added successfully	76
240	Example 75 User access removed successfully	77
241	Example 76 User access listing successfully	77
242	Example 77 User added successfully.....	77
243	Example 78 User listing.....	77
244	Example 79 Provider added successfully.....	78
245	Example 80 Provider listing	78
246	Example 81 View organization preferences	79

247	Example 82 Provider user access added	80
248	Example 83 Provider user access removed	80
249	Example 84 Provider user access listing	80
250	Example 85 Provider preferences view	81
251	Example 86 Provider address updated	81
252	Example 87 User Access Listing	82
253	Example 87 SOAP Request	97
254	Example 88 SOAP Response	98
255		
256		

1 Introduction

1.1 Purpose

The e-MS Exchange Protocol (e-MSEP) provides an open, non-proprietary digital message format for supporting the sharing of electronic medical summaries between care providers. This specification focuses on defining a communications-protocol neutral method for exchanging electronic medical summaries. It defines specific messaging communication constructs supporting reliable delivery of e-MS documents. The protocol supports delivery of messages in a store-and-forward manner, where a centralized messaging broker is periodically polled by receiving (EMR) systems to receive delivered messages.

1.2 Audience

The target audience for this specification is the community of electronic medical records (EMR) systems software developers who will implement the e-MS Exchange Protocol in order to integrate the capability to send and receive e-MS clinical documents.

1.3 Caveats and Assumptions

It is assumed the reader has an understanding of communications protocols, MIME, XML, XML Schema, SOAP, HL7 RIM and HL7 CDA Release 2 standards [HL7-CDA].

All examples are to be considered non-normative. If inconsistencies exist between the specification and the examples, the specification supersedes the examples.

The embedded W3C XML Schema (WXS) in this document is also to be considered non-normative. If inconsistencies exist between the referenced WXS documents and the embedded WXS copy, the referenced WXS documents supersede the content in this document.

1.4 Terminology

The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*, and *optional* in this document are to be interpreted as described in [RFC2119].

Electronic Medical Summary (e-MS):

An electronic medical summary (e-MS) is a subset of patient data suitable for communication among primary health care practitioners and other health care providers for the purpose of sharing the care of a patient.

Electronic Medical Record (EMR):

Term used to describe a physician office software system that is capable of storing electronic patient charts, replacing the need for paper-based patient records.

1.5 Notational Conventions

This specification uses the following namespace prefixes:

Prefix	Namespace
soap	http://schemas.xmlsoap.org/soap/envelope

Prefix	Namespace
SOAP-ENV	http://schemas.xmlsoap.org/soap/envelope
xs	http://www.w3.org/2001/XMLSchema/
hl7	urn:hl7-org:v3 (HL7 Version 3 CDA namespace)
b	http://schemas.microsoft.com/BizTalk/2003
emsep	http://www.e-ms.ca/Schema/v1

292

293

1.6 Relation to Other Specifications

294 **HL7 CDA Release 2** is the base document specification for the message content. This
295 specification defines the structure of the electronic medical summary document.

296 The e-MS project uses defined and normative CDA template specifications that describe the
297 structure, syntax and vocabulary used in an e-MS CDA template, such as a referral template.
298 Instances of the e-MS templates (XML documents) are, by design, 100% compatible with the
299 normative CDA release 2 (January 2005 member ballot). Validations of instances may be done
300 against the less restrictive normative CDA HL7 standards specification, or against the e-MS
301 templates using a combination of validation against the template WXS schema followed by a
302 verification process using Schematron with XPATH, or XPATH and XSLT. See the e-MS Clinical
303 Document Architecture Implementation Guide **[EMS-CDA]** for details.

304

1.7 Conformance

305 To conform to this specification, a system must implement all parts of this specification that are
306 relevant to the interfaces for which conformance is claimed. A system conforming to this
307 specification for the purposes of representing an e-MS Exchange Service, must implement this
308 complete specification and must implement an authorization model consistent with the
309 authorization model specified in this specification. The service system must implement an
310 authentication and security model that is consistent with the intent of this specification.

311

312

313

2 System Overview (non-normative)

2.1 What the e-MS Exchange Service Does

An Electronic Medical Summary (e-MS) is an electronic document that delivers a standard subset of patient data suitable for communication amongst Primary Health Care (PHC) practitioners and other practitioners for the purpose of sharing the care of an individual patient. The e-MS Exchange Service provides integration and interoperability between disparate physician office systems and primary care physicians to support the workflow process that benefit from the exchange of primary care patient medical summaries.

The e-MS Exchange Service's document structure leverages the HL7 CDA Release 2.

The users of the system will be physicians (including family physicians, consultants, and locums) and medical office assistants (MOA). However, an e-MS can only be addressed directly to a family physician, consultant, or organization. Other users will be given delegate privileges to view and act upon a primary physician's e-MS.

The e-MS Exchange Service is a clinical broker that handles the flow of all e-MS documents. The e-MS documents will be created and/or received either through the e-MS web application or an integrated EMR vendor application. It is anticipated that EMR vendors will integrate e-MS functionality directly into their existing systems. The e-MS Exchange Service supports message types to handle provider or organization lookups, e-MS message sending and receiving, administrative functions, delivery notifications and error handling.

In keeping with the BCMA position, the Exchange Service does not implement a central repository of data. Rather, it is a mechanism that supports the communication of relevant health information between practitioners. As this product will not be a repository, all reporting and auditing requirements focus solely on e-MS usage patterns, not content.

Different components of the e-MS system are depicted in the overview diagram below, including a Microsoft BizTalk 2004 server that manages the delivery of e-MS messages and documents between authorized e-MS users. The e-MSEP protocol is built from standards – XML, W3C Web Services and WS-I. As such, this protocol is not bound to Microsoft products. Microsoft BizTalk was selected by VIHA for the initial implementation.

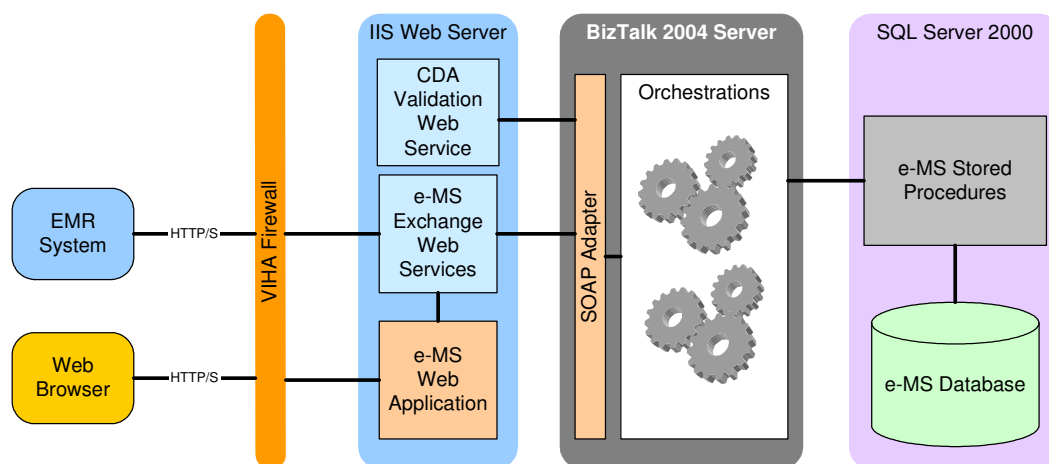


Figure 1 System Overview – Pilot implementation of e-MSEP using Microsoft products.

2.2 Use Cases

The following use cases illustrate the features that the e-MS exchange service exposed as XML schemas. The use cases and system requirements are further detailed in the referenced [EMS-SRS] document. The use cases described below capture the features supported by the e-MS Exchange Service.

Use Case Name	Description
Administration Services	
Audit Message Delivery	This use case describes how the actor audits the message delivery of the system.
Maintain Care Providers	This use case describes how the actor sets up and maintains care providers in the system. Care Providers are persons who are registered care providers and are recognized by the Province. Care Providers are registered with the Province's Provider Registry System (PRS). Care Providers become members of organizations within the system. They may be a member of more than one organization .
Maintain Organizations	This use case describes how the actor sets up and maintains organizations in the system. Organizations are identities such as Clinics or Hospitals that have Care Providers on staff. An Organization may be sent messages directly, or messages may be sent to the Care Providers who are members of the Organization.
Maintain Users	This use case describes how the actor sets up users of the system. Users are identities that actors use while interacting with the system. A User's username is unique across the entire system. Each User is associated with a real person or an EMR and with the role the person has within the organization . A User created for granting access to an EMR system would be configured to have the role of type 'EMR'.
Notify Document Removal	This use case describes when e-MS documents are about to reach a pre-defined expiry time and the system sends a notification message that the e-MS document is to be deleted from the Mailbox.
Remove Document	This use case describes when the system permanently deletes e-MS documents from storage in the e-MS system when the document expiration date is reached.
Configuration Services	
Configure Membership	An organization administrator adds a new Care Provider as a member of the organization. The Administrator adds the new member using functionality integrated into the EMR system. The actor EMR generates a command message and submits it to the system in order to fulfill the membership addition.
Configure Organization Settings	An organization administrator updates the organization preferences. The preferences relate to the organization's mailbox, and include: <ul style="list-style-type: none"> • Keep copy in sent items • Inbox retention settings for documents • Notification settings
Configure Organization Users	An organization Administrator adds and maintains users within the organization. A User is assigned a role or roles within the organization and may be granted access to the organization's mailbox.
Messaging Services	
Receive Messages	This use case describes how the actor EMR receives messages from the system using the system's service interface.
Reset Received Status	This use case describes how the actor resets the received status of a previously received message. Submitting this control message causes the messages or specific message (referenced in the control) to be reset. Resetting causes the message to be downloaded once when receiving messages from the system. This control message is used to recover previously downloaded messages.

Use Case Name	Description
Send Message	This use case describes how the actor EMR system sends either command messages or e-MS document messages to the system for delivery or for command processing.
Send Read Receipt	This use case describes how the actor EMR system submits read receipts to the system to indicate that a message was read by the recipient. The EMR sends a control message to the system indicating by message identifier which previously received message or messages have been read by the targeted recipient.
Track Message	This use case describes how a User checks the status of a previously sent message. The EMR system exposes the capabilities of the service interface, allowing a user to check whether a message has been delivered and read by the recipient. This message is an optional message. It is not required to be implemented by EMR vendors to pass conformance testing.
Query Services	
Lookup Care Provider	The use case describes how the actor EMR searches for care providers. When searching the system responds with a summary result set. When looking up, the system responds with a detailed response for the selected care provider or organization. This is used to expose the available care providers or organizations that may be sent e-MS documents.
Lookup Organization	The use case describes how the actor EMR searches for organizations, or looks up details on a specific organization. When searching the system responds with a summary result set. When looking up, the system responds with a detailed response for the selected organization. This is used to expose the available organizations that may be sent e-MS documents.
Lookup User	The use case describes how the actor EMR searches for registered users, or looks up details on a specific user. When searching the system responds with a summary result set. When looking up, the system responds with a detailed response for the selected user.

2.3 Basic Messaging Operation

Initially, the e-MS server host starts running the e-MS Exchange Service by listening on HTTPS ports to support Web Services as the transport for e-MS messages. When a client (an EMR system) wishes to make use of the service, it submits the request over an HTTPS connection. This involves the preamble of the HTTPS protocol to issue an authentication challenge and when successful sets up a Secure Socket transport layer before submitting the request.

Requests in the e-MSEP consist of composing the correct XML request message and wrapping it in a SOAP envelope and delivering it to the service URL.

Responses in the e-MSEP consist of a corresponding response XML message with either a success or failure indication. Failure indication implies the optional existence of a list of Errors contained in the message.

A client session progresses through a number of steps in order to retrieve and deliver messages. The client is expected to poll the exchange service periodically in order to determine whether messages are waiting to be retrieved. Sending messages may be done during the poll sessions or may be done at any time by initiating the send request.

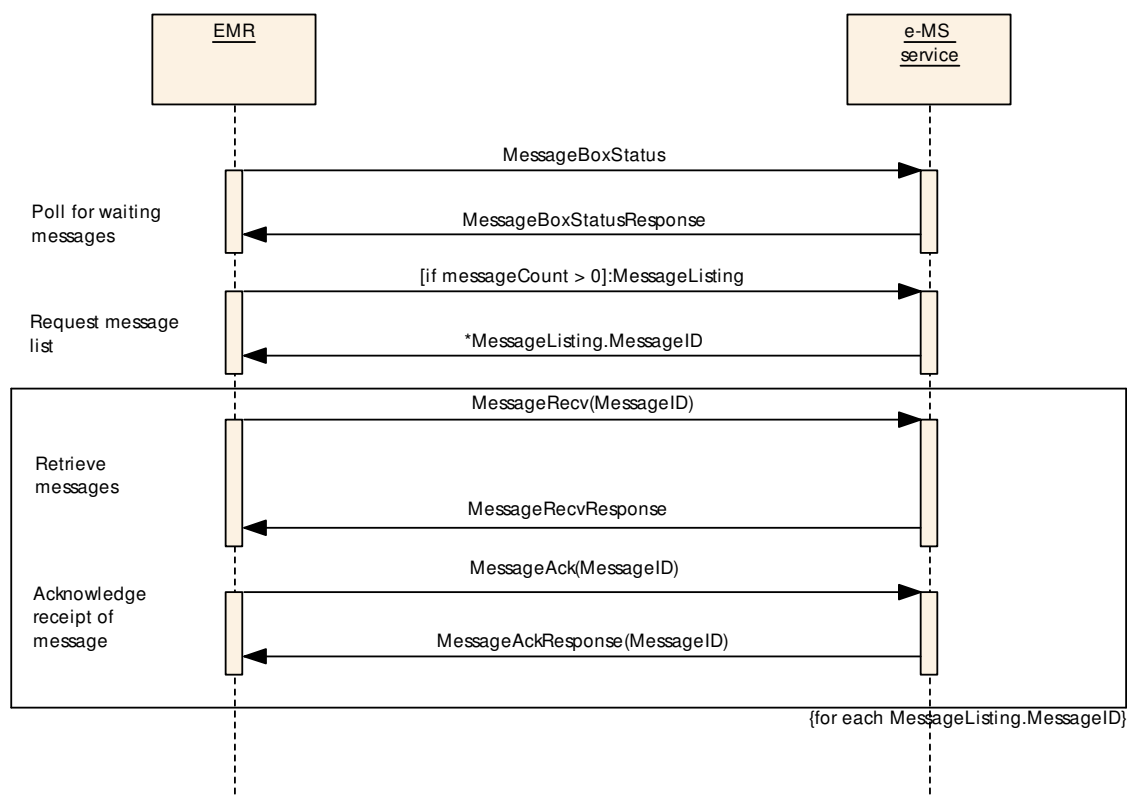
2.3.1 Retrieving Messages (Polling)

The typical sequence of requests by the client is:

Submit a **MessageBoxStatus** Request

If Status indicated messages waiting then

371 Submit a **MessageListing** Request
372 For each message in the list:
373 Submit a **MessageRecv** Request to receive a message
374 Submit a **MessageAck** Request acknowledging the retrieval.
375
376 The typical retrieval sequence is illustrated in Figure 2 below:
377



378
379 *Figure 2 UML Sequence Diagram of polling activity for receiving messages*

380 2.3.2 Sending Messages

381 To send a message:
382 1. Lookup recipient (provider or organization)
383 2. Create the Clinical Document
384 3. Create a **MessageSend** with a unique **MessageID**.
385 4. Submit a **MessageSend** Web Service Request (one per message to be send).
386 5. Wait for a **MessageSendResponse** for confirmation.

387 The sending system (source) creates a new **MessageID** for each new **SendMessage**. A
388 **MessageID** is a UUID or GUID, represented as a String. This provides for reliable delivery.
389 Should the transport fail, the source system simply resends the message, with its header and
390 contents preserved. Regardless of whether the e-MS service had previously receive the
391 message, it will receive it again, but will not duplicate the storage: it will simply update the entry
392 and return the confirming **MessageSendResponse** as it it had not already done so. This ensures

that the sending system ultimately receive the confirmation of delivery. It also ensures that the e-MS system does not store duplicate messages.

The typical message sending sequence is illustrated in Figure 3 below:

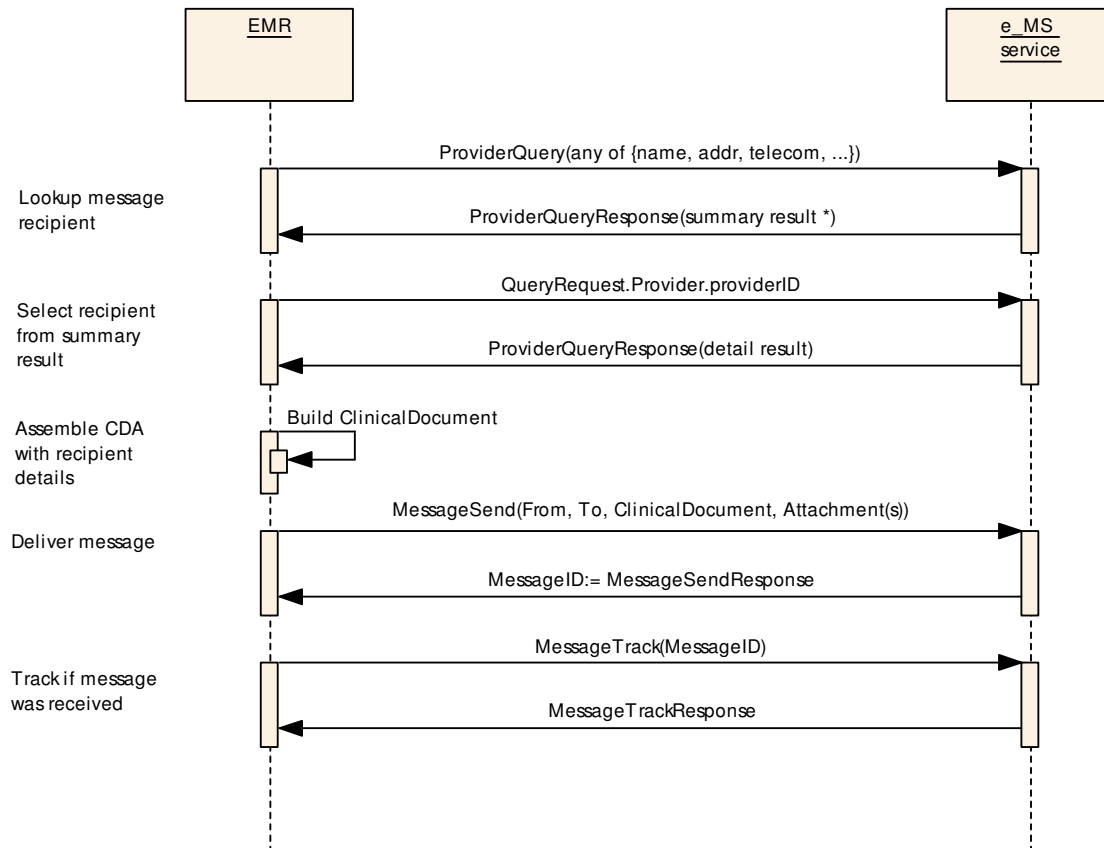


Figure 3 UML Sequence Diagram for Sending a Message

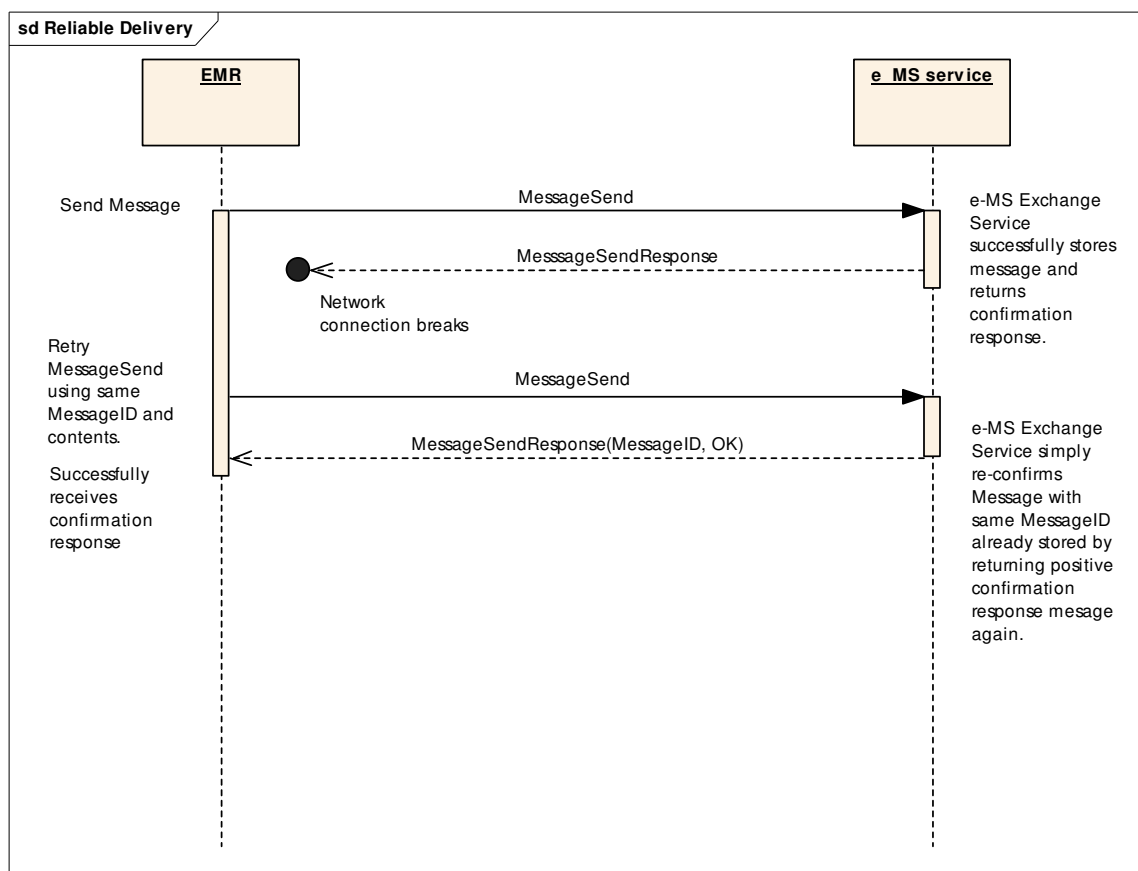


Figure 4 UML Sequence Diagram illustrating reliable Sending of a Message

2.4 Examples of Use Scenarios

The following fictitious examples illustrate a typical use of the e-MS Exchange Service. In the examples below, the following fictitious entities are used:

Dr. Susan Peters is a neurologist. She works at two clinics. The first clinic is her private practice that she shares with her associate Dr. Steven John. The clinic is known as the *PJ Neurology Clinic*. Ms. Amy Brooks is the medical office assistant at the PJ Neurology Clinic. They use a physician office software system to handle electronic medical records.

Dr. Peters lets Amy handle all incoming referrals and makes initial arrangements regarding the referrals.

Dr. Peters also works part time at a public health clinic. This clinic is known as South Central Public Health Clinic. This clinic also uses an EMR system for paperless records. Dr. Peters does not rely on an MOA at this clinic. She typically retrieves incoming referrals and deals with them directly.

Dr. Ed J. Williams, a general practitioner, often refers his patients to Dr. Peters. His practice is known as Williams Health Clinic.

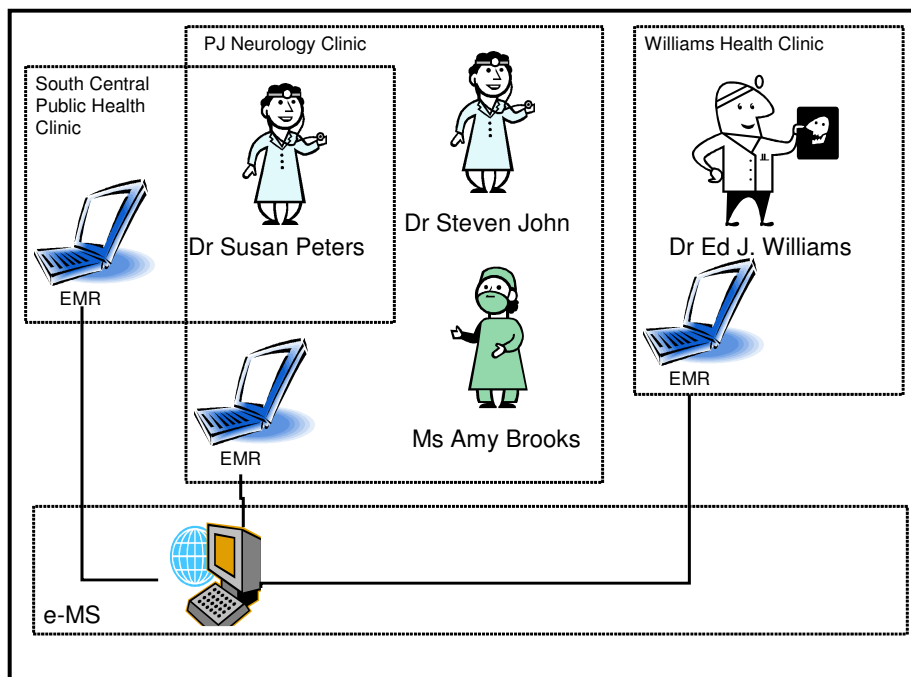


Figure 5 Fictitious Scenario

2.4.1 Sending a Message

Dr. Williams wants to send a referral letter to Dr. Peters. Using his EMR system that has been integrated into the e-MS Services, he creates a CDA document based referral letter using a form that has been built into his EMR.

He then looks up Dr. Peters. He sees two addresses for Dr. Peters – one at the PJ Neurology Clinic, the other at South Central. He chooses to send to Dr. Peters at the PJ Neurology Clinic.

He completes the form and saves it. The EMR system connects to the e-MS service and delivers the message. The message is now waiting in the message box of Dr. Peters at the PJ Neurology Clinic.

Dr. Peters' EMR system is configured to poll the e-MS system periodically to retrieve messages.

2.4.2 Receiving Messages

The PJ Neurology Clinic's EMR system polls the e-MS Exchange Service for waiting messages. The system returns a response indicating that one message is waiting. The EMR requests a message list. The system responds with one row containing a message identifier. The EMR requests to retrieve the message, passing the message identifier. The system responds by returning the referral document that Dr. Williams had sent to Dr. Peters. The EMR places this document into Dr. Peters work list. The EMR system sends an acknowledgement to the system to indicate that it successfully retrieved the message. It then shuts down until the next polling time.

2.4.3 Delegating Access

Dr. Susan Peters is planning a vacation, and wants to delegate access to her messages to Dr. Steven John. She uses her EMR system to lookup Steven's e-MS address and then adds Steven as a Delegate with a start date and end date specified that match her vacation time period. To

support this, the EMR system submits a Provider Query to lookup Steven's address, followed by an administrative request message to assign the delegate access rights for Steven.

2.4.4 Organization Administration

The PJ Neurology Clinic has just been setup to use the e-MS service. System accounts were created with usernames of `abrooks` for Amy Brooks, `sjohn` for Steven John, and `speters` for Susan Peters, as well as an account to use for the EMR system: `pjn_emr`. Amy was given `OrgAdministrator` role for PJ Neurology. Amy has been assigned the task of completing the setup. PJ Neurology was setup with a unique address of `urn:org:pjneurology.ca`, used when receiving messages to the organization, with no care provider targeted.

Amy logs into the e-MS system and assigns the following roles to the usernames:

Username	Roles
abrooks	MOA
sc_emr	EMR
sjohn	Provider
speters	Provider, OrgAdministrator

With that done, Amy then connects up the PJ Neurology EMR using the `pjn_emr` username for authenticating with the e-MS service.

Using the integrated features of the EMR, Amy then performs a search for Dr. Peters and Dr. John using the QueryRequest services of the e-MS service. She finds them on the system and adds them both to the PJ Neurology Organization, providing `urn:provider:pjneurology.ca:speters` and `urn:provider:pjneurology.ca:sjohn` as messaging addresses for them respectively.

Lastly, Amy grants the following access rights for users to be able to login and read and send messages:

Username	Access Rights	Provider/Organization (address)
abrooks	Delegate	urn:provider:pjneurology.ca:speters
abrooks	Owner	urn:org:pjneurology
speters	Owner	urn:provider:pjneurology.ca:speters
sjohn	Owner	urn:provider:pjneurology.ca:sjohn
speters	Delegate	urn:org:pjneurology
sjohn	Delegate	urn:org:pjneurology

3 e-MS Exchange Service Architecture

3.1 Overview

The e-MS Exchange Service architecture consists of an e-MS Exchange Service messaging (middleware) broker and e-MS clients (EMR systems). The e-MS Exchange Service broker provides the methods for managing the delivery and retrieval of e-MS documents. In addition it provides methods to manage the e-MS clients (care provider organization EMR systems) that are registered as clients of the e-MS Exchange Service. Care providers are managed through messages that support managing care provider membership to an organization and user association with an organization. User access is managed through messages that support assigning user association with an organization and granting rights of a user to access a care provider's message drop (mailbox) and preferences.

3.2 Design Considerations

The messaging protocol could not depend on the ability to push data directly to the target recipient (EMR system). Network connectivity of a physician's EMR system is not guaranteed to be available at all times. As such, the protocol needs to support a queue of messages for pickup by the EMR systems. An EMR system connects (periodically) to the e-MS Exchange Service to receive queued messages. This connectivity pattern is similar to the way that POP3 email messaging user agent, such as Microsoft Outlook Express polls an email server using POP3 protocol.

A polling mechanism is available in the HL7 V3 messaging, as well as in Web Service WS-I specifications. The EMR vendors were not interested in supporting HL7 V3 messaging and the use of WS-I bind the specification to Web Services. The e-MSEP protocol is designed to be independent of Web Services, though the initial implementation will make use of Web Services.

The e-MS messaging system must not act as a permanent repository of e-MS documents - a design constraint imposed by the e-MS project charter. The protocol does not directly expose this constraint, but contains error messages that would indicate a message is no longer available (due to retention expiring).

The goal of the protocol was to provide as simple and as minimal a set of commands to facilitate a polling-consumer. The popular and robust POP3 protocol was used as inspiration. The design was not constrained to follow any existing XML-based messaging standards.

3.3 Request/Response Messaging

All interfaces exposed by the service return a response message. There is a separate XML schema for response messages for each type of service interface.

Messages are sent synchronously. When a message is sent synchronously, the e-MS Exchange Service holds the communication mechanism open until the service returns a response. The response message contains:

- Status attribute (OK, ERROR)
- Error element when status set to ERROR.

Each request is atomic and either succeeds or fails in total. In the event of success, the e-MS Exchange Service returns the appropriate response related to the request type. The table shown below lists the request types and the corresponding response. When failure occurs, the response returns an ERROR value for the status attribute and returns one or more Error elements indicating the error(s) that occurred.

Purpose	Request Root Elements	Response Root Elements
Messaging	Message{Ack BoxStatus Listing Read Recv Reset Send Track}	Message{Ack BoxStatus Listing Read Recv Reset Send Track}Response
Query	{User Provider Org}Query	{User Provider Org}QueryResponse
Administration	{User Provider Org}Admin	{User Provider Org}AdminResponse

3.4 Command Message Design Pattern

Each of the e-MSEP Request messages implements a *command message*. A command message is simply a regular message that happens to describe commands and the command's arguments. For Web Services, each request is described as a `document` type SOAP Input request. The benefits of the command message pattern are:

The pattern allows a request to be invoked by a client without exposing mechanisms, such as objects and their types that implement the command.

It preserves a coarse-grained request/response command set – the entire command and its arguments are highly structured within the single command message. All that is needed is sent at once in a single request.

The command arguments and structure are not directly exposed as parameters. They are “hidden” within the command message. This permits the command arguments to be more complex and may change without affecting the message interface definition.

The command messages are portable – independent of the transport of the command message.

3.5 Web Services

Web Services is a platform independent mechanism for applications to expose services using standards-based interoperable protocols. Web Services are commonly defined to:

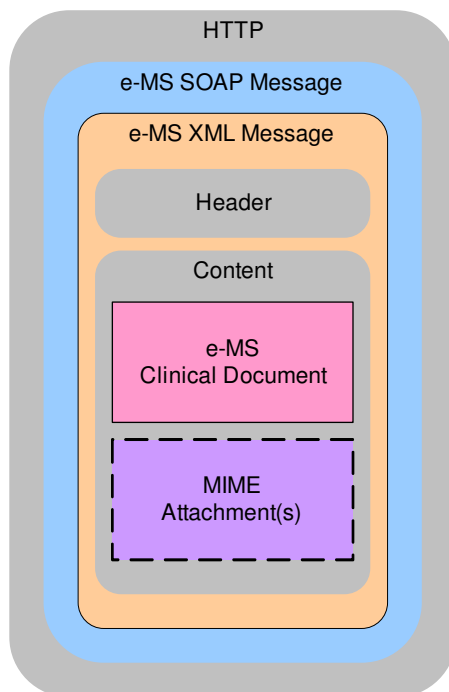
Expose useful service-based functionality using Web Services protocols.

Describe their interfaces in sufficient detail to allow a client application to communicate effectively. This description is provided using an XML document called a Web Services Description Language (WSDL) document.

Web Services may be registered in future so that potential client applications can find them. This is done using the Universal Discovery Description and Integration (UDDI) specification.

The e-MS Exchange Protocol is intended to be wrapped within a SOAP message over HTTP. However, it is not dependent on using SOAP. SOAP provides a standard way of exposing the services and includes security and authentication using HTTPS. The Send envelope contains a

542 “From” address and a “To” address which contain e-MS Exchange Service addresses that
543 represent the care provider the message is originating from and to be delivered to respectively.
544 The envelope layering is illustrated in Figure 6.



545
546 *Figure 6 Message Wrapping*

547 **3.5.1 Security Considerations**

548 **3.5.1.1 Authentication**

549 The HTTP (SOAP) interface supports authentication on a per message basis by verifying the
550 HTTP transport level using Basic or Digest authentication. HTTP allows for third-party
551 authentication mechanisms as well. A common authentication mechanism is Kerberos and
552 Microsoft Windows Integrated (NTLM). Either way, using HTTP over SSL (HTTPS) ensures that
553 the authentication credentials and messages transported are secure.

554
555 Only user (accounts) that have been authorized [see 3.5.1.2 Authorization] may be granted
556 access to the system. Authentication (username and password) is managed external to the e-MS
557 Exchange Service, but authorization is deployed within the system. As such, the system may be
558 deployed using any of the popular authentication mechanisms. The most basic is the use of
559 HTTP Basic authentication over HTTPS. The policies and requirements for authentication
560 methodologies are outside the scope of this document.

561 **3.5.1.2 Authorization**

562 The authenticated user is verified by the e-MS server to have the appropriate credentials to
563 execute the message being requested.

564 Authenticated Users that have been granted access to the e-MS Exchange Service will have
565 execution access limited by User Role and Member Access rights that have been granted.

566 For example, an EMR role user – a user account that is used by the EMR system will by default
567 have Owner Access rights granted by members of the organization. This allows the EMR system

to act as a fully privileged proxy to the actual user performing the action within the EMR system. Authorization to e-MS Exchange Services encapsulated and exposed from within the EMR system is the responsibility of the EMR system.

Service Request	EMR System	MOA	Provider	Organization Administrator
MessageAck	✱	N/A	N/A	N/A
MessageBoxStatus	✱	✱	✱	✱
MessageListing	✱	✱	✱	✱
MessageRead	✱	✱	✱	✱
MessageRecv	✱	✱	✱	✱
MessageReset	✱	N/A	N/A	N/A
MessageSend	✱	✱	✱	✱
MessageTrack	✱	✱	✱	✱
OrgQuery	◆	◆	◆	◆
OrgAdmin	◆	○	N/A	◆
ProviderQuery	◆	◆	◆	◆
ProviderAdmin	◆	○	○	◆
UserQuery	◆	◆	◆	◆
UserAdmin	◆	N/A	N/A	◆

✱ = Execute requires "Delegate" or "Owner" Access Rights.

○ = Execute requires "Owner" Access Rights

◆ = Execute Granted (Administration restricted to within Organization)

Users must have access to at least one message box in order to use the e-MS Exchange Service messaging area. By default EMR users are granted Owner access to each member's message box who is a member of the same organization. This may be disabled by removing the default access rights.

An organization must have at least one Organization Administrator. It is an error to attempt to delete the last remaining Organization Administrator.

A care provider (Organization Member) does need to have a user account, but must have at least one EMR user type granted at the very least Delegate access in order to facilitate message exchange.

3.5.1.3 Provider Message Box Access Rights

A user is granted either Owner access rights to a care provider's message box, or is granted Delegate access rights or no access (default). Delegate access may view messages and send

messages (on behalf of the provider), but may not alter message box preferences. Delegate access may be granted for a specified period of time, or may be granted indefinitely. Delegates can only be administered by a user account that has been granted Owner access to the member. A User may only be granted access if the user belongs to the same organization as the Member.

Owner Access Rights	A user account granted Owner Access to a member may read/write messages as well as modify member settings and preferences. An Owner Access user may also assign delegates.
Delegate Access Rights	A delegate may only read and write messages (on behalf of the member), but may not alter any member settings. A delegate may not delegate any other users (i.e. delegate chaining is not permitted)

3.5.2 Error Handling

Error handling is done either at the HTTP level when transport level errors occur, or at the Response message level, where a status attribute is set to ERROR and one or more Error elements are returned to indicate the errors that have occurred.

When transport errors occur, the HTTP status code is returned. A detailed list of HTTP status codes can be found in [RFC2616].

3.5.3 Logical Data Model

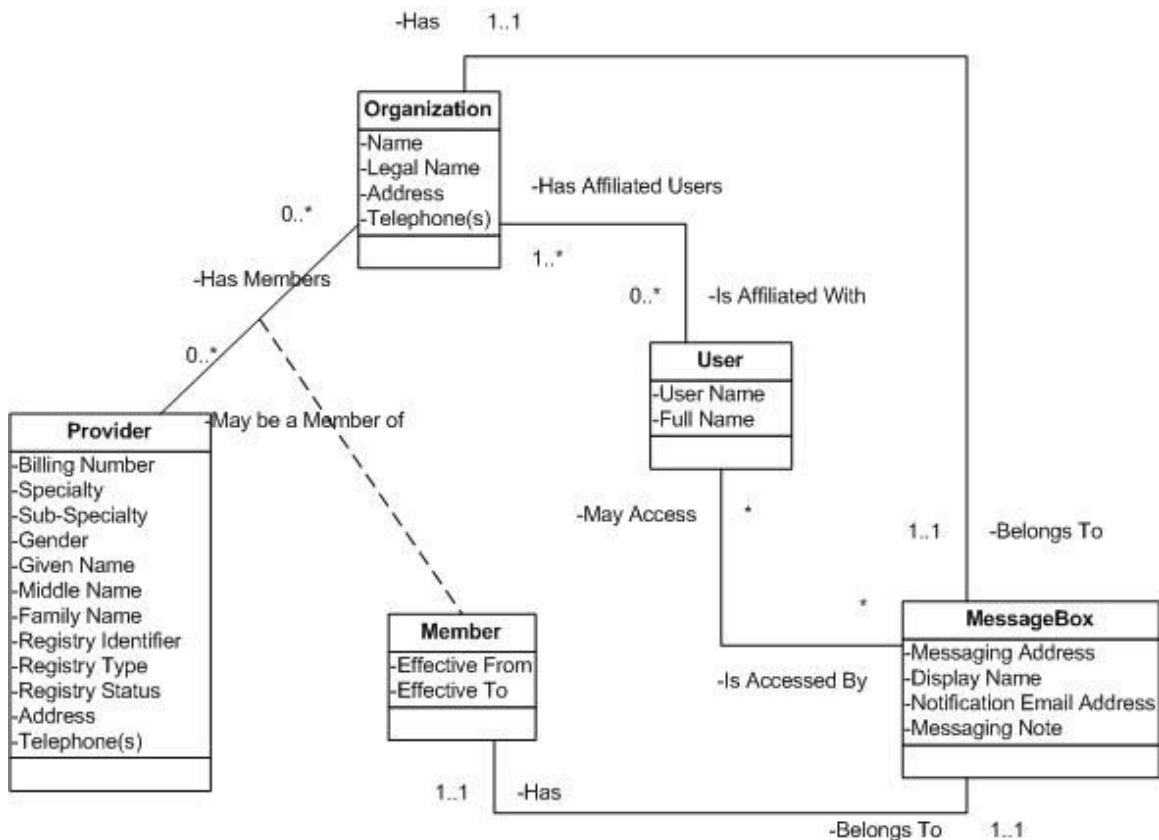


Figure 7 Logical Data Model

3.6 Mutable and Immutable Key Identifiers

Within the domain model, entities are identified by either mutable or immutable key identifiers. This section describes the key identifiers for the model entities.

3.6.1 Provider and Organization Addresses

Messages are sent to unique addresses. The address may be for an Organization or for a Provider at an organization. A provider has a unique address for each organization membership. The messaging addresses are unique within the service, but are mutable. The address of an organization or a provider may be modified provided it remains unique. Provider addresses may contain a base part that specifies the organization they belong to. This is not enforced. The only restriction is that the provider's address be unique and be prefixed with `urn:provider:.` However, in practice it is recommended that the provider's address reflect the organization they belong to – particularly if the provider is a member of more than one organization. An organization's address must start with: `urn:org:`

Addresses are restricted to the URN Syntax. The URN syntax is defined in [RFC2141].

3.6.1.1 Examples

For example, Dr. Plumber is a member of the Home Health Care Clinic and the Health Authority West Clinic:

Possible address for Home Health Care Clinic :

```
urn:org:home.healthcare.clinic
```

Possible address for Dr. Peter Plumber:

```
urn:provider:home.healthcare.clinic:pplumber
```

Possible address for Health Authority West:

```
urn:org:health-authority-west
```

Possible address for Dr. Peter Plumber at the Health Authority West:

```
urn:provider:ha-west:peter_plumber
```

If the Home Health Care Clinic changed its address, the change would not automatically be reflected in its members' addresses. This is because the only restriction is that the addresses are unique. The addresses are voluntarily descriptive enough to indicate which clinic the provider is at:

New address for Home Health Care Clinic:

```
urn:org:new-home.healthcare.clinic.ca
```

The address for Dr. Peter Plumber would remain as:

```
urn:provider:home.healthcare.clinic:pplumber
```

3.6.2 Unique Identifiers (Immutable)

Universally unique identifiers (UUID) are used to define immutable key identifiers for:

- Providers
- Messages

Provider and message (immutable) identifiers are defined using URN syntax, as defined in [RFC2141].

```
urn:uuid:0FB220FF-9D91-43FB-8D77-E9B3A5EFCEDA
```

Figure 8 Example of a valid UUID URN

3.6.3 User Names

User names are unique within the service system. The username is a login account on the system and is maintained by the e-MS service system administrators. User names may be updated, if need be, with no negative effect – all user access rights and roles are preserved.

Passwords are restricted by the policies established within the operating environment that the e-MS service is deployed to. The password length, complexity and expiration must comply with the guidelines defined by the organization responsible for hosting the e-MS Exchange Service, and must comply with the [EMS-PIA]

3.7 Lifecycle of an e-MS Message

The lifecycle of a sent message is illustrated in Figure 9. The message is not deleted from the e-MS Exchange Service until the message delivery is complete or until after a timeout occurs while waiting for the message to be read by the recipient. The lifecycle permits the receiving client to reset the received state of a previously received message. This makes that message available for retrieval once again. This is supported by the `MessageReset` and the `MessageRecv` requests. The diagram shows the requests that drive the message into the next state. The cleanup event is a process within the server that cleans up messages on a periodic basis. This prevents messages from being permanently stored. Messages are retained only for a period sufficient to support reliable delivery using a store-and-forward design.

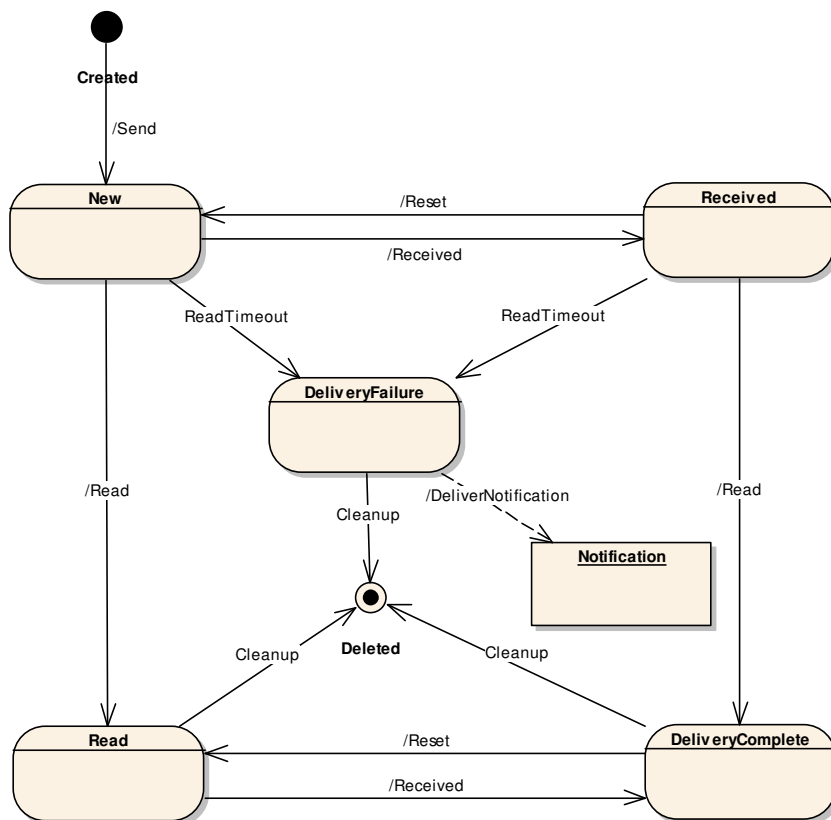


Figure 9 Lifecycle of a Sent Message in the e-MS Server

3.7.1 New Status

A message is considered new until the intended recipient has either downloaded the message or has read it. The message may be reset back to *New* status once received. Only messages not received or marked as *Delivery Complete* are listed for retrieval.

3.7.2 Read Status

When a message is read, a notification is optionally sent to the recipient to indicate this. The business transaction of delivering the message to the recipient is considered successful when the message is read within a reasonable time period. Messages not read on time cause a notification to be delivered to the sender informing of the failed delivery. A sending EMR system may inquire about the status of a sent message using the `MessageTrack (messageID)` request.

A message may only be marked as read by the intended recipient EMR system. The recipient's EMR sends a `MessageRead (messageID)` request to the e-MS Exchange Service to indicate that the message has now been read. Once a message is marked as read, it can't be unmarked as read.

A person (the recipient) marks a message as having been read. This is an explicit action by the user. The user must ensure that whatever message persistency is needed has been established prior to marking the message as having been read, since read messages are eligible for deletion (in order to preserve the non-repository mandate).

3.7.3 Received Status

An EMR client is responsible for marking a message as received. Once marked as received, a `MessageListing` Request does not return the message ID. Sending a `MessageAck` is intended as a positive acknowledgement by the EMR that it has successfully received the message. The EMR system may choose to acknowledge each message immediately after receiving it, or it may send a `MessageAck` message after receiving all the new messages. The protocol is designed to not force a sequence of `MessageAck (messageID)`, followed by `MessageRecv (messageID)`. This allows the client to determine when it is okay to mark a message as received. Once marked as `Received`, the message won't be listed as a result of the `MessageListing` request unless the optional `received` attribute is set to `true`.

A message can be reset to be no longer considered acknowledged as received by sending a `MessageReset` request.

An EMR system may issue a `MessageReset` when it needs to receive the message(s) again. This allows for recovery of lost messages.

3.7.4 Delivery Complete Status

The delivery of a message is completed when the client system has downloaded a message and the intended recipient has read it. The client system submits a `MessageRead (messageID)` to indicate the message has been read. Alternatively, it is possible for a recipient to read a new message that has not been received when a user of the web application opens the message. If an e-MS document is read from the web application, then the delivery of the message is considered complete. Once the delivery of the message is complete, the broker will keep the document on the server until the cleanup process determines that it is time to delete the message.

The sequence for the delivery of a message to be complete is `MessageRecv (messageID)`, followed by `MessageAck (messageID)` and finally `MessageRead (messageID)` is sent once a person using the EMR system has explicitly indicated that the message has been read and safely stored.

3.7.5 Delivery Failure Status

When a message in `New` or `Received` status has not been read for a period of time, the message delivery is considered to have failed. When the time comes to do message cleanup, the failed messages are removed. A warning notification is sent to the recipient via their email address when a message is about to be deleted and when it has not yet been marked as read.

3.7.6 Delivery Failure Notification

For every message with a Delivery Failure status, the e-MS service originates a Delivery Failure Notification message for the sender of the failed message, indicating the message delivery failure. The sender specifies the number of days that the recipient is expected to have read the message. When this time is exceeded, the notification is sent to the sender.

3.8 Notification Messages

Notification Messages are messages generated and delivered by the e-MS service. General Notifications may be sent consisting of textual notes, or it may be related to a Message delivery. `MessageEvents` are the most typical usage of a Notification message. They are used to inform of delivery status and alert of unread message deletion. The types of `MessageEvent` Notifications are listed in the table below.

Message Event Name	Description
NotRead-Purged	This notification is sent to the intended recipient via email. If no email exists for the recipient, it is sent to the organization's message box. This is only sent when a message has not been read. Messages are purged based upon a system-level maximum retention period, or by the user defined retention period.
NotRead-WillPurge	A message has not been read and will soon be deleted. This event may be sent to the message recipient after a system-configurable time, in days, before the retention time period. If the message has not been marked as read, this notification is sent to the notification email address of the recipient.
NotRead-WithinExpectedDays	This notification is sent to the sender when a sent message has not been read within the sender's expected number of days. The number of days is set by the sender.
ReadConfirmation	Message has been read. This notification is (optionally) sent to the sender when the recipient marks a message as read. The sender controls whether they want this notification with the ReadReceipt preference.

4 Messaging Services Interface

4.1 Overview

The exchange messaging services expose the messages necessary to send and receive messages to and from the e-MS broker.

4.2 Messaging Request

The messaging request consists of one of the following root elements:

- MessageAck
- MessageBoxStatus
- MessageListing
- MessageRead
- MessageRecv
- MessageReset
- MessageSend
- MessageTrack

4.2.1 MessageAck Request

The `MessageAck` request is used to mark the message in the message drop on the server as having been received by the client. This keeps the message from being returned in subsequent `MessageListing` requests that occur during the polling from an EMR system. Once a message is marked as received, attempting to retrieve does not cause an error. This is to support the ability to submit a `MessageListing` request to include received messages – for the purpose of listing all messages that are in the message drop on the server.

4.2.1.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageAck" type="MessageRequestBase"/>
</xs:schema>

<xs:complexType name="MessageRequestBase">
  <xs:sequence>
    <xs:element name="MessageID" type="MessageID"/>
  </xs:sequence>
</xs:complexType>
```

4.2.1.2 Returns

This request returns a Received Response. See section 4.3.1 for details.

4.2.1.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchMessage (100)
- MessageAlreadyAcked (102)

4.2.1.4 Example

Example 1 Received Request with required Message ID

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageAck xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
</MessageAck>
```

4.2.2 MessageBoxStatus Request

The `MessageBoxStatus` request is intended to be sent by the EMR system to check whether there are new messages waiting for retrieval. The command takes no arguments. The system determines the organization to query based upon the authenticated EMR system user credentials. The `MessageBoxStatus` request takes no arguments and normally returns the count of messages waiting in the mail drop, or returns an exception response when an error occurred.

4.2.2.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageBoxStatus" type="MessageBoxStatus"/>

  <xs:complexType name="MessageBoxStatus">
    <xs:sequence><xs:any minOccurs="0"
      maxOccurs="0"/></xs:sequence>
  </xs:complexType>
</xs:schema>
```

4.2.2.2 Returns

The return type for this request is a `MessageBoxStatusResponse`. See section 4.3.2 for details.

4.2.2.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- `PermissionError(200)`
- `NoMessageBoxes(103)`

4.2.2.4 Example

Example 2 Message Box Status Request

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageBoxStatus xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
</MessageBoxStatus>
```

4.2.3 MessageListing Request

The `MessageListing` request passed with no arguments returns the lists of new messages by message ID. If the request is sent with the optional `received` attribute set to `true`, then messages that have been marked as Received are included in the listing. When the optional attribute `showHeaderInfo` is set to `true`, the response includes message header information. See 0 `MessageListingResponse` for header information detail.

4.2.3.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
targetNamespace="http://www.e-ms.ca/Schema/v1"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageListing" type="MessageListing"/>

  <xs:complexType name="MessageListing">
    <xs:attribute name="showHeaderInfo" type="xs:boolean" use="optional"
default="false"/>
    <xs:attribute name="received" type="xs:boolean" use="optional"
default="false"/>
  </xs:complexType>
</xs:schema>
```

4.2.3.2 Returns

This request returns a `MessageListingResponse`. See section 0 for details.

4.2.3.3 Errors

- There are no additional exceptions beyond those common to all requests.

4.2.3.4 Examples

Example 3 Message Listing Request

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageListing xmlns="http://www.e-ms.ca/Schema/v1"
:xsi="http://www.w3.org/2001/XMLSchema-instance"
:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
</MessageListing>
```

Example 4 Message Listing Request with detail header information

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageListing xmlns="http://www.e-ms.ca/Schema/v1"
:xsi="http://www.w3.org/2001/XMLSchema-instance"
:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd"="true">
</MessageListing>
```

Example 5 Message Listing Request with request to include received messages in the listing

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageListing xmlns="http://www.e-ms.ca/Schema/v1"
:xsi="http://www.w3.org/2001/XMLSchema-instance"
:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd"="true">
</MessageListing>
```

4.2.4 MessageRead Request

The MessageRead request is issued by an EMR when a person has read the message previously received. The e-MS Exchange Service tracks that status of the delivered message and informs the sender when messages are read (if subscribed to that notification). This command takes a message ID as an argument.

4.2.4.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
targetNamespace="http://www.e-ms.ca/Schema/v1"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:include schemaLocation="admin/admin.xsd"/>
<xs:include schemaLocation="query/query.xsd"/>
<xs:include schemaLocation="messaging/messaging.xsd"/>
<xs:include schemaLocation="MessagingDataTypes.xsd"/>

<xs:element name="MessageRead" type="MessageRequestBase"/>
</xs:schema>
```

4.2.4.2 Returns

This request returns a Read Response. See section 0 for details.

4.2.4.3 Errors

In addition to the exceptions common to all requests, the following errors may be returned:

- NoSuchMessage (100)
- MessageAlreadyRead (101)

4.2.4.4 Example

Example 6 MessageRead Request with required Message ID

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageRead xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
<MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
</MessageRead>
```

4.2.5 MessageRecv Request

The `MessageRecv` Request takes a single argument – a message ID. It returns the message referenced by the ID, or an error is returned if the message is not found. If the message ID refers to a message previously received (via a `MessageRecv` request), this is an error.

4.2.5.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageRecv" type="MessageRequestBase"/>
</xs:schema>
```

4.2.5.2 Returns

This request returns a `MessageRecvResponse`. See section 0 for details.

4.2.5.3 Errors

In addition to the exceptions common to all requests, the following error may be returned:

- `NoSuchMessage` (100)

4.2.5.4 Example

Example 7 MessageRecv Request with required Message ID

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageRecv xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
</MessageRecv>
```

4.2.6 MessageReset Request

The purpose of the `MessageReset` Request is to allow the client EMR system to recover from message losses due to some form of EMR failure.

The `MessageReset` request with no arguments resets all messages in the server's message drop as no longer marked as received. This allows the POS to reissue a `MessageListing` request for the messages and they will be listed once more. The `MessageReset` request may also take either a timestamp or a message ID. A timestamp as an argument resets all messages stored since the timestamp provided. When the argument contains a message ID, only the message with the referred message ID will be reset.

4.2.6.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1">
```



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

<xs:include schemaLocation="admin/admin.xsd"/>
<xs:include schemaLocation="query/query.xsd"/>
<xs:include schemaLocation="messaging/messaging.xsd"/>
<xs:element name="MessageReset" type="MessageReset"/>

<xs:complexType name="MessageReset">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element name="DateTime" type="xs:dateTime"/>
      <xs:element name="MessageID" type="MessageID"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

4.2.6.2 Returns

The return type for this request is a `MessageResetResponse`. See section 0 for details.

4.2.6.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- `NoSuchMessage` (100) indicates that the message ID provided in the request is not valid.
- `NoDataFound`(302) indicates that the reset request did not reset any messages.

4.2.6.4 Examples

Example 8 Reset Request

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageReset xmlns="http://www.e-ms.ca/Schema/v1"
: xsi="http://www.w3.org/2001/XMLSchema-instance"
: schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
</MessageReset>
```

Example 9 Reset Request with optional Message ID

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageReset xmlns="http://www.e-ms.ca/Schema/v1"
: xsi="http://www.w3.org/2001/XMLSchema-instance"
: schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
</MessageReset>
```

Example 10 Reset Request with optional date and time

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageReset xmlns="http://www.e-ms.ca/Schema/v1"
: xsi="http://www.w3.org/2001/XMLSchema-instance"
: schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <DateTime>2004-05-26T21:32:52.12679</DateTime>
</MessageReset>
```

4.2.7 MessageSend Request

The `MessageSend` request is used to send an e-MS clinical document from one care provider to another. It carries a `Message` which consists of three required arguments: `From`, `To` and the `Content` and two optional arguments `Subject` and `Priority`. The `Message` element is designed to carry **any** `<xs:any/>` element type. However, the e-MS Exchange Service expects

to receive a child element in the Body of `<ClinicalDocument xmlns="urn:hl7-org:v3"/>`, which is an HL7 CDA Release 2 compliant clinical document element. The e-MS Exchange Service will validate the `ClinicalDocument` against a set of e-MS clinical document templates that have been created following the HL7 CDA Release 2 templating rules.

By keeping this message schema open through the use of `<xs:any/>`, templates and other document types may be supported in the future without requiring changes to this messaging specification.

The `Message` element is a complex type that, in addition to a single `ClinicalDocument`, may contain zero or more elements of type `Attachment`. Attachments are intended to contain MIME Version 1.0 plain text or base-64 encoded multimedia. The `Attachment` element should contain a complete MIME formatted document. Attachments transmitted are intended to be associated (by URI reference) to sections within the clinical document.

4.2.7.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageSend" type="MessageSend.Any"/>

  <xs:complexType name="MessageSend.Any">
    <xs:sequence>
      <xs:element name="Message" type="MessageTx.Any"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="MessageTx.Any">
    <xs:sequence>
      <xs:group ref="MessageHeaderGrp"/>
      <xs:group ref="MessagePayload.Any"/>
    </xs:sequence>
  </xs:complexType>

  <xs:group name="MessageHeaderGrp">
    <xs:sequence>
      <xs:element name="From" type="MessagingAddress"/>
      <xs:element name="To" type="MessagingAddress"/>
      <xs:element name="Subject" type="xs:string" minOccurs="0"/>
      <xs:element name="Priority" type="PriorityType" minOccurs="0"/>
      <xs:element name="MessageID" type="MessageID"/>
    </xs:sequence>
  </xs:group>

  <xs:group name="MessagePayload.Any">
    <xs:sequence>
      <xs:any namespace="##other" processContents="skip"/>
      <xs:element name="Attachment" type="Attachment" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

</xs:schema>
```

4.2.7.2 Returns

The return type for this request is a Send Response. See section 0 for details.

4.2.7.3 Errors

In addition to the errors common to all requests, the following errors may be returned:

- NoSuchAddress(100)
- UnsupportedMessage(103)
- ValidationError(400)
- NotWellFormedError(401)

4.2.7.4 Examples

Example 11 Send Request with only a Clinical Document

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageSend xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:hl7="urn:hl7-
org:v3" :schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Message>
    <From displayName="Dr. Patricia Primary"
value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
value="urn:provider:urology.ca:pplumber"/>
    <Subject>Referral Letter from Level 7 Healthcare, Inc.</Subject>
    <Priority>Highest</Priority>
    <MessageID>urn:uuid:69E9CA58-305E-434e-B4BA-6587E91C80C8</MessageID>

    <Content>
      <ClinicalDocument xmlns="urn:hl7-org:v3">

        ...

      </ClinicalDocument>
    </Content>
  </Message>
</MessageSend>
```

Example 12 Send Request containing a ClinicalDocument and Attachment

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageSend xmlns="http://www.e-ms.ca/Schema/v1"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP_CDA.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Message>
    <From displayName="Dr. Patricia Primary"
value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
value="urn:provider:urology.ca:pplumber"/>
    <Subject>Referral Letter from Level 7 Healthcare, Inc.</Subject>
    <Priority>Highest</Priority>
    <MessageID>urn:uuid:6823AC2F-B374-4214-AF1B-407091BBED37</MessageID>
    <Content>
      <ClinicalDocument
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:hl7-org:v3 EMS_CDA.xsd"
        xmlns="urn:hl7-org:v3">
        <id/>
        <code/>
        <effectiveTime/>
        <author>
          <time/>
          <assignedAuthor>
            <id/>
          </assignedAuthor>
        </ClinicalDocument>
      </Content>
    </Message>
  </MessageSend>
```

```

1154      </author>
1155      <custodian>
1156        <assignedCustodian/>
1157      </custodian>
1158      <recordTarget>
1159        <patientRole>
1160          <id/>
1161        </patientRole>
1162      </recordTarget>
1163      <component>
1164        <bodyChoice>
1165          <StructuredBody>
1166            <component>
1167              <section>
1168                <entry>
1169                  <entryChoice>
1170                    <ObservationMedia>
1171                      <value mediaType="image/jpeg"
1172                        encoding="B64">
1173                        <reference value="test.jpg"/>
1174                      </value>
1175                    </ObservationMedia>
1176                  </entryChoice>
1177                </entry>
1178              </section>
1179            </component>
1180          </StructuredBody>
1181        </bodyChoice>
1182      </component>
1183    </ClinicalDocument>
1184    <Attachment fileName="test.jpg">
1185      MIME-Version: 1.0 Content-Type: image/jpeg;
1186      name="test.jpg" Content-Transfer-Encoding: base64
1187      Content-Description: test.jpg
1188      Content-Disposition: attachment; filename="test.jpg"
1189      Content-Location: test.jpeg
1190
1191      naXZlIG9ldCB5b3VyIHBhc3N3b3JkIG9yIGNyZWRpdCBjYXJkIG5lbWJlciBpbiBh
1192
1193      biBpbmN0YW50IGl1c3NhZ2UgY29udmVyc2F0aW9uLg0KICANC1Zlcm9uaWNhIFJ5YW4gKFZvbHQ
1194      DQpoaSBhZGUNC1ZlcYSBSeWFuIChWb2x0KSBzYXlzOg0KaSB0eYXZlIGEgcHJl
1195
1196      bWllciBjYXWxYmFjaw0KQWRlIEZhbW90aSBzYXlzOg0KSSJtIG9uIHRoZSBwaG9uZSB3aXR0IGE
1197    </Attachment>
1198  </Content></Message>
1199 </MessageSend>

```

4.2.8 MessageTrack Request

The MessageTrack request takes a single argument message ID. It returns the status of the message – whether it has been retrieved (by the recipient's EMR System) and whether it has been read by the recipient.

4.2.8.1 Definition

```

1205 <?xml version="1.0" encoding="UTF-8"?>
1206 <xs:schema
1207   xmlns="http://www.e-ms.ca/Schema/v1"
1208   xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
1209   xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
1210   targetNamespace="http://www.e-ms.ca/Schema/v1"
1211   xmlns:xs="http://www.w3.org/2001/XMLSchema">
1212
1213   <xs:include schemaLocation="admin/admin.xsd"/>
1214   <xs:include schemaLocation="query/query.xsd"/>
1215   <xs:include schemaLocation="messaging/messaging.xsd"/>
1216   <xs:element name="MessageTrack" type="MessageRequestBase"/>
1217 </xs:schema>

```

4.2.8.2 Returns

The return type for this request is a `MessageTrackResponse`. See section 0 for details.

4.2.8.3 Errors

In addition to the exceptions common to all requests, the following error may be returned:

- `NoSuchMessage(100)`

4.2.8.4 Example

Example 13 Track Request with required Message ID parameter

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageTrack xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
</MessageTrack>
```

4.3 Messaging Responses

The messaging responses consist of one of the following root elements:

- `MessageAckReponse`
- `MessageBoxStatusResponse`
- `MessageListingResponse`
- `MessageReadResponse`
- `MessageRecvResponse`
- `MessageResetResponse`
- `MessageSendResponse`
- `MessageTrackResponse`

4.3.1 MessageAckResponse

The `MessageAckResponse` is used to return from a received request. If an error occurred, it returns an `Error` element within the response.

4.3.1.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>
  <xs:element name="MessageAckResponse" type="MessageRefResponse"/>
</xs:schema>
```

4.3.1.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchMessage(100)
- MessageAlreadyAcked(102)

4.3.1.3 Examples

Example 14 Received Response acknowledging the request

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageAckResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd"
  status="OK">
  <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
</MessageAckResponse>
```

Example 15 Received Response indicating a No Such Message Error

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageAckResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd"
  status="ERROR">
  <Error errorCode="100">
    <ErrorMsg>No such message</ErrorMsg>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
  </Error>
</MessageAckResponse>
```

Example 16 Received Response indicating a Message Already Received Error

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageAckResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd"
  status="ERROR">
  <Error errorCode="102">
    <ErrorMsg>Message already acknowledged</ErrorMsg>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
  </Error>
</MessageAckResponse>
```

4.3.2 MessageBoxStatusResponse

The MessageBoxStatusResponse is used to return from a MessageBoxStatus request. If an error occurred, it returns an Error element within the response and the status attribute is set to ERROR.

4.3.2.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>
  <xs:element name="MessageBoxStatusResponse"
    type="MessageBoxStatusResponse"/>

  <xs:complexType name="MessageBoxStatusResponse">
    <xs:complexContent>
      <xs:extension base="ResponseBase">
        <xs:choice>
```

```
1317         <xs:sequence>
1318             <xs:element name="MessageCount" type="xs:long"/>
1319         </xs:sequence>
1320     <xs:sequence>
1321         <xs:element name="Error" type="Error"
1322             maxOccurs="unbounded"/>
1323     </xs:sequence>
1324 </xs:choice>
1325 </xs:extension>
1326 </xs:complexContent>
1327 </xs:complexType>
1328 </xs:schema>
```

4.3.2.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- PermissionError(200)
- NoMessageBoxes(103)

4.3.2.3 Examples

Example 17 Status Response indicating 3 new messages are available for retrieval.

```
1335 <?xml version="1.0" encoding="UTF-8"?>
1336 <MessageBoxStatusResponse xmlns="http://www.e-ms.ca/Schema/v1"
1337     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1338     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
1339     <MessageCount>3</MessageCount>
1340 </MessageBoxStatusResponse>
```

Example 18 Message Box Status Response indicating a Permission Error

```
1342 <?xml version="1.0" encoding="UTF-8"?>
1343 <MessageBoxStatusResponse xmlns="http://www.e-ms.ca/Schema/v1"
1344     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1345     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
1346     <Error errorCode="200">
1347         <ErrorMsg>Permission Error for username: 'tcollins'</ErrorMsg>
1348     </Error>
1349 </MessageBoxStatusResponse>
```

Example 19 Message Box Status Response indicating a No Message Boxes Error

```
1351 <?xml version="1.0" encoding="UTF-8"?>
1352 <MessageBoxStatusResponse xmlns="http://www.e-ms.ca/Schema/v1"
1353     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1354     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
1355     <Error errorCode="102">
1356         <ErrorMsg>No Message Boxes: User has not been given access to any message
1357         boxes</ErrorMsg>
1358     </Error>
1359 </MessageBoxStatusResponse>
```

4.3.3 MessageListingResponse

The MessageListingResponse is used to return a response to the MessageListing request. If an error occurred, it returns an Error element within the response.

4.3.3.1 MessageInfo Element

When showHeaderInfo=true as specified in the MessageListing request, the response returned zero or more MessageInfo elements. In this is the MessageType element which specifies the type of message.

Valid `MessageType` types are `ClinicalDocument`, `Notification` and `Other`. Each of these types may have a value, which defines its subtype. For example, the `ClinicalDocument` type may have a value of "Referral" or "ConsultationNote". The value is not constrained to have a scoped range of values, but is designed to be extended as the protocol supports more kinds of Clinical Documents. The `Notification` type has values which describe the type of notification being returned.

The `MessageType` element is useful when a `MessageListingResponse` is used to provide a "master" view of the message box contents. By knowing the message type, the messages of higher business importance could be selectively received before the less important messages. Knowing the type helps to make this decision.

4.3.3.2 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageListingResponse" type="MessageListingResponse"/>
  <xs:complexType name="MessageListingResponse">
    <xs:complexContent>
      <xs:extension base="ResponseBase">
        <xs:choice>
          <xs:sequence>
            <xs:element name="MessageInfo" type="MessageInfo"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:sequence>
            <xs:element name="Error" type="Error"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

4.3.3.3 Errors

- There are no additional errors beyond those common to all requests.

4.3.3.4 Examples

Example 20 Message Listing Response with Header Info included

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageListingResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageInfo>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
      value="urn:provider:urology.ca:pplumber"/>
    <Subject>Referral Letter regarding patient: Everywoman, Eve</Subject>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
```



```
<MessageType type="ClinicalDocument" value="Referral"/>
<Date>2004-06-19T21:32:52.12679</Date>
<Read value="true" dateTime="2004-05-29T22:30:00.00000"/>
<Received value="false"/>
</MessageInfo>
<MessageInfo>
  <From displayName="Level Seven Healthcare Inc."
    value="urn:org:level7.healthcare.inc"/>
  <To displayName="Dr. Peter Plumber"
    value="urn:provider:urology.ca:pplumber"/>
  <MessageID>urn:uuid:AF6CD91C-0D0B-4d22-B141-7C7F4D152640</MessageID>
  <MessageType type="Notification" value="ReadConfirmation"/>
  <Date>2004-06-19T21:32:52.12679</Date>
</MessageInfo>
</MessageListingResponse>
```

Example 21 Message Listing Response – simple response

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageListingResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
  <MessageID>urn:uuid:AF6CD91C-0D0B-4d22-B141-7C7F4D152640</MessageID>
</MessageListingResponse>
```

4.3.4 MessageReadResponse

The MessageReadResponse is used to return from a MessageRead request. If an error occurred, it returns an Error element within the response.

4.3.4.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageReadResponse" type="MessageRefResponse"/>
</xs:schema>
```

4.3.4.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchMessage (100)
- MessageAlreadyRead (101)

4.3.4.3 Examples

Example 22 Message Read Response acknowledging the request

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageReadResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd"
  status="OK">
  <MessageID>urn:uuid:AF6CD91C-0D0B-4d22-B141-7C7F4D152640</MessageID>
</MessageReadResponse>
```

Example 23 Message Read Response indicating a No Such Message Error

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageReadResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
  <Error errorCode="100">
    <ErrorMsg>No such Message</ErrorMsg>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
  </Error>
</MessageReadResponse>
```

Example 24 Message Read Response indicating a Message Already Read Error

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageReadResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
  <Error errorCode="101">
    <ErrorMsg>Message Already Read</ErrorMsg>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
  </Error>
</MessageReadResponse>
```

4.3.5 MessageResetResponse

The MessageResetResponse is used to return from a MessageReset request. If an error occurred, it returns an Error element within the response.

4.3.5.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageResetResponse" type="MessageResetResponse"/>
  <xs:complexType name="MessageResetResponse">
    <xs:complexContent>
      <xs:extension base="ResponseBase">
        <xs:choice>
          <xs:sequence>
            <xs:element name="MessageCount" type="xs:long"/>
            <xs:choice minOccurs="0">
              <xs:element name="DateTime" type="xs:dateTime"/>
              <xs:element name="MessageID" type="MessageID"
maxOccurs="unbounded"/>
            </xs:choice>
          </xs:sequence>
          <xs:sequence>
            <xs:element name="Error" type="Error"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

4.3.5.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchMessage(100)
- NoDataFound(302)

4.3.5.3 Examples

Example 25 Message Reset Response indicating 3 messages were reset based on a date and time requested.

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageResetResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageCount>3</MessageCount>
  <DateTime>2004-05-26T21:32:52.12679</DateTime>
</MessageResetResponse>
```

Example 26 Message Reset Response indicating 6 messages were reset

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageResetResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageCount>6</MessageCount>
</MessageResetResponse>
```

Example 27 Reset Response indicating that the specified message was reset

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageResetResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageCount>1</MessageCount>
  <MessageID>urn:uuid:14E7125E-30BE-42c8-897A-61B8AB8B11B3</MessageID>
</MessageResetResponse>
```

Example 28 Reset Response indicating a No Such Message Error

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageResetResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance"
  :schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
  <Error errorCode="100">
    <ErrorMsg>No such Message</ErrorMsg>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
  </Error>
</MessageResetResponse>
```

Example 29 Reset Response indicating No Data Found – reset didn't reset any messages

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageResetResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance"
  :schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
  <Error errorCode="302">
    <ErrorMsg>No data found</ErrorMsg>
  </Error>
</MessageResetResponse>
```

4.3.6 MessageRecvResponse

The MessageRecvResponse is used to return a message from the server. If an error occurred, it returns an Error element within the response. The Message Content consists of either a system generated Notification, or an e-MS ClinicalDocument sent by a care provider followed by zero or more related Attachments (encoded as MIME using base-64).

4.3.6.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageRecvResponse" type="MessageRecvResponse.Any"/>

  <xs:complexType name="MessageRecvResponse.Any">
    <xs:complexContent>
      <xs:extension base="ResponseBase">
        <xs:choice>
          <xs:element name="Message" type="MessageRx.Any"/>
          <xs:element name="Error" type="Error"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:group name="MessageHeaderRxGrp">
    <xs:sequence>
      <xs:group ref="MessageHeaderGrp"/>
      <xs:element name="MessageType" type="MessageType" minOccurs="0"/>
      <xs:element name="Date" type="xs:dateTime"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="MessageRx.Any">
    <xs:sequence>
      <xs:group ref="MessageHeaderRxGrp"/>
      <xs:element name="Content" type="ContentRx.Any"/>
    </xs:sequence>
  </xs:complexType>

  <xs:group name="ContentGrp.Any">
    <xs:sequence>
      <xs:any namespace="##other" processContents="skip"/>
      <xs:element name="Attachment" type="Attachment" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>

  <xs:complexType name="ContentRx.Any">
    <xs:choice>
      <xs:group ref="ContentGrp.Any"/>
      <xs:element name="Notification" type="Notification"/>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="MessageEvent">
    <xs:sequence>
      <xs:element name="MessageInfo" type="MessageInfo"/>
    </xs:sequence>
    <xs:attribute name="type" type="MessageEventType" use="required"/>
  </xs:complexType>

  <xs:complexType name="Notification">
    <xs:sequence>
      <xs:element name="MessageEvent" type="MessageEvent" minOccurs="0"/>
      <xs:element name="Note" type="xs:normalizedString"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="MessageEvent">
    <xs:restriction base="xs:string">
```

```
<xs:enumeration value="NotRead-Purged">
</xs:enumeration>
<xs:enumeration value="NotRead-WillPurge">
</xs:enumeration>
<xs:enumeration value="NotRead-WithinExpectedDays">
</xs:enumeration>
<xs:enumeration value="ReadConfirmation">
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

4.3.6.2 Errors

In addition to the errors possible in all responses, the following error may be returned:

- NoSuchMessage(100)

4.3.6.3 Examples

Example 30 MessageRecv Response returning a Read Confirmation Notification

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageRecvResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <Message>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
      value="urn:provider:urology.ca:pplumber"/>
    <Subject>This is the subject</Subject>
    <Priority>Highest</Priority>
    <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
    <MessageType type="ClinicalDocument" value="Referral"/>
    <Date>2004-06-19T21:32:52.12679</Date>
    <Content>
      <Notification>
        <MessageEvent type="ReadConfirmation">
          <MessageInfo>
            <From displayName="Dr. Peter Plumber"
              value="urn:provider:urology.ca:pplumber"/>
            <To displayName="Dr. Patricia Primary"
              value="urn:provider:level7.healthcare.inc:pprimary"/>
            <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-
              3971D6408A30</MessageID>
            <MessageType type="ClinicalDocument" value="Referral"/>
            <Date>2004-06-19T21:32:52.12679</Date>
            <Read value="true"/>
            <Received value="false"/>
          </MessageInfo>
        </MessageEvent>
        <Note>The message was delivered successfully</Note>
      </Notification>
    </Content>
  </Message>
</MessageRecvResponse>
```

Example 31 Message Receive Response returning a Clinical Document with Attachment

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageRecvResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <Message>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
```

```

1722     <To displayName="Dr. Peter Plumber"
1723 value="urn:provider:urology.ca:pplumber"/>
1724     <Subject>This is the subject</Subject>
1725     <Priority>Highest</Priority>
1726     <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>
1727     <Date>2004-06-19T21:32:52.12679</Date>
1728     <Content>
1729       <ClinicalDocument
1730         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1731         xsi:schemaLocation="urn:hl7-org:v3 EMS_CDA.xsd"
1732         xmlns="urn:hl7-org:v3">
1733         <id/>
1734         <code/>
1735         <effectiveTime/>
1736         <author>
1737           <time/>
1738           <assignedAuthor>
1739             <id/>
1740           </assignedAuthor>
1741         </author>
1742         <custodian>
1743           <assignedCustodian/>
1744         </custodian>
1745         <recordTarget>
1746           <patientRole>
1747             <id/>
1748           </patientRole>
1749         </recordTarget>
1750         <component>
1751           <bodyChoice>
1752             <StructuredBody>
1753               <component>
1754                 <section>
1755                   <entry>
1756                     <entryChoice>
1757                       <ObservationMedia>
1758                         <value mediaType="image/jpeg"
1759 encoding="B64">
1760                           <reference value="test.jpg"/>
1761                         </value>
1762                       </ObservationMedia>
1763                     </entryChoice>
1764                   </entry>
1765                 </section>
1766               </component>
1767             </StructuredBody>
1768           </bodyChoice>
1769         </component>
1770       </ClinicalDocument>
1771       <Attachment fileName="test.jpg"> MIME-Version: 1.0 Content-Type:
1772 image/jpeg;
1773         name="test.jpg" Content-Transfer-Encoding: base64 Content-
1774 Description: test.jpg
1775         Content-Disposition: attachment; filename="test.jpg" Content-
1776 Location: test.jpg
1777         naXZlIG9ldCB5b3VyIHBhc3N3b3JkIG9yIGNyZWVpdCBjYXJkIG51bWJlciBpbiBh
1778 biBpbnN0YW50IG1lc3NhZ2UgY29udmVyc2F0aW9uLg0KICANC1Zlcm9uaWNhIFJ5YW4gKFZvbHQ
1779 DQpoaSBhZGUNC1ZlcYSBSeWFWuICbWb2x0KSBzYXlzMg0KaSBoYXZlIGVgcHJl
1780 bWllciBjYXN5YmFjaw0KQWRlIEZhbW90aSBzYXlzMg0KSSJtIG9uIHRoZSBwaG9uZSB3aXR0IGE
1781 </Attachment>
1782 </Content></Message>
1783 </MessageRecvResponse>

```

4.3.7 MessageSendResponse

The MessageSendResponse is returned to the EMR as a result of a MessageSend request. It returns a unique message ID that was created when the message was stored. This message ID

1789 may be used by the sender to track the status of the sent message, and to match against delivery
1790 notifications received. If an error occurred, it returns an `Error` element within the response.

1791 4.3.7.1 Definition

```
1792 <?xml version="1.0" encoding="UTF-8"?>  
1793 <xs:schema  
1794   xmlns="http://www.e-ms.ca/Schema/v1"  
1795   xmlns:b="http://schemas.microsoft.com/BizTalk/2003"  
1796   xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"  
1797   targetNamespace="http://www.e-ms.ca/Schema/v1"  
1798   xmlns:xs="http://www.w3.org/2001/XMLSchema">  
1799  
1800   <xs:include schemaLocation="admin/admin.xsd"/>  
1801   <xs:include schemaLocation="query/query.xsd"/>  
1802   <xs:include schemaLocation="messaging/messaging.xsd"/>  
1803  
1804   <xs:element name="MessageSendResponse" type="MessageRefResponse"/>  
1805 </xs:schema>
```

1806 4.3.7.2 Errors

1807 In addition to the errors common to all requests, the following errors may be returned:

- 1808 • `UnsupportedMessage(103)`
- 1809 • `PermissionError(200)`
- 1810 • `NoSuchAddress(300)`
- 1811 • `ValidationError(400)`
- 1812 • `NotWellFormedError(401)`

1813 4.3.7.3 Examples

1814 *Example 32 Send Response with message ID returned.*

```
1815 <?xml version="1.0" encoding="UTF-8"?>  
1816 <MessageSendResponse xmlns="http://www.e-ms.ca/Schema/v1"  
1817   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
1818   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">  
1819   <MessageID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</MessageID>  
1820 </MessageSendResponse>
```

1821 *Example 33 Message Send Response returning an Error indicating No Such Address*

```
1822 <?xml version="1.0" encoding="UTF-8"?>  
1823 <MessageSendResponse xmlns="http://www.e-ms.ca/Schema/v1"  
1824   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
1825   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">  
1826   <Error errorCode="300">  
1827     <ErrorMsg>No such address: 'urn:provider:nobody'</ErrorMsg>  
1828   </Error>  
1829 </MessageSendResponse>
```

1830 4.3.8 MessageTrackResponse

1831 The `MessageTrackResponse` is used to return from a `MessageTrack` request. If an error
1832 occurred, it returns an `Error` element within the response and the `status` attribute is set to
1833 `ERROR`. Messages may not be tracked unless sent by a care provider that has granted rights to
1834 the user issuing the `MessageTrack` request. An error is returned when a message ID is
1835 specified and is a valid message ID, but it doesn't represent a message sent by the care provider
1836 that the user is granted access to.

4.3.8.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="MessageTrackResponse" type="MessageTrackResponse"/>

  <xs:complexType name="MessageTrackResponse">
    <xs:complexContent>
      <xs:extension base="ResponseBase">
        <xs:choice>
          <xs:sequence>
            <xs:element name="MessageInfo" type="MessageInfo"/>
          </xs:sequence>
          <xs:sequence>
            <xs:element name="Error" type="Error"/>
          </xs:sequence>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

4.3.8.2 Errors

In addition to the exceptions common to all requests, the following error may be returned:

- NoSuchMessage(100)

4.3.8.3 Examples

Example 34 Track Response indicating the message not yet been received or read

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageTrackResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <MessageInfo>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
      value="urn:provider:urology.ca:pplumber"/>
    <MessageID>urn:uuid:14E7125E-30BE-42c8-897A-61B8AB8B11B3</MessageID>
    <Date>2004-08-22T11:32:52.12679</Date>
    <Read value="false"/>
    <Received value="false"/>
  </MessageInfo>
</MessageTrackResponse>
```

Example 35 Track Response indicating the message has been received but not read

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageTrackResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageInfo>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
      value="urn:provider:urology.ca:pplumber"/>
    <MessageID>urn:uuid:14E7125E-30BE-42c8-897A-61B8AB8B11B3</MessageID>
```



```
<Date>2004-08-22T11:32:52.12679</Date>
<Read value="false"/>
<Received value="true" dateTime="2004-06-15T20:30:00"/>
</MessageInfo>
</MessageTrackResponse>
```

*Example 36 Track Response indicating the message has been read but not received**

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageTrackResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageInfo>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
      value="urn:provider:urology.ca:pplumber"/>
    <MessageID>urn:uuid:14E7125E-30BE-42c8-897A-61B8AB8B11B3</MessageID>
    <Date>2004-08-22T11:32:52.12679</Date>
    <Read value="true" dateTime="2004-06-16T09:44:00"/>
    <Received value="false"/>
  </MessageInfo>
</MessageTrackResponse>
```

**This example may occur when the message has been read using the e-MS web application, but not yet received by the EMR system.*

Example 37 Track Response indicating the message has been received and read

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageTrackResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="OK">
  <MessageInfo>
    <From displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
    <To displayName="Dr. Peter Plumber"
      value="urn:provider:urology.ca:pplumber"/>
    <MessageID>urn:uuid:14E7125E-30BE-42c8-897A-61B8AB8B11B3</MessageID>
    <Date>2004-08-22T11:32:52.12679</Date>
    <Read value="true" dateTime="2004-06-16T09:44:00"/>
    <Received value="true" dateTime="2004-06-15T20:30:00"/>
  </MessageInfo>
</MessageTrackResponse>
```

4.4 Messaging Data Types XML Schema

The common data types used in Messaging Service Interface are defined in `messaging.xsd`, a WXS schema file that is included in the eMSEP schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:emsep="http://www.e-
ms.ca/Schema/v1"
  xmlns="http://www.e-ms.ca/Schema/v1" elementFormDefault="qualified">
  <xs:include schemaLocation="../dt/DataTypes.xsd"/>
  <!--
e-MS Project: Messaging types Schema for eMS Message Protocol.
Author: Brad Head <mailto:bradhead@sierrasystems.com>
Date: Jun 04, 2004

$Revision: 22 $

Copyright (c) 2004 by Vancouver Island Health Authority. All rights reserved.
-->
```

```
<xs:simpleType name="MessageID">
  <xs:restriction base="URN.UUID"/>
</xs:simpleType>
<xs:simpleType name="PriorityType">
  <xs:annotation>
    <xs:documentation>Based upon common Mail X-Priority
extension</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:token">
    <xs:enumeration value="Highest"/>
    <xs:enumeration value="High"/>
    <xs:enumeration value="Normal"/>
    <xs:enumeration value="Low"/>
    <xs:enumeration value="Lowest"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MessageTypes">
  <xs:restriction base="xs:token">
    <xs:enumeration value="ClinicalDocument"/>
    <xs:enumeration value="Notification"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="MessageType">
  <xs:complexContent>
    <xs:extension base="Coded">
      <xs:attribute name="type" type="MessageTypes" use="required"/>
      <xs:attribute name="value" type="xs:token" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:group name="MessageHeaderGrp">
  <xs:sequence>
    <xs:element name="From" type="MessagingAddress"/>
    <xs:element name="To" type="MessagingAddress"/>
    <xs:element name="Subject" type="xs:string" minOccurs="0"/>
    <xs:element name="Priority" type="PriorityType" minOccurs="0"/>
    <xs:element name="MessageID" type="MessageID"/>
  </xs:sequence>
</xs:group>
<xs:group name="MessageHeaderRxGrp">
  <xs:sequence>
    <xs:group ref="MessageHeaderGrp"/>
    <xs:element name="MessageType" type="MessageType" minOccurs="0"/>
    <xs:element name="Date" type="xs:dateTime"/>
  </xs:sequence>
</xs:group>
<xs:complexType name="MessageRequestBase">
  <xs:sequence>
    <xs:element name="MessageID" type="MessageID"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MessageInfo">
  <xs:sequence>
    <xs:group ref="MessageHeaderRxGrp"/>
    <xs:element name="Read" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="value" type="xs:boolean" use="required"/>
        <xs:attribute name="dateTime" type="xs:dateTime"
use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="Received" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="value" type="xs:boolean" use="required"/>
        <xs:attribute name="dateTime" type="xs:dateTime"
use="optional"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
```

```
</xs:complexType>
<xs:complexType name="MessageRefResponse">
  <xs:complexContent>
    <xs:extension base="ResponseBase">
      <xs:choice>
        <xs:sequence>
          <xs:element name="MessageID" type="MessageID"/>
        </xs:sequence>
        <xs:sequence>
          <xs:element name="Error">
            <xs:complexType mixed="true">
              <xs:complexContent mixed="true">
                <xs:extension base="Error"/>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="AttachmentFormat">
  <xs:restriction base="xs:token">
    <xs:enumeration value="MIME"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MIME">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse"/>
    <xs:minLength value="15"/>
    <xs:pattern value=".*MIME-Version.*"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Attachment">
  <xs:simpleContent>
    <xs:extension base="MIME">
      <xs:attribute name="fileName" type="xs:anyURI" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="MessageEventType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotRead-Purged">
      <xs:annotation>
        <xs:appinfo>This notification is sent to the intended
recipient via Message Box
and, if not marked as received, to the notification email
address of the
recipient when the unread message has been deleted due to
the message
retention time period expiring. </xs:appinfo>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="NotRead-WillPurge">
      <xs:annotation>
        <xs:appinfo>A message has not been read and will soon be
deleted. This event may
be sent to the message recipient after several system-
configurable warning
periods. If the message has not been marked as received,
this notification
is also sent to the notification email address of the
recipient.</xs:appinfo>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="NotRead-WithinExpectedDays">
      <xs:annotation>
        <xs:appinfo>This notification is sent when a sent message has
not been read
```

```
2096         within the expected number of days. The number of days is
2097 set by the sender..</xs:appinfo>
2098         </xs:annotation>
2099         </xs:enumeration>
2100         <xs:enumeration value="ReadConfirmation">
2101         </xs:annotation>
2102         <xs:appinfo>Message has been read. This notification is
2103 (optionally) sent to the
2104         sender and to recipient if not yet marked as received by
2105 the EMR, but marked
2106         as read by non-EMR user via the e-MS Web Application. The
2107 sender controls
2108         whether they want this notification with the ReadReceipt
2109 preference.</xs:appinfo>
2110         </xs:annotation>
2111         </xs:enumeration>
2112     </xs:restriction>
2113 </xs:simpleType>
2114 <xs:complexType name="MessageEvent">
2115     <xs:sequence>
2116         <xs:element name="MessageInfo" type="MessageInfo"/>
2117     </xs:sequence>
2118     <xs:attribute name="type" type="MessageEventType" use="required"/>
2119 </xs:complexType>
2120 <xs:complexType name="Notification">
2121     <xs:sequence>
2122         <xs:element name="MessageEvent" type="MessageEvent" minOccurs="0"/>
2123         <xs:element name="Note" type="xs:normalizedString"/>
2124     </xs:sequence>
2125 </xs:complexType>
2126 <!-- Message Definitions = Generic Message and ClinicalDocument messages -->
2127 <xs:group name="ContentGrp.Any">
2128     <xs:sequence>
2129         <xs:any namespace="##other" processContents="skip"/>
2130         <xs:element name="Attachment" type="Attachment" minOccurs="0"
2131 maxOccurs="unbounded"/>
2132     </xs:sequence>
2133 </xs:group>
2134 <xs:complexType name="ContentRx.Any">
2135     <xs:choice>
2136         <xs:group ref="ContentGrp.Any"/>
2137         <xs:element name="Notification" type="Notification"/>
2138     </xs:choice>
2139 </xs:complexType>
2140 <xs:complexType name="ContentTx.Any">
2141     <xs:group ref="ContentGrp.Any"/>
2142 </xs:complexType>
2143 <xs:complexType name="MessageRx.Any">
2144     <xs:sequence>
2145         <xs:group ref="MessageHeaderRxGrp"/>
2146         <xs:element name="Content" type="ContentRx.Any"/>
2147     </xs:sequence>
2148 </xs:complexType>
2149 <xs:complexType name="MessageTx.Any">
2150     <xs:sequence>
2151         <xs:group ref="MessageHeaderGrp"/>
2152         <xs:element name="Content" type="ContentTx.Any"/>
2153     </xs:sequence>
2154 </xs:complexType>
2155 <!-- ===== Complex Types for Root Elements
2156 ===== -->
2157 <xs:complexType name="MessageAck">
2158     <xs:complexContent>
2159         <xs:extension base="MessageRequestBase"/>
2160     </xs:complexContent>
2161 </xs:complexType>
2162 <xs:complexType name="MessageAckResponse">
2163     <xs:complexContent>
2164         <xs:extension base="MessageRefResponse"/>
2165     </xs:complexContent>
```

```
2166 </xs:complexType>
2167 <xs:complexType name="MessageBoxStatus">
2168   <xs:complexContent>
2169     <xs:restriction base="EmptyElement"/>
2170   </xs:complexContent>
2171 </xs:complexType>
2172 <xs:complexType name="MessageBoxStatusResponse">
2173   <xs:complexContent>
2174     <xs:extension base="ResponseBase">
2175       <xs:choice>
2176         <xs:sequence>
2177           <xs:element name="MessageCount" type="xs:long"/>
2178         </xs:sequence>
2179         <xs:sequence>
2180           <xs:element name="Error" type="Error"/>
2181         </xs:sequence>
2182       </xs:choice>
2183     </xs:extension>
2184   </xs:complexContent>
2185 </xs:complexType>
2186 <xs:complexType name="MessageListing">
2187   <xs:attribute name="showHeaderInfo" type="xs:boolean" use="optional"
2188 default="false"/>
2189   <xs:attribute name="received" type="xs:boolean" use="optional"
2190 default="false"/>
2191 </xs:complexType>
2192 <xs:complexType name="MessageListingResponse">
2193   <xs:complexContent>
2194     <xs:extension base="ResponseBase">
2195       <xs:choice>
2196         <xs:sequence>
2197           <xs:element name="MessageInfo" type="MessageInfo"
2198 minOccurs="0" maxOccurs="unbounded"/>
2199         </xs:sequence>
2200         <xs:sequence>
2201           <xs:element name="MessageID" type="MessageID"
2202 minOccurs="0" maxOccurs="unbounded"/>
2203         </xs:sequence>
2204         <xs:sequence>
2205           <xs:element name="Error" type="Error"/>
2206         </xs:sequence>
2207       </xs:choice>
2208     </xs:extension>
2209   </xs:complexContent>
2210 </xs:complexType>
2211 <xs:complexType name="MessageRead">
2212   <xs:complexContent>
2213     <xs:extension base="MessageRequestBase"/>
2214   </xs:complexContent>
2215 </xs:complexType>
2216 <xs:complexType name="MessageReadResponse">
2217   <xs:complexContent>
2218     <xs:extension base="MessageRefResponse"/>
2219   </xs:complexContent>
2220 </xs:complexType>
2221 <xs:complexType name="MessageRecv">
2222   <xs:complexContent>
2223     <xs:extension base="MessageRequestBase"/>
2224   </xs:complexContent>
2225 </xs:complexType>
2226 <xs:complexType name="MessageRecvResponse.Any">
2227   <xs:complexContent>
2228     <xs:extension base="ResponseBase">
2229       <xs:choice>
2230         <xs:element name="Message" type="MessageRx.Any"/>
2231         <xs:element name="Error" type="Error"/>
2232       </xs:choice>
2233     </xs:extension>
2234   </xs:complexContent>
2235 </xs:complexType>
```

```
2236 <xs:complexType name="MessageReset">
2237   <xs:sequence>
2238     <xs:choice minOccurs="0">
2239       <xs:element name="DateTime" type="xs:dateTime"/>
2240       <xs:element name="MessageID" type="MessageID"/>
2241     </xs:choice>
2242   </xs:sequence>
2243 </xs:complexType>
2244 <xs:complexType name="MessageResetResponse">
2245   <xs:complexContent>
2246     <xs:extension base="ResponseBase">
2247       <xs:choice>
2248         <xs:sequence>
2249           <xs:element name="MessageCount" type="xs:long"/>
2250           <xs:choice minOccurs="0">
2251             <xs:element name="DateTime" type="xs:dateTime"/>
2252             <xs:element name="MessageID" type="MessageID"
2253 maxOccurs="unbounded"/>
2254           </xs:choice>
2255         </xs:sequence>
2256       <xs:sequence>
2257         <xs:element name="Error" type="Error"/>
2258       </xs:sequence>
2259     </xs:choice>
2260   </xs:extension>
2261 </xs:complexContent>
2262 </xs:complexType>
2263 <xs:complexType name="MessageSend.Any">
2264   <xs:sequence>
2265     <xs:element name="Message" type="MessageTx.Any"/>
2266   </xs:sequence>
2267 </xs:complexType>
2268 <xs:complexType name="MessageSendResponse">
2269   <xs:complexContent>
2270     <xs:extension base="MessageRefResponse"/>
2271   </xs:complexContent>
2272 </xs:complexType>
2273 <xs:complexType name="MessageTrack">
2274   <xs:complexContent>
2275     <xs:extension base="MessageRequestBase"/>
2276   </xs:complexContent>
2277 </xs:complexType>
2278 <xs:complexType name="MessageTrackResponse">
2279   <xs:complexContent>
2280     <xs:extension base="ResponseBase">
2281       <xs:choice>
2282         <xs:sequence>
2283           <xs:element name="MessageInfo" type="MessageInfo"/>
2284         </xs:sequence>
2285       <xs:sequence>
2286         <xs:element name="Error" type="Error"/>
2287       </xs:sequence>
2288     </xs:choice>
2289   </xs:extension>
2290 </xs:complexContent>
2291 </xs:complexType>
2292 </xs:schema>
```

2293

2294

5 Query Services Interface

5.1 Overview

The purpose of the Query Services interface is to support looking up users, care providers and organizations. The results of a query are then used as inputs to the exchange messaging services or administrative services. The query responses provide key identifiers and messaging addresses for organizations and care providers.

5.2 Query Request Messages

The Query Request messages consist of a choice of the following requests:

- OrganizationQuery
- ProviderQuery
- UserQuery

5.2.1 Organization Query Request

The `OrgQuery` organization request is used to search for an organization or get details of an organization.

The search results contain essential summary information for each organization.

A detail lookup does not include search parameters, but rather one or more organization identifiers – the unique keys used to get the details of an organization.

An error is returned when either the search parameters were invalid or missing. When the search parameters are too vague, an error message is returned indicating that too many results would have been returned, and a search refinement is necessary. An error is also returned during a detail lookup when either the organization ID is invalid or missing.

5.2.1.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="OrgQuery" type="OrganizationQuery"/>
</xs:schema>

<xs:complexType name="OrganizationQuery">
  <xs:sequence>
    <xs:choice>
      <xs:element name="MessagingAddress" type="MessagingAddress"/>
      <xs:group ref="OrgSearchParamGrp"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```


5.2.1.2 Returns

This request returns an `OrgQueryResponse`. See section 5.3.1 for details.

5.2.1.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- `NoSuchAddress` (300)
- `ValidationError` (400)
- `NotWellFormedError`(401)
- `TooManyResults` (430)

5.2.1.4 Examples

Example 38 Query Request for Organization details passing messaging address

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgQuery xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <MessagingAddress value="urn:org:levelseven.healthcare.inc" />
</OrgQuery>
```

Example 39 Query Request for Organization passing search parameters

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgQuery xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Name>Level Seven Healthcare, Inc.</Name>
  <Telecom value="tel:+250-555-1111" use="WP"/>
  <PostalAddress>
    <city>Victoria</city>
    <streetName>Healthcare Drive</streetName>
    <country>Canada</country>
    <houseNumber>1001</houseNumber>
  </PostalAddress>
</OrgQuery>
```

5.2.2 Provider Query Request

The `ProviderQuery` request is used to query for a care provider. The query takes two forms: a search form, with search arguments, and a query where the care provider's URN is passed as a single argument. The first type returns a summary of the care providers matching, including the URN. The latter returns the details of the specified care provider. During a search, a summary row is returned for each distinctly addressable care provider. If a care provider is a member of more than one organization, one row per membership is returned.

5.2.2.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="ProviderQuery" type="ProviderQuery"/>
```



```
<xs:complexType name="ProviderQuery">
  <xs:choice>
    <xs:element name="ProviderID" type="URN.UUID"/>
    <xs:element name="MessagingAddress"
      type="MessagingAddress.Provider"/>
    <xs:element name="Listing" type="ProviderListing"/>
    <xs:sequence minOccurs="1" maxOccurs="7">
      <xs:group ref="ProviderSearchGrp"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>
</xs:schema>
```

5.2.2.2 Returns

This request returns a `QueryProviderResponse`. See section 0 for details.

5.2.2.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchAddress (300)
- ValidationError (400)
- TooManyResults (430)

5.2.2.4 Examples

Example 40 Query Request for Provider details passing the Provider's unique messaging address

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderQuery xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <MessagingAddress
    value="urn:provider:level7.healthcare.inc:pprimary"/>
</ProviderQuery>
```

Example 41 Query Request for Provider passing search parameters

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderQuery xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd">
  <Name>Patricia</Name>
  <PostalAddress>
    <city>Victoria</city>
    <streetName>Healthcare</streetName>
    <country>Canada</country>
    <state>BC</state>
    <houseNumber>1001</houseNumber>
  </PostalAddress>
  <Telecom value="tel:+2503851535"/>
  <Specialty>neurology</Specialty>
  <Organization>
    <Name>Level Seven</Name>
  </Organization>
</ProviderQuery>
```

Example 42 Query Request for Provider Listing for all records modified since a date/time

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderQuery xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd">
  <Listing modifiedSince="2004-11-19T21:00:00.00"/>
</ProviderQuery>
```

5.2.3 User Query Request

The `UserQuery` request is used to query a given User. Finding an existing User is useful when adding a user to more than one organization, or adding the user as a delegate.

User names are unique across the entire system. If a user has an `EMR` role or an `OrgAdministrator` user role, then they may only be associated with one organization.

5.2.3.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="UserQuery" type="UserQuery"/>

  <xs:complexType name="UserQuery">
    <xs:choice>
      <xs:element name="UserName" type="UserName"/>
      <xs:group ref="UserQueryParamGrp"/>
    </xs:choice>
  </xs:complexType>
  <xs:group name="UserQueryParamGrp">
    <xs:sequence>
      <xs:element name="FullName" type="xs:string" minOccurs="0"/>
      <xs:element name="Role" type="UserRole" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="Organization" type="OrganizationQuery"
minOccurs="0"/>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

5.2.3.2 Returns

This request returns a `QueryUserResponse`. See section 0 for details.

5.2.3.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- `NoSuchKeyIdentifier` (420)
- `ValidationError` (400)
- `NotWellFormedError`(401)
- `TooManyResults` (430)

5.2.3.4 Examples

Example 43 Query Request for User details passing the Username (unique key)

```
<?xml version="1.0" encoding="UTF-8"?>
<UserQuery xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
  <UserName>patricia.primary</UserName>
```

2499 </UserQuery>

2500 *Example 44 Query Request for User passing search parameters*

```
2501     <?xml version="1.0" encoding="UTF-8"?>
2502     <UserQuery xmlns="http://www.e-ms.ca/Schema/v1"
2503       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2504       xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd">
2505       <FullName>Patricia Primary</FullName>
2506       <Role>Provider</Role>
2507       <Organization>
2508         <Name>Level Seven</Name>
2509       </Organization>
2510     </UserQuery>
```

2511 5.3 Query Response Messages

2512 The Query Response messages consist of one of the following root elements:

- 2513 • OrgQueryResponse
- 2514 • ProviderQueryResponse
- 2515 • UserQueryResponse

2516 5.3.1 Organization Query Response

2517 An OrgQueryResponse is the response returned from an OrgQuery request. The response
2518 result comes in two flavors – a summary result based on a parameterized search and a detail
2519 result based on a query using an organization address.

2520 If an error occurred, it returns an Error element within the response and the response status
2521 attribute is set to ERROR.

2522 5.3.1.1 Definition

```
2523     <?xml version="1.0" encoding="UTF-8"?>
2524     <xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
2525       xmlns="http://www.e-ms.ca/Schema/v1"
2526       xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
2527       xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
2528       targetNamespace="http://www.e-ms.ca/Schema/v1"
2529       xmlns:xs="http://www.w3.org/2001/XMLSchema">
2530       <xs:include schemaLocation="admin/admin.xsd"/>
2531       <xs:include schemaLocation="query/query.xsd"/>
2532       <xs:include schemaLocation="messaging/messaging.xsd"/>
2533
2534       <xs:element name="OrgQueryResponse" type="OrgQueryResponse"/>
2535
2536       <xs:complexType name="OrgQueryResponse">
2537         <xs:complexContent>
2538           <xs:extension base="ResponseBase">
2539             <xs:choice>
2540               <xs:element name="Organization"
2541                 type="OrganizationResult"
2542                 minOccurs="0" maxOccurs="unbounded"/>
2543               <xs:sequence>
2544                 <xs:element name="Error" type="Error"/>
2545               </xs:sequence>
2546             </xs:choice>
2547           </xs:extension>
2548         </xs:complexContent>
2549       </xs:complexType>
2550
2551       <xs:complexType name="OrganizationResult">
2552         <xs:sequence>
2553           <xs:group ref="OrgDetailGrp"/>
2554         </xs:sequence>
```

```
2555     </xs:sequence>
2556     <xs:attribute name="view" type="ResultViewType"/>
2557 </xs:complexType>
2558
2559 </xs:schema>
```

5.3.1.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchAddress (300)
- ValidationError (400)
- NotWellFormedError(401)
- TooManyResults (430)

5.3.1.3 Examples

Example 45 Query Response showing summarized Organization search results

```
2568 <?xml version="1.0" encoding="UTF-8"?>
2569 <OrgQueryResponse xmlns="http://www.e-ms.ca/Schema/v1"
2570   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2571   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
2572   status="OK">
2573   <Organization view="Summary">
2574     <MessagingAddress value="urn:org:levelseven.inc"/>
2575     <Name>Level Seven Healthcare, Inc.</Name>
2576     <Telecom value="tel:+1-855-555-1001" use="WP"/>
2577     <Telecom value="mailto:info@levelseven.com"/>
2578     <PostalAddress>
2579       <streetName>Healthcare Drive</streetName>
2580       <city>Ann Arbor</city>
2581       <houseNumber>1001</houseNumber>
2582       <postalCode>99999</postalCode>
2583       <state>MI</state>
2584     </PostalAddress>
2585     <Note/>
2586   </Organization>
2587   <Organization view="Summary">
2588     <MessagingAddress value="urn:org:workisfun.inc"/>
2589     <Name>Work Is Fun Clinic</Name>
2590     <Telecom value="tel:+8855551234"/>
2591     <PostalAddress>
2592       <streetName>Healthcare Drive</streetName>
2593       <city>Ann Arbor</city>
2594       <houseNumber>1001</houseNumber>
2595       <postalCode>99999</postalCode>
2596       <state>MI</state>
2597     </PostalAddress>
2598     <Note/>
2599   </Organization>
2600 </OrgQueryResponse>
```

Example 46 Query Response showing detail of a single Organization

```
2602 <?xml version="1.0" encoding="UTF-8"?>
2603 <OrgQueryResponse xmlns="http://www.e-ms.ca/Schema/v1"
2604   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2605   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
2606   status="OK">
2607   <Organization view="Detail">
2608     <Name>PJ Neuro Clinic</Name>
2609     <Telecom value="fax:+16045551234"/>
2610     <Telecom value="tel:+2503851535"/>
2611     <PostalAddress>
2612       <streetName>Healthcare Street</streetName>
2613       <country>Canada</country>
```

```
<streetAddressLine>Suite 500</streetAddressLine>
<city>Victoria</city>
<houseNumber>1001</houseNumber>
<postalCode>V8W 2B9</postalCode>
<state>BC</state>
</PostalAddress>
<MessagingAddress>urn:org:pj-neuroclinic.bc.ca</MessagingAddress>
<Note>This clinic requires that you call ahead before sending referrals
electronically.</Note>
</Organization>
</OrgQueryResponse>
```

Example 47 Query Response showing zero results returned

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgQueryResponse xmlns="http://www.e-ms.ca/Schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema eMSEP.xsd" status="OK">
</OrgQueryResponse>
```

5.3.2 Provider Query Response

The `ProviderQueryResponse` is the response returned from a `ProviderQuery` request for Providers. The response result comes in two flavors – a summary result based on a parameterized search and a detail result based on a query using a care provider address.

If an error occurred, it returns an `Error` element within the response and the response `status` attribute is set to `ERROR`.

5.3.2.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="ProviderQueryResponse" type="ProviderQueryResponse"/>

  <xs:complexType name="ProviderQueryResponse">
    <xs:complexContent>
      <xs:extension base="ResponseBase">
        <xs:choice>
          <xs:element name="Provider" type="Provider" minOccurs="0"
maxOccurs="unbounded"/>
          <xs:sequence>
            <xs:element name="Error" type="Error"/>
          </xs:sequence>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

5.3.2.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchAddress (300)
- ValidationError (400)

- TooManyResults (430)

5.3.2.3 Examples

Example 48 Provider Query Response showing summarized Provider search results

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderQueryResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <Provider view="Summary">
    <ProviderID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</ProviderID>
    <PersonName><prefix xmlns="urn:hl7-org:v3">Dr.</prefix>
      <given xmlns="urn:hl7-org:v3">Peter</given>
      <family xmlns="urn:hl7-org:v3">Plumber</family>
      <suffix xmlns="urn:hl7-org:v3">MD</suffix>
      Dr. Peter Plumber
    </PersonName>
    <BillingNumber>23304495-011</BillingNumber>
    <Specialty>Urology</Specialty>
    <Membership>
      <MessagingAddress value="urn:provider:urology.inc:pplumber"/>
      <Organization>
        <MessagingAddress value="urn:org:urology.inc"/>
        <Name>PP Urology Inc.</Name>
      </Organization>
      <Note>Referrals must be sent using the e-MS system. Call for urgent
referrals</Note>
    </Membership>
    <Membership>
      <MessagingAddress
value="urn:provider:urology.clinic.bc.ca:pplumber"/>
      <Organization>
        <MessagingAddress value="urn:org:urology.clinic.bc.ca"/>
        <Name>Urology Clinic</Name>
      </Organization>
      <Note>Part time employed. Call before sending a referral</Note>
    </Membership>
  </Provider>
  <Provider view="Summary">
    <ProviderID>urn:uuid:24739DFB-F34E-44F9-A556-3971D6408A30</ProviderID>
    <PersonName>Dr. Susan Peters MD<prefix xmlns="urn:hl7-
org:v3">Dr.</prefix>
      <given xmlns="urn:hl7-org:v3">Susan</given>
      <family xmlns="urn:hl7-org:v3">Peters</family>
      <suffix xmlns="urn:hl7-org:v3">MD</suffix>
    </PersonName>
    <BillingNumber>23304495-011</BillingNumber>
    <Specialty>Neurology</Specialty>
    <Membership>
      <MessagingAddress value="urn:provider:pj-
neuroclinic.bc.ca:stevenjohn"/>
      <Organization>
        <Name>PJ Neurology Clinic</Name>
        <MessagingAddress>urn:pj-neuroclinic.bc.ca</MessagingAddress>
      </Organization>
      <Note>Neurological Referrals must be sent using the e-MS system. Call
for urgent referrals</Note>
    </Membership>
  </Provider>
</ProviderQueryResponse>
```

5.3.3 User Query Response

The UserQueryResponse is the response returned from a UserQuery request. The response result comes in two flavors – a summary result based on a parameterized search and a detail result based on a query using a username identifier.

2733 If an error occurred, it returns an `Error` element within the response and the
2734 `UserQueryResponse` status attribute is set to `ERROR`.

2735 5.3.3.1 Definition

```
2736 <?xml version="1.0" encoding="UTF-8"?>
2737 <xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
2738   xmlns="http://www.e-ms.ca/Schema/v1"
2739   xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
2740   xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
2741   targetNamespace="http://www.e-ms.ca/Schema/v1"
2742   xmlns:xs="http://www.w3.org/2001/XMLSchema">
2743   <xs:include schemaLocation="admin/admin.xsd"/>
2744   <xs:include schemaLocation="query/query.xsd"/>
2745   <xs:include schemaLocation="messaging/messaging.xsd"/>
2746
2747   <xs:element name="UserQueryResponse" type="UserQueryResponse"/>
2748
2749   <xs:complexType name="UserQueryResponse">
2750     <xs:complexContent>
2751       <xs:extension base="ResponseBase">
2752         <xs:choice>
2753           <xs:element name="User" type="UserResult" minOccurs="0"
2754             maxOccurs="unbounded"/>
2755           <xs:sequence>
2756             <xs:element name="Error" type="Error"/>
2757           </xs:sequence>
2758         </xs:choice>
2759       </xs:extension>
2760     </xs:complexContent>
2761   </xs:complexType>
2762   <xs:complexType name="UserResult">
2763     <xs:complexContent>
2764       <xs:extension base="User">
2765         <xs:attribute name="view" type="ResultViewType"/>
2766       </xs:extension>
2767     </xs:complexContent>
2768   </xs:complexType>
2769
2770 </xs:schema>
2771
```

2772 5.3.3.2 Errors

2773 In addition to the errors possible in all responses, the following errors may be returned:

- 2774 • NoSuchKeyIdentifier (403)
- 2775 • ValidationError (400)
- 2776 • TooManyResults (430)

2777 5.3.3.3 Examples

2778 *Example 49 Query Response showing summarized User search results*

```
2779 <?xml version="1.0" encoding="UTF-8"?>
2780 <UserQueryResponse xmlns="http://www.e-ms.ca/Schema/v1"
2781   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2782   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
2783   status="OK">
2784   <User view="Summary">
2785     <UserName>speters</UserName>
2786     <FullName>Dr. Susan Peters</FullName>
2787     <Affiliation>
2788       <Role value="Provider"/>
2789       <Organization>
2790         <MessagingAddress value="urn:org:pj-neuroclinic.bc.ca"/>

```



```

2791         </Organization>
2792     </Affiliation>
2793     <Affiliation>
2794         <Role value="Provider"/>
2795         <Organization>
2796             <MessagingAddress value="urn:org:level7.healthcare.inc"/>
2797         </Organization>
2798     </Affiliation>
2799 </User>
2800 <User view="Summary">
2801     <UserName>sjohn</UserName>
2802     <FullName>Dr. Steven John</FullName>
2803     <Affiliation>
2804         <Role value="Provider"/>
2805         <Organization>
2806             <MessagingAddress value="urn:org:pj-neuroclinic.bc.ca"/>
2807         </Organization>
2808     </Affiliation>
2809 </User>
2810 </UserQueryResponse>

```

2811 *Example 50 Query Response showing User Response Error – No Such Key Identifier*

```

2812 <?xml version="1.0" encoding="UTF-8"?>
2813 <UserQueryResponse xmlns="http://www.e-ms.ca/Schema/v1"
2814     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2815     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 eMSEP.xsd" status="ERROR">
2816     <Error errorCode="405">
2817         <ErrorMsg>No such key identifier</ErrorMsg>
2818     </Error>
2819 </UserQueryResponse>

```

2820 5.3.4 Query Data Types XML Schema

2821 The common data types used in Query Service Interface are defined in `query.xsd`, a WXS
2822 schema file that is included in the `eMSEP.xsd` schema file.

```

2823 <?xml version="1.0" encoding="UTF-8"?>
2824 <xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1" xmlns="http://www.e-
2825 ms.ca/Schema/v1"
2826     xmlns:emsep="http://www.e-ms.ca/Schema/v1"
2827     xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
2828     <xs:include schemaLocation="../dt/DataTypes.xsd"/>
2829     <!--
2830     e-MS Project: Query types Schema for eMS Message Protocol.
2831     Author: Brad Head <mailto:bradhead@sierrasystems.com>
2832     Date: Jun 04, 2004
2833
2834     $Revision: 9 $
2835
2836     Copyright (c) 2004 by Vancouver Island Health Authority. All rights reserved.
2837
2838     -->
2839     <xs:simpleType name="ResultViewType">
2840         <xs:restriction base="xs:NMTOKEN">
2841             <xs:enumeration value="Detail"/>
2842             <xs:enumeration value="Summary"/>
2843         </xs:restriction>
2844     </xs:simpleType>
2845     <xs:group name="OrgSearchParamGrp">
2846         <xs:sequence>
2847             <xs:element name="Name" type="OrgName" minOccurs="0" maxOccurs="2"/>
2848             <xs:element name="Telecom" type="Telecom" minOccurs="0"
2849 maxOccurs="unbounded"/>
2850             <xs:element name="PostalAddress" type="PostalAddress" minOccurs="0"/>
2851         </xs:sequence>
2852     </xs:group>
2853     <xs:complexType name="OrganizationQuery">
2854         <xs:sequence>

```



```
2855         <xs:choice>
2856             <xs:element name="MessagingAddress"
2857 type="MessagingAddress.Organization"/>
2858             <xs:group ref="OrgSearchParamGrp"/>
2859         </xs:choice>
2860     </xs:sequence>
2861 </xs:complexType>
2862 <xs:complexType name="OrgQueryResponse">
2863     <xs:complexContent>
2864         <xs:extension base="ResponseBase">
2865             <xs:choice>
2866                 <xs:element name="Organization" type="OrganizationResult"
2867 minOccurs="0" maxOccurs="unbounded"/>
2868                 <xs:sequence>
2869                     <xs:element name="Error" type="Error"/>
2870                 </xs:sequence>
2871             </xs:choice>
2872         </xs:extension>
2873     </xs:complexContent>
2874 </xs:complexType>
2875 <xs:complexType name="OrganizationResult">
2876     <xs:sequence>
2877         <xs:group ref="OrgDetailGrp"/>
2878     </xs:sequence>
2879     <xs:attribute name="view" type="ResultViewType"/>
2880 </xs:complexType>
2881 <xs:complexType name="ProviderQuery">
2882     <xs:choice>
2883         <xs:element name="ProviderID" type="ProviderID"/>
2884         <xs:element name="MessagingAddress"
2885 type="MessagingAddress.Provider"/>
2886         <xs:element name="Listing" type="ProviderListing"/>
2887         <xs:sequence minOccurs="1" maxOccurs="7">
2888             <xs:group ref="ProviderSearchGrp"/>
2889         </xs:sequence>
2890     </xs:choice>
2891 </xs:complexType>
2892 <xs:group name="ProviderSearchGrp">
2893     <xs:choice>
2894         <xs:element name="Name" type="PersonName"/>
2895         <xs:element name="PostalAddress" type="PostalAddress"/>
2896         <xs:element name="Telecom" type="Telecom" maxOccurs="unbounded"/>
2897         <xs:element name="BillingNumber" type="BillingNumber"/>
2898         <xs:element name="Specialty" type="ProviderSpecialty" maxOccurs="2"/>
2899         <xs:element name="AdministrativeGenderCode"
2900 type="AdministrativeGenderCode"/>
2901         <xs:element name="Organization" type="OrganizationQuery"/>
2902     </xs:choice>
2903 </xs:group>
2904 <xs:complexType name="ProviderListing">
2905     <xs:attribute name="modifiedSince" type="xs:dateTime" use="optional"/>
2906 </xs:complexType>
2907 <xs:complexType name="ProviderQueryResponse">
2908     <xs:complexContent>
2909         <xs:extension base="ResponseBase">
2910             <xs:choice>
2911                 <xs:element name="Provider" type="Provider" minOccurs="0"
2912 maxOccurs="unbounded"/>
2913                 <xs:sequence>
2914                     <xs:element name="Error" type="Error"/>
2915                 </xs:sequence>
2916             </xs:choice>
2917         </xs:extension>
2918     </xs:complexContent>
2919 </xs:complexType>
2920 <xs:complexType name="ProviderMembership">
2921     <xs:sequence>
2922         <xs:element name="MessagingAddress"
2923 type="MessagingAddress.Provider"/>
2924         <xs:element name="Organization" type="Organization"/>
```

```
2925     <xs:element name="Telecom" type="Telecom" minOccurs="0"
2926 maxOccurs="unbounded"/>
2927     <xs:element name="Note" type="xs:string" minOccurs="0"/>
2928   </xs:sequence>
2929 </xs:complexType>
2930 <xs:complexType name="Provider">
2931   <xs:sequence>
2932     <xs:element name="ProviderID" type="ProviderID"/>
2933     <xs:element name="PersonName" type="PersonName"/>
2934     <xs:element name="BillingNumber" type="BillingNumber" minOccurs="0"/>
2935     <xs:element name="Specialty" type="ProviderSpecialty" minOccurs="0"
2936 maxOccurs="unbounded"/>
2937     <xs:element name="AdministrativeGenderCode"
2938 type="AdministrativeGenderCode" minOccurs="0"/>
2939     <xs:element name="Note" type="xs:string" minOccurs="0"/>
2940     <xs:element name="PostalAddress" type="PostalAddress" minOccurs="0"/>
2941     <xs:element name="Telecom" type="Telecom" minOccurs="0"
2942 maxOccurs="unbounded"/>
2943     <xs:element name="Membership" type="ProviderMembership" minOccurs="0"
2944 maxOccurs="unbounded"/>
2945   </xs:sequence>
2946   <xs:attribute name="view" type="ResultViewType"/>
2947 </xs:complexType>
2948 <xs:complexType name="UserQuery">
2949   <xs:choice>
2950     <xs:element name="UserName" type="UserName"/>
2951     <xs:group ref="UserQueryParamGrp"/>
2952   </xs:choice>
2953 </xs:complexType>
2954 <xs:group name="UserQueryParamGrp">
2955   <xs:sequence>
2956     <xs:element name="FullName" type="xs:string" minOccurs="0"/>
2957     <xs:element name="Role" type="UserRole" minOccurs="0"
2958 maxOccurs="unbounded"/>
2959     <xs:element name="Organization" type="OrganizationQuery"
2960 minOccurs="0"/>
2961   </xs:sequence>
2962 </xs:group>
2963 <xs:complexType name="UserQueryResponse">
2964   <xs:complexContent>
2965     <xs:extension base="ResponseBase">
2966       <xs:choice>
2967         <xs:element name="User" type="UserResult" minOccurs="0"
2968 maxOccurs="unbounded"/>
2969         <xs:sequence>
2970           <xs:element name="Error" type="Error"/>
2971         </xs:sequence>
2972       </xs:choice>
2973     </xs:extension>
2974   </xs:complexContent>
2975 </xs:complexType>
2976 <xs:complexType name="UserResult">
2977   <xs:complexContent>
2978     <xs:extension base="User">
2979       <xs:attribute name="view" type="ResultViewType"/>
2980     </xs:extension>
2981   </xs:complexContent>
2982 </xs:complexType>
2983 </xs:schema>
```

6 Administrative Services Interface

6.1 Overview

The purpose of the Administrative Services interface is to support organization-level administration of users, care providers and organization settings.

6.1.1 Organization Administration

- Add, remove and list user affiliation with an organization.
- Add, remove and list user access rights to the organization's message box.
- Add, remove and list provider membership
- Maintain organization preferences regarding message delivery
- Maintain organization's unique system address
- Maintain organization details: name, address, telecom, notes, etc.

6.1.2 Provider Administration

- Add, remove and list user access rights to Provider
- Maintain provider preferences for message delivery
- Maintain provider's unique system address
- Maintain provider note

6.1.3 User Administration

- List access rights to Providers and Organizations

6.1.4 Preference Settings

Provider and Organization messaging preferences are maintained using the OrgAdmin and ProviderAdmin messages. These preferences may not be altered unless the User's access type is "Owner". A Delegate User may not alter the preferences.

Preference	Purpose
DeliveryNotification	Set to "true", the sender will receive a delivery notification when the sent message has been read by the intended recipient. The optional email address provides an additional means to receive the notification. Set to "false" disables the delivery notification.
MessageRetention	Sets the number of days to retain messages in the message box.
NotifySentUnread	This sets the number of days after which sent messages that haven't yet been read will result in a notification back to the sender.

3008

3009

6.2 Administrative Request Message

3010 The Administrative Request message consists of one of the following root elements:

- 3011 • OrgAdmin element
- 3012 • ProviderAdmin element
- 3013 • UserAdmin element

3014

6.2.1 Organization Administrative Request

3015 The OrgAdmin request is used to maintain organization data: user access, user affiliation,
3016 provider membership and other organization details.3017 The corresponding response is the OrgAdminResponse which indicates a successful request or
3018 an error.3019

6.2.1.1 Definition

```
3020 <?xml version="1.0" encoding="UTF-8"?>
3021 <xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
3022   xmlns="http://www.e-ms.ca/Schema/v1"
3023   xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
3024   xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
3025   targetNamespace="http://www.e-ms.ca/Schema/v1"
3026   xmlns:xs="http://www.w3.org/2001/XMLSchema">
3027   <xs:include schemaLocation="admin/admin.xsd"/>
3028   <xs:include schemaLocation="query/query.xsd"/>
3029   <xs:include schemaLocation="messaging/messaging.xsd"/>
3030
3031   <xs:element name="OrgAdmin" type="OrgAdmin"/>
3032
3033   <xs:complexType name="OrgAdmin">
3034     <xs:sequence>
3035       <xs:element name="Organization" type="Organization.Key"/>
3036       <xs:choice>
3037         <xs:element name="Access" type="AdminAccess"/>
3038         <xs:element name="Users" type="OrgAdminUsers"/>
3039         <xs:element name="Providers" type="OrgAdminProviders"/>
3040         <xs:element name="Preferences" type="AdminPreferences"/>
3041         <xs:element name="Details" type="OrgAdminDetails"/>
3042       </xs:choice>
3043     </xs:sequence>
3044   </xs:complexType>
3045
3046 </xs:schema>
```

3047

3048

6.2.1.2 Returns

3049 This request returns an Organization Administrative Response. See section 6.3.1 for details.

3050

6.2.1.3 Errors

3051 In addition to the errors possible in all responses, the following errors may be returned:

- 3052 • NoSuchAddress (300)
- 3053 • AddressInUse (301)
- 3054 • NoSuchKeyIdentifier (420)
- 3055 • ValidationError (400)

6.2.1.4 Examples: user access

Example 51 Add user access

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level7.healthcare.inc"/>
  </Organization>
  <Access>
    <Add>
      <UserName>pplumber</UserName>
      <Rights value="Delegate"/>
      <EffectiveTo>2004-05-26T21:32:52.12679</EffectiveTo>
    </Add>
  </Access>
</OrgAdmin>
```

Example 52 Remove user access

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Access>
    <Remove>
      <UserName>patricia.primary</UserName>
    </Remove>
  </Access>
</OrgAdmin>
```

Example 53 List user access

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Access>
    <List/>
  </Access>
</OrgAdmin>
```

6.2.1.5 Examples: user affiliation

Example 54 Affiliate a user with an organization

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Users>
    <Add>
      <UserName>pplumber</UserName>
      <Role value="OrgAdministrator"/>
      <Role value="Provider"/>
    </Add>
  </Users>
</OrgAdmin>
```

Example 55 Remove a user from a role within an organization

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Users>
    <Remove>
      <UserName>speters</UserName>
      <Role value="OrgAdministrator"/>
    </Remove>
  </Users>
</OrgAdmin>
```

Example 56 Remove a user from an organization

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Users>
    <Remove>
      <UserName>pplumber</UserName>
    </Remove>
  </Users>
</OrgAdmin>
```

Example 57 List users affiliated with an organization

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Users>
    <List/>
  </Users>
</OrgAdmin>
```

Example 58 List only Organization Administrator users affiliated with an organization

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
  </Organization>
  <Users>
    <List>
      <Role value="OrgAdministrator"/>
    </List>
  </Users>
</OrgAdmin>
```

6.2.1.6 Examples: provider membership

Example 59 List provider membership

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Organization>
    <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
```

```
3179     </Organization>
3180     <Providers>
3181         <List/>
3182     </Providers>
3183 </OrgAdmin>
```

3184 *Example 60 Add a provider to an organization*

```
3185 <?xml version="1.0" encoding="UTF-8"?>
3186 <OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3187     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3188     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3189     <Organization>
3190         <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
3191     </Organization>
3192     <Providers>
3193         <Add>
3194             <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-
3195 E9B3A5EFCEDA</ProviderID>
3196             <MessagingAddress displayName="Dr. Patricia Primary"
3197                 value="urn:provider:level7.healthcare.inc:pprimary"/>
3198         </Add>
3199     </Providers>
3200 </OrgAdmin>
```

3201 *Example 61 Remove a provider from an organization (by address)*

```
3202 <?xml version="1.0" encoding="UTF-8"?>
3203 <OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3204     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3205     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3206     <Organization>
3207         <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
3208     </Organization>
3209     <Providers>
3210         <Remove>
3211             <MessagingAddress
3212                 value="urn:provider:level7.healthcare.inc:pprimary"/>
3213         </Remove>
3214     </Providers>
3215 </OrgAdmin>
```

3216 *Example 62 Remove a provider from an organization (by provider ID)*

```
3217 <?xml version="1.0" encoding="UTF-8"?>
3218 <OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3219     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3220     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3221     <Organization>
3222         <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
3223     </Organization>
3224     <Providers>
3225         <Remove>
3226             <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-
3227 E9B3A5EFCEDA</ProviderID>
3228         </Remove>
3229     </Providers>
3230 </OrgAdmin>
```

3231 **6.2.1.7 Examples: organization preferences and details**

3232 *Example 63 Update organization preferences*

```
3233 <?xml version="1.0" encoding="UTF-8"?>
3234 <OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3235     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3236     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3237     <Organization>
3238         <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
3239     </Organization>
3240     <Preferences>
```



```
3241      <Update>
3242        <NotificationEmail>moa@levelseven.com</NotificationEmail>
3243        <ReadReceipt active="true"/>
3244        <NotReadNotification days="10"/>
3245        <MessageRetention days="60"/>
3246      </Update>
3247    </Preferences>
3248  </OrgAdmin>
```

3249 Example 64 View organization preferences

```
3250 <?xml version="1.0" encoding="UTF-8"?>
3251 <OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3252   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3253   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd">
3254   <Organization>
3255     <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
3256   </Organization>
3257   <Preferences>
3258     <View/>
3259   </Preferences>
3260 </OrgAdmin>
```

3261 Example 65 Update organization details – modify city

```
3262 <?xml version="1.0" encoding="UTF-8"?>
3263 <OrgAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3264   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3265   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd">
3266   <Organization>
3267     <MessagingAddress value="urn:org:level.seven.healthcare.inc"/>
3268   </Organization>
3269   <Details>
3270     <Update>
3271       <PostalAddress>
3272         <city>Ann Arbor</city>
3273       </PostalAddress>
3274     </Update>
3275   </Details>
3276 </OrgAdmin>
```

3277 6.2.2 Provider Administrative Request

3278 The `ProviderAdmin` request maintains provider data: user access, user affiliation, provider
3279 membership and other organization details.

3280 The corresponding response is the `ProviderAdminResponse` document, which indicates a
3281 successful request or an `Error`.

3282 6.2.2.1 Definition

```
3283 <?xml version="1.0" encoding="UTF-8"?>
3284 <xs:schema
3285   xmlns="http://www.e-ms.ca/Schema/v1"
3286   xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
3287   targetNamespace="http://www.e-ms.ca/Schema/v1"
3288   xmlns:xs="http://www.w3.org/2001/XMLSchema">
3289   <xs:include schemaLocation="admin/admin.xsd"/>
3290   <xs:include schemaLocation="query/query.xsd"/>
3291   <xs:include schemaLocation="messaging/messaging.xsd"/>
3292
3293   <xs:element name="ProviderAdmin" type="ProviderAdmin"/>
3294
3295   <xs:complexType name="ProviderAdmin">
3296     <xs:sequence>
3297       <xs:element name="Provider" type="Provider.MemberKey"/>
3298       <xs:choice>
3299         <xs:element name="Access" type="AdminAccess"/>
3300         <xs:element name="Preferences" type="AdminPreferences"/>
3301       </xs:choice>
3302     </xs:sequence>
3303   </xs:complexType>
```



```
<xs:element name="MessagingAddress"
            type="MessagingAddress.Provider"/>
<xs:element name="Note" type="xs:string"/>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

The definition reuses the data types `AdmAccess` and `AdmPreferences` as described previously.

6.2.2.2 Returns

This request returns a `ProviderAdminResponse`. See section 0 for details.

6.2.2.3 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchAddress (300)
- AddressInUse (301)
- NoSuchKeyIdentifier (420)
- ValidationError (400)

6.2.2.4 Examples

Example 66 Add user access rights to Provider Membership

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Provider>
    <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
  </Provider>
  <Access>
    <Add>
      <UserName>pplumber</UserName>
      <Rights value="Owner"/>
    </Add>
  </Access>
</ProviderAdmin>
```

Example 67 Remove user access to Provider membership

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <Provider>
    <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
  </Provider>
  <Access>
    <Remove>
      <UserName>speters</UserName>
    </Remove>
  </Access>
</ProviderAdmin>
```

Example 68 List user access to Provider membership

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
```

```
3354 <Provider>
3355 <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
3356 </Provider>
3357 <Access>
3358 <List/>
3359 </Access>
3360 </ProviderAdmin>
```

3361 *Example 69 Update Provider's messaging address*

```
3362 <?xml version="1.0" encoding="UTF-8"?>
3363 <ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3364 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3365 xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3366 <Provider>
3367 <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
3368 </Provider>
3369 <MessagingAddress value="urn:provider:level7:peter.plumber"/>
3370 </ProviderAdmin>
```

3371 *Example 70 Update Provider's note*

```
3372 <?xml version="1.0" encoding="UTF-8"?>
3373 <ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3374 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3375 xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3376 <Provider>
3377 <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
3378 </Provider>
3379 <Note>For urgent referrals please call before sending patient
3380 information</Note>
3381 </ProviderAdmin>
```

3382 *Example 71 Update Provider's preferences*

```
3383 <?xml version="1.0" encoding="UTF-8"?>
3384 <ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3385 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3386 xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3387 <Provider>
3388 <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
3389 </Provider>
3390 <Preferences>
3391 <Update>
3392 <NotificationEmail>pplumber@level7.org</NotificationEmail>
3393 <ReadReceipt active="true"/>
3394 <NotReadNotification days="10"/>
3395 <MessageRetention days="60"/>
3396 </Update>
3397 </Preferences>
3398 </ProviderAdmin>
```

3399 *Example 72 View Provider's preferences*

```
3400 <?xml version="1.0" encoding="UTF-8"?>
3401 <ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1"
3402 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3403 xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
3404 <Provider>
3405 <MessagingAddress value="urn:provider:level7.healthcare.inc:pplumber"/>
3406 </Provider>
3407 <Preferences>
3408 <View/>
3409 </Preferences>
3410 </ProviderAdmin>
```

3411 **6.2.3 User Administrative Request**

3412 The UserAdmin request is intended to list user access rights to provider and organization
3413 message boxes. The modification of these settings is maintained by the provider and organization

administrative areas. The purpose of this is to expose a convenient and secure view of the user's access. This request may only be issued by the user.

The corresponding response is the `UserAdminResponse` document, which indicates a successful request or an `Error`.

6.2.3.1 Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"
  xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified"
  targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="admin/admin.xsd"/>
  <xs:include schemaLocation="query/query.xsd"/>
  <xs:include schemaLocation="messaging/messaging.xsd"/>

  <xs:element name="UserAdmin" type="UserAdmin"/>

  <xs:complexType name="UserAdmin">
    <xs:sequence>
      <xs:element name="UserName" type="UserName"/>
      <xs:choice>
        <xs:element name="ListAccess">
          <xs:complexType>
            <xs:attribute name="list" fixed="true"/>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

6.2.3.2 Returns

In addition to the errors possible in all responses, the following errors may be returned:

- `PermissionError` (200)
- `NoSuchKeyIdentifier` (420)

6.2.3.3 Example

Example 73 Request User Access Listing

```
<?xml version="1.0" encoding="UTF-8"?>
<UserAdmin xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd">
  <UserName>speters</UserName>
  <ListAccess/>
</UserAdmin>
```

6.3 Administrative Response Message

An administrative response message is returned by the e-MS service when a corresponding administrative request is received. It consists of one of the following root elements:

- `OrgAdminResponse` element
- `ProviderAdminResponse` element
- `UserAdminResponse` element

6.3.1 Organization Administrative Response

The `OrgAdminResponse` is used to return a response to the `OrgAdmin` request described in section 6.2.1. If an error occurred, it returns an `Error` element within the response. This response is a return from the synchronous request.

6.3.1.1 Definition

```
<xs:element name="OrgAdminResponse" type="OrgAdminResponse"/>

<xs:complexType name="OrgAdminResponse">
  <xs:complexContent>
    <xs:extension base="ResponseBase">
      <xs:sequence>
        <xs:element name="Organization" type="Organization.Key"/>
        <xs:choice>
          <xs:element name="Access" type="AdminAccessResp"/>
          <xs:element name="Users" type="OrgAdminUsersResp"/>
          <xs:element name="Providers" type="OrgAdminMembersResp"/>
          <xs:element name="Preferences"
type="AdminPreferencesResp"/>
          <xs:element name="Details" type="OrgAdminDetailsResp"/>
          <xs:sequence>
            <xs:element name="Error" type="Error"/>
          </xs:sequence>
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Viewing organization details is covered by the `OrgQueryResponse` schema as described in section 5.3.1.

6.3.1.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchAddress (300)
- AddressInUse (301)
- NoSuchKeyIdentifier (420)
- ValidationError (400)

6.3.1.3 Examples: user access response

Example 74 User access added successfully

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
  status="OK">
  <Organization>
    <MessagingAddress value="urn:org:level7.healthcare.inc"/>
  </Organization>
  <Access>
    <Added>
      <UserName>speters</UserName>
    </Added>
  </Access>
</OrgAdminResponse>
```

3519 *Example 75 User access removed successfully*

```
3520 <?xml version="1.0" encoding="UTF-8"?>
3521 <OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
3522   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3523   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
3524   status="OK">
3525   <Organization>
3526     <MessagingAddress value="urn:org:level7.healthcare.inc"/>
3527   </Organization>
3528   <Access>
3529     <Removed>
3530       <UserName>speters</UserName>
3531     </Removed>
3532   </Access>
3533 </OrgAdminResponse>
```

3534 *Example 76 User access listing successfully*

```
3535 <?xml version="1.0" encoding="UTF-8"?>
3536 <OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
3537   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3538   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
3539   status="OK">
3540   <Organization>
3541     <MessagingAddress value="urn:org:level7.healthcare.inc"/>
3542   </Organization>
3543   <Access>
3544     <Listing>
3545       <User>
3546         <UserName>speters</UserName>
3547         <FullName>Dr. Susan Peters</FullName>
3548         <Access>
3549           <Rights value="Owner"/>
3550           <EffectiveFrom>2004-06-20T20:00:00</EffectiveFrom>
3551         </Access>
3552       </User>
3553       <User>
3554         <UserName>mjones</UserName>
3555         <Access>
3556           <Rights value="Delegate"/>
3557           <EffectiveFrom>2004-06-23T18:00:00</EffectiveFrom>
3558         </Access>
3559       </User>
3560     </Listing>
3561   </Access>
3562 </OrgAdminResponse>
```

3563 **6.3.1.4 Examples: user affiliation response**3564 *Example 77 User added successfully*

```
3565 <?xml version="1.0" encoding="UTF-8"?>
3566 <OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
3567   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3568   xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
3569   status="OK">
3570   <Organization>
3571     <MessagingAddress value="urn:org:level7.healthcare.inc"/>
3572   </Organization>
3573   <Users>
3574     <Added>
3575       <UserName>speters</UserName>
3576     </Added>
3577   </Users>
3578 </OrgAdminResponse>
```

3579 *Example 78 User listing*

```
3580 <?xml version="1.0" encoding="UTF-8"?>
3581 <OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1">
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
status="OK">
  <Organization>
    <MessagingAddress value="urn:org:level7.healthcare.inc"/>
  </Organization>
  <Users>
    <Listing>
      <User>
        <UserName>mjones</UserName>
        <FullName>M. Jones</FullName>
        <Role value="Provider" effectiveFrom="2004-06-23T20:00:00"/>
        <Role value="OrgAdministrator" effectiveFrom="2004-06-
23T20:00:00"/>
      </User>
      <User>
        <UserName>speters</UserName>
        <FullName>Susan Peters</FullName>
        <Role value="Provider" effectiveFrom="2004-06-23T20:00:00"/>
      </User>
    </Listing>
  </Users>
</OrgAdminResponse>
```

6.3.1.5 Examples: providers response

Example 79 Provider added successfully

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
  status="OK">
  <Organization>
    <MessagingAddress value="urn:org:level7.healthcare.inc"/>
  </Organization>
  <Providers>
    <Added>
      <ProviderID>urn:uuid:6F40BC6C-6855-480a-A335-
B83CD77DC3EE</ProviderID>
    </Added>
  </Providers>
</OrgAdminResponse>
```

Example 80 Provider listing

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
  status="OK">
  <Organization>
    <MessagingAddress value="urn:org:level7.healthcare.inc"/>
  </Organization>
  <Providers>
    <Listing>
      <Provider>
        <ProviderID>urn:uuid:24739DFB-F34E-44F9-A556-
3971D6408A30</ProviderID>
        <MessagingAddress displayName="Dr. Peter Plumber"
value="urn:provider:level7.healthcare.inc:pplumber"/>
        <EffectiveFrom>2004-05-29T22:30:00.00000</EffectiveFrom>
      </Provider>
      <Provider>
        <ProviderID>urn:uuid:24739DFB-A00F-44F9-A556-
3971D6408B00</ProviderID>
        <MessagingAddress displayName="Dr. Patricia Primary"
value="urn:provider:level7.healthcare.inc:pprimary" />
        <EffectiveFrom>2004-06-30T12:00:00.00000</EffectiveFrom>
      </Provider>
    </Listing>
  </Providers>
</OrgAdminResponse>
```

```
</Listing>
</Providers>
</OrgAdminResponse>
```

6.3.1.6 Examples: preferences response

Example 81 View organization preferences

```
<?xml version="1.0" encoding="UTF-8"?>
<OrgAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
  status="OK">
  <Organization>
    <MessagingAddress value="urn:org:level7.healthcare.inc"/>
  </Organization>
  <Preferences>
    <View>
      <DeliveryNotification active="true">
        <Email>info@levelseven.com</Email>
      </DeliveryNotification>
      <MessageRetention days="60"/>
      <NotifySentUnread days="10"/>
    </View>
  </Preferences>
</OrgAdminResponse>
```

6.3.2 Provider Administrative Response

The `ProviderAdminResponse` is used to return a response to the `ProviderAdmin` request described in section 0. If an error occurred, it returns an `Error` element within the response. This response is a return from the synchronous request.

6.3.2.1 Definition

```
<xs:element name="ProviderAdminResponse" type="ProviderAdminResponse"/>

<xs:complexType name="ProviderAdminResponse">
  <xs:complexContent>
    <xs:extension base="ResponseBase">
      <xs:sequence>
        <xs:element name="Provider" type="Addressable"/>
        <xs:choice>
          <xs:element name="Access" type="AdminAccessResp"/>
          <xs:element name="Preferences" type="AdminPreferences"/>
          <xs:element name="MessagingAddress"
type="AddressUpdateResp"/>
          <xs:element name="Note" type="NoteUpdateResp"/>
        <xs:sequence>
          <xs:element name="Error" type="Error"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:choice>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
```

6.3.2.2 Errors

In addition to the errors possible in all responses, the following errors may be returned:

- NoSuchAddress (300)
- AddressInUse (301)

- NoSuchKeyIdentifier (420)
- ValidationError (400)

6.3.2.3 Examples

Example 82 Provider user access added

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <Provider>
    <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-E9B3A5EFCEDA</ProviderID>
    <MessagingAddress displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
  </Provider>
  <Access>
    <Added>
      <UserName>speters</UserName>
    </Added>
  </Access>
</ProviderAdminResponse>
```

Example 83 Provider user access removed

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <Provider>
    <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-E9B3A5EFCEDA</ProviderID>
    <MessagingAddress displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
  </Provider>
  <Access>
    <Removed>
      <UserName>speters</UserName>
    </Removed>
  </Access>
</ProviderAdminResponse>
```

Example 84 Provider user access listing

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../eMSEP/eMSEP.xsd"
  status="OK">
  <Provider>
    <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-E9B3A5EFCEDA</ProviderID>
    <MessagingAddress displayName="Dr. Patricia Primary"
      value="urn:provider:level7.healthcare.inc:pprimary"/>
  </Provider>
  <Access>
    <Listing>
      <User>
        <UserName>speters</UserName>
        <Access>
          <Rights value="Owner"/>
          <EffectiveFrom>2004-05-29T22:30:00.0000</EffectiveFrom>
        </Access>
      </User>
      <User>
        <UserName>mjones</UserName>
        <Access>
          <Rights value="Delegate"/>
          <EffectiveFrom>2004-05-29T22:30:00.0000</EffectiveFrom>
        </Access>
      </User>
    </Listing>
  </Access>
</ProviderAdminResponse>
```

```
3765         </User>
3766     </Listing>
3767 </Access>
3768 </ProviderAdminResponse>
```

3769 Example 85 Provider preferences view

```
3770 <?xml version="1.0" encoding="UTF-8"?>
3771 <ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
3772     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3773     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
3774     status="OK">
3775     <Provider>
3776         <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-E9B3A5EFCEDA</ProviderID>
3777         <MessagingAddress displayName="Dr. Patricia
3778 Primary="urn:provider:level7.healthcare.inc:pprimary"/>
3779     </Provider>
3780     <Preferences>
3781         <View>
3782             <NotificationEmail>speters@southcentral.ca</NotificationEmail>
3783             <ReadReceipt active="true"/>
3784             <NotReadNotification days="7"/>
3785             <MessageRetention days="45"/>
3786         </View>
3787     </Preferences>
3788 </ProviderAdminResponse>
```

3789 Example 86 Provider address updated

```
3790 <?xml version="1.0" encoding="UTF-8"?>
3791 <ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
3792     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3793     xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
3794     status="OK">
3795     <Provider>
3796         <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-E9B3A5EFCEDA</ProviderID>
3797         <MessagingAddress displayName="Dr. Patricia
3798 Primary="urn:provider:level7.healthcare.inc:pprimary"/>
3799     </Provider>
3800     <MessagingAddress updated="true"/>
3801 </ProviderAdminResponse>
```

3802 6.3.3 User Administrative Response

3803 The UserAdminResponse for is intended to list user access rights to provider and organization
3804 message boxes.

3805 6.3.3.1 Definition

```
3806 <xs:element name="UserAdminResponse" type="UserAdminResponse"/>
3807 <xs:complexType name="UserAdminResponse">
3808     <xs:complexContent>
3809         <xs:extension base="ResponseBase">
3810             <xs:choice>
3811                 <xs:element name="User" type="User"/>
3812                 <xs:sequence>
3813                     <xs:element name="Error" type="Error"
3814 maxOccurs="unbounded"/>
3815                 </xs:sequence>
3816             </xs:choice>
3817         </xs:extension>
3818     </xs:complexContent>
3819 </xs:complexType>
```

3820 6.3.3.2 Returns

3821 In addition to the errors possible in all responses, the following errors may be returned:

- 3822 • PermissionError (200)

- NoSuchKeyIdentifier (420)

6.3.3.3 Example

Example 87 User Access Listing

```
<?xml version="1.0" encoding="UTF-8"?>
<UserAdminResponse xmlns="http://www.e-ms.ca/Schema/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.e-ms.ca/Schema/v1 ../../eMSEP/eMSEP.xsd"
  status="OK">
  <User>
    <UserName>pplumber</UserName>
    <Access>
      <Provider>
        <MessagingAddress displayName="Dr. Pete Plumber"
          value="urn:provider:level7.healthcare.inc:pplumber"/>
        </Provider>
        <Rights value="Owner"/>
        <EffectiveFrom>2004-06-20T20:00:00</EffectiveFrom>
      </Access>
      <Access>
        <Provider>
          <MessagingAddress displayName="Dr. Pat Primary"
            value="urn:provider:level7.healthcare.inc:pprimary"/>
          </Provider>
          <Rights value="Delegate"/>
          <EffectiveFrom>2004-06-20T20:00:00</EffectiveFrom>
        </Access>
        <Access>
          <Provider>
            <MessagingAddress displayName="Sophie Scratch, MD"
              value="urn:provider:level7.healthcare.inc:sophie.scratch"/>
            </Provider>
            <Rights value="Delegate"/>
            <EffectiveFrom>2004-05-28T12:00:00</EffectiveFrom>
          </Access>
          <Access>
            <Organization>
              <MessagingAddress displayName="Level Seven Healthcare"
                value="urn:org:level7.healthcare.inc"/>
            </Organization>
            <Rights value="Delegate"/>
            <EffectiveFrom>2004-05-29T20:00:00</EffectiveFrom>
          </Access>
        </Access>
      </User>
    </UserAdminResponse>
```

6.4 Admin Data Types XML Schema

The common data types used in Administrative Services Interface are defined in `admin.xsd`, a WXS schema file that is included in the `eMSEP.xsd` schema file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1" xmlns="http://www.e-
ms.ca/Schema/v1"
  xmlns:emsep="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:include schemaLocation="../../dt/DataTypes.xsd"/>
  <!--
    e-MS Project: Administrative Schema for eMS Message Protocol.
    Author: Brad Head <mailto:bradhead@sierrasystems.com>
    Date: Jun 04, 2004

    $Revision: 12 $
```

```
3884 Copyright (c) 2004 by Vancouver Island Health Authority. All rights reserved.
3885
3886 -->
3887 <xs:complexType name="AdminAccess">
3888   <xs:choice>
3889     <xs:element name="Add">
3890       <xs:complexType>
3891         <xs:sequence>
3892           <xs:element name="UserName" type="UserName"/>
3893           <xs:element name="Rights" type="RightsElement"/>
3894           <xs:element name="EffectiveFrom" type="xs:dateTime"
3895 minOccurs="0"/>
3896           <xs:element name="EffectiveTo" type="xs:dateTime"
3897 minOccurs="0"/>
3898         </xs:sequence>
3899       </xs:complexType>
3900     </xs:element>
3901     <xs:element name="Remove">
3902       <xs:complexType>
3903         <xs:sequence>
3904           <xs:element name="UserName" type="UserName"/>
3905         </xs:sequence>
3906       </xs:complexType>
3907     </xs:element>
3908     <xs:element name="List">
3909       <xs:complexType>
3910         <xs:sequence>
3911           <xs:element name="Rights" type="RightsElement"
3912 minOccurs="0"/>
3913         </xs:sequence>
3914       </xs:complexType>
3915     </xs:element>
3916   </xs:choice>
3917 </xs:complexType>
3918 <xs:complexType name="AdminPreferences">
3919   <xs:choice>
3920     <xs:element name="Update">
3921       <xs:complexType>
3922         <xs:group ref="AdmPreferencesGrp"/>
3923       </xs:complexType>
3924     </xs:element>
3925     <xs:element name="View" type="EmptyElement"/>
3926   </xs:choice>
3927 </xs:complexType>
3928 <xs:complexType name="AdminPreferencesResp">
3929   <xs:choice>
3930     <xs:element name="Updated">
3931       <xs:complexType>
3932         <xs:group ref="AdmPreferencesGrp"/>
3933       </xs:complexType>
3934     </xs:element>
3935     <xs:element name="View">
3936       <xs:complexType>
3937         <xs:group ref="AdmPreferencesGrp"/>
3938       </xs:complexType>
3939     </xs:element>
3940   </xs:choice>
3941 </xs:complexType>
3942 <xs:simpleType name="NotificationDelivery">
3943   <xs:restriction base="xs:NMTOKEN">
3944     <xs:enumeration value="Email"/>
3945     <xs:enumeration value="NotificationMessage"/>
3946   </xs:restriction>
3947 </xs:simpleType>
3948 <xs:complexType name="NotificationSetting">
3949   <xs:attribute name="deliveryMethod" type="NotificationDelivery"
3950 use="required"/>
3951 </xs:complexType>
3952 <xs:group name="AdmPreferencesGrp">
3953   <xs:sequence>
```

```
3954         <xs:element name="NotificationEmail" minOccurs="0" type="Email"/>
3955         <xs:element name="ReadReceipt" minOccurs="0">
3956             <xs:complexType>
3957                 <xs:attribute name="active" type="xs:boolean"
3958 use="required"/>
3959             </xs:complexType>
3960         </xs:element>
3961         <xs:element name="NotReadNotification" minOccurs="0">
3962             <xs:complexType>
3963                 <xs:attribute name="days" type="xs:integer" use="required"/>
3964             </xs:complexType>
3965         </xs:element>
3966         <xs:element name="MessageRetention">
3967             <xs:complexType>
3968                 <xs:attribute name="days" type="xs:integer" use="required"/>
3969             </xs:complexType>
3970         </xs:element>
3971     </xs:sequence>
3972 </xs:group>
3973 <xs:complexType name="User.AccessDetail">
3974     <xs:sequence>
3975         <xs:element name="UserName" type="UserName"/>
3976         <xs:element name="FullName" type="xs:string" minOccurs="0"/>
3977         <xs:element name="Access" type="AccessRights" maxOccurs="unbounded"/>
3978     </xs:sequence>
3979 </xs:complexType>
3980 <xs:complexType name="AdminAccessResp">
3981     <xs:choice>
3982         <xs:element name="Added">
3983             <xs:complexType>
3984                 <xs:sequence>
3985                     <xs:element name="UserName" type="UserName"/>
3986                 </xs:sequence>
3987             </xs:complexType>
3988         </xs:element>
3989         <xs:element name="Removed">
3990             <xs:complexType>
3991                 <xs:sequence>
3992                     <xs:element name="UserName" type="UserName"/>
3993                 </xs:sequence>
3994             </xs:complexType>
3995         </xs:element>
3996         <xs:element name="Listing">
3997             <xs:complexType>
3998                 <xs:sequence>
3999                     <xs:element name="User" type="User.AccessDetail"
4000 minOccurs="0" maxOccurs="unbounded"/>
4001                 </xs:sequence>
4002             </xs:complexType>
4003         </xs:element>
4004     </xs:choice>
4005 </xs:complexType>
4006 <!-- Organization Administration -->
4007 <xs:complexType name="OrgAdminProviders">
4008     <xs:choice>
4009         <xs:element name="Add">
4010             <xs:complexType>
4011                 <xs:sequence>
4012                     <xs:element name="ProviderID" type="ProviderID"/>
4013                     <xs:element name="MessagingAddress"
4014 type="MessagingAddress.Provider"/>
4015                 </xs:sequence>
4016             </xs:complexType>
4017         </xs:element>
4018         <xs:element name="Remove">
4019             <xs:complexType>
4020                 <xs:choice>
4021                     <xs:element name="ProviderID" type="ProviderID"/>
4022                     <xs:element name="MessagingAddress"
4023 type="MessagingAddress.Provider"/>

```

```
4024         </xs:choice>
4025     </xs:complexType>
4026 </xs:element>
4027 <xs:element name="List" type="EmptyElement"/>
4028 </xs:choice>
4029 </xs:complexType>
4030 <xs:complexType name="OrgAdminDetails">
4031     <xs:choice>
4032         <xs:element name="Update" type="Organization"/>
4033     </xs:choice>
4034 </xs:complexType>
4035 <xs:complexType name="OrgAdminUsers">
4036     <xs:choice>
4037         <xs:element name="Add">
4038             <xs:complexType>
4039                 <xs:sequence>
4040                     <xs:element name="UserName" type="UserName"/>
4041                     <xs:element name="Role" type="Role"
4042 maxOccurs="unbounded"/>
4043                 </xs:sequence>
4044             </xs:complexType>
4045         </xs:element>
4046         <xs:element name="Remove">
4047             <xs:complexType>
4048                 <xs:sequence>
4049                     <xs:element name="UserName" type="UserName"/>
4050                     <xs:element name="Role" type="Role" minOccurs="0"
4051 maxOccurs="unbounded"/>
4052                 </xs:sequence>
4053             </xs:complexType>
4054         </xs:element>
4055         <xs:element name="List">
4056             <xs:complexType>
4057                 <xs:sequence>
4058                     <xs:element name="Role" type="Role" minOccurs="0"
4059 maxOccurs="unbounded"/>
4060                 </xs:sequence>
4061             </xs:complexType>
4062         </xs:element>
4063     </xs:choice>
4064 </xs:complexType>
4065 <xs:complexType name="OrgAdminResp.User">
4066     <xs:sequence>
4067         <xs:element name="UserName" type="UserName"/>
4068         <xs:element name="FullName" type="xs:string"/>
4069         <xs:element name="Role" type="Role" maxOccurs="unbounded"/>
4070     </xs:sequence>
4071 </xs:complexType>
4072 <xs:complexType name="OrgAdminUsersResp">
4073     <xs:choice>
4074         <xs:element name="Added">
4075             <xs:complexType>
4076                 <xs:sequence>
4077                     <xs:element name="UserName" type="UserName"/>
4078                 </xs:sequence>
4079             </xs:complexType>
4080         </xs:element>
4081         <xs:element name="Removed">
4082             <xs:complexType>
4083                 <xs:sequence>
4084                     <xs:element name="UserName" type="UserName"/>
4085                 </xs:sequence>
4086             </xs:complexType>
4087         </xs:element>
4088         <xs:element name="Listing">
4089             <xs:complexType>
4090                 <xs:sequence>
4091                     <xs:element name="User" type="OrgAdminResp.User"
4092 minOccurs="0" maxOccurs="unbounded"/>
4093                 </xs:sequence>
```

```
4094         </xs:complexType>
4095     </xs:element>
4096     </xs:choice>
4097 </xs:complexType>
4098 <!-- OrgAdmin Request Type -->
4099 <xs:complexType name="OrgAdmin">
4100     <xs:sequence>
4101         <xs:element name="Organization" type="Organization.Key"/>
4102         <xs:choice>
4103             <xs:element name="Access" type="AdminAccess"/>
4104             <xs:element name="Users" type="OrgAdminUsers"/>
4105             <xs:element name="Providers" type="OrgAdminProviders"/>
4106             <xs:element name="Preferences" type="AdminPreferences"/>
4107             <xs:element name="Details" type="OrgAdminDetails"/>
4108         </xs:choice>
4109     </xs:sequence>
4110 </xs:complexType>
4111 <xs:complexType name="OrgAdminDetailsResp">
4112     <xs:choice>
4113         <xs:element name="Updated" type="EmptyElement"/>
4114     </xs:choice>
4115 </xs:complexType>
4116 <!-- OrgAdmin Response -->
4117 <xs:complexType name="OrgAdminResponse">
4118     <xs:complexContent>
4119         <xs:extension base="ResponseBase">
4120             <xs:sequence>
4121                 <xs:element name="Organization" type="Organization.Key"/>
4122                 <xs:choice>
4123                     <xs:element name="Access" type="AdminAccessResp"/>
4124                     <xs:element name="Users" type="OrgAdminUsersResp"/>
4125                     <xs:element name="Providers" type="OrgAdminMembersResp"/>
4126                     <xs:element name="Preferences"
4127 type="AdminPreferencesResp"/>
4128                     <xs:element name="Details" type="OrgAdminDetailsResp"/>
4129                 </xs:sequence>
4130                 <xs:element name="Error" type="Error"/>
4131             </xs:sequence>
4132         </xs:choice>
4133     </xs:sequence>
4134 </xs:extension>
4135 </xs:complexContent>
4136 </xs:complexType>
4137 <xs:complexType name="Provider.Member">
4138     <xs:sequence>
4139         <xs:element name="ProviderID" type="ProviderID"/>
4140         <xs:element name="MessagingAddress"
4141 type="MessagingAddress.Provider"/>
4142         <xs:element name="EffectiveFrom" type="xs:dateTime" minOccurs="0"/>
4143     </xs:sequence>
4144 </xs:complexType>
4145 <xs:complexType name="OrgAdminMembersResp">
4146     <xs:choice>
4147         <xs:element name="Added">
4148             <xs:complexType>
4149                 <xs:sequence>
4150                     <xs:element name="ProviderID" type="ProviderID"/>
4151                     <xs:element name="MessagingAddress"
4152 type="MessagingAddress.Provider"/>
4153                 </xs:sequence>
4154             </xs:complexType>
4155         </xs:element>
4156         <xs:element name="Removed">
4157             <xs:complexType>
4158                 <xs:sequence>
4159                     <xs:element name="ProviderID" type="ProviderID"/>
4160                 </xs:sequence>
4161             </xs:complexType>
4162         </xs:element>
4163         <xs:element name="Listing">
```



```
4164         <xs:complexType>
4165             <xs:sequence>
4166                 <xs:element name="Provider" type="Provider.Member"
4167 minOccurs="0" maxOccurs="unbounded"/>
4168             </xs:sequence>
4169         </xs:complexType>
4170     </xs:element>
4171 </xs:choice>
4172 </xs:complexType>
4173 <!-- Provider Admin -->
4174 <xs:complexType name="ProviderAdmin">
4175     <xs:sequence>
4176         <xs:element name="Provider" type="Provider.MemberKey"/>
4177         <xs:choice>
4178             <xs:element name="Access" type="AdminAccess"/>
4179             <xs:element name="Preferences" type="AdminPreferences"/>
4180             <xs:element name="MessagingAddress"
4181 type="MessagingAddress.Provider"/>
4182             <xs:element name="Note" type="xs:string"/>
4183         </xs:choice>
4184     </xs:sequence>
4185 </xs:complexType>
4186 <!-- Provider Admin Response -->
4187 <xs:complexType name="ProviderAdminResponse">
4188     <xs:complexContent>
4189         <xs:extension base="ResponseBase">
4190             <xs:sequence>
4191                 <xs:element name="Provider" type="Provider.Member"/>
4192                 <xs:choice>
4193                     <xs:element name="Access" type="AdminAccessResp"/>
4194                     <xs:element name="Preferences"
4195 type="AdminPreferencesResp"/>
4196                     <xs:element name="MessagingAddress"
4197 type="AddressUpdateResp"/>
4198                     <xs:element name="Note" type="NoteUpdateResp"/>
4199                 </xs:choice>
4200                 <xs:sequence>
4201                     <xs:element name="Error" type="Error"/>
4202                 </xs:sequence>
4203             </xs:choice>
4204         </xs:extension>
4205     </xs:complexContent>
4206 </xs:complexType>
4207 <xs:complexType name="AddressUpdateResp">
4208     <xs:attribute name="updated" type="xs:boolean" default="true"/>
4209 </xs:complexType>
4210 <xs:complexType name="NoteUpdateResp">
4211     <xs:attribute name="updated" type="xs:boolean" default="true"/>
4212 </xs:complexType>
4213 <xs:complexType name="UserAdmin">
4214     <xs:sequence>
4215         <xs:element name="UserName" type="UserName"/>
4216         <xs:choice>
4217             <xs:element name="ListAccess">
4218                 <xs:complexType>
4219                     <xs:attribute name="list" fixed="true"/>
4220                 </xs:complexType>
4221             </xs:element>
4222         </xs:choice>
4223     </xs:sequence>
4224 </xs:complexType>
4225 <xs:complexType name="UserAdminResponse">
4226     <xs:complexContent>
4227         <xs:extension base="ResponseBase">
4228             <xs:choice>
4229                 <xs:element name="User" type="User"/>
4230                 <xs:sequence>
4231                     <xs:element name="Error" type="Error"/>
4232                 </xs:sequence>
4233             </xs:choice>
```

4234
4235
4236
4237

```
</xs:extension>  
</xs:complexContent>  
</xs:complexType>  
</xs:schema>
```

4238

7 Error Messages

7.1 Overview

For each request schema previously described, the corresponding response may return one or more errors. The status attribute of the response element will indicate a status of `ERROR`. The normal status for a response is `OK`.

The `Error` element is contained within the root response element. There may be more than one error returned.

7.2 Error Codes

Code	Name	Description
100	NoSuchMessage	This error occurs when the message ID does not exist. It may be an invalid ID, or the message may no longer exist.
101	MessageAlreadyRead	This warning is returned when a message is requested to be marked as read when it had already been marked as read.
102	MessageAlreadyAked	This warning is returned when a message is requested to be marked as received, but it already is marked as received.
103	NoMessageBoxes	This error occurs when the user authenticated does not have access to any message boxes on the server.
200	PermissionError	This error occurs when a user does not have the correct role or access rights to perform the request.
103	UnsupportedMessage	The message received is not a supported message.
300	NoSuchAddress	The messaging address is not valid. It may be incorrect or does not exist.
301	AddressInUse	This is an error to indicate that the address being updated is already in use.
302	NoDataFound	The request found no data, or no data was affected.
400	ValidationError	The message XML is not valid against the schema or the ClinicalDocument sent does not pass validation or CDA specification conformance.
401	NotWellFormedError	The message elements were not in the

		correct order according to the schema, or required elements were missing. The XML is not wellformed.
420	NoSuchKeyIdentifier	This is a generalized error to indicate that a key value passed was not found.
430	TooManyResults	This error is returned for a query that would return too many results.
500	ApplicationError	A catchall for internal and system errors not covered by all other error codes.

4247

8 XML Schemas

8.1 Main eMSEP XML Schema – eMEP.xsd

The public elements of the eMSEP service definitions are defined in eMSEP.xsd.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:hl7="urn:hl7-org:v3" xmlns="http://www.e-ms.ca/Schema/v1"
  elementFormDefault="qualified">
  <xs:include schemaLocation="admin/admin.xsd" />
  <xs:include schemaLocation="query/query.xsd" />
  <xs:include schemaLocation="messaging/messaging.xsd" />
  <!--
  e-MS Project: eMSEP Schema for eMS Message Protocol.
  This version is what is exposed as SOAP Messages. It does not have
  constrained payload. The payload is not restricted to be
  validated eMS Clinical Document.

  Author: Brad Head <mailto:bradhead@sierrasystems.com>
  Date: Jun 04, 2004

  $Revision: 30 $

  Copyright (c) 2004 by Vancouver Island Health Authority. All rights reserved.
  -->
  <xs:element name="OrgAdmin" type="OrgAdmin" />
  <xs:element name="OrgAdminResponse" type="OrgAdminResponse"/>
  <xs:element name="ProviderAdmin" type="ProviderAdmin" />
  <xs:element name="ProviderAdminResponse" type="ProviderAdminResponse"/>
  <xs:element name="UserAdmin" type="UserAdmin" />
  <xs:element name="UserAdminResponse" type="UserAdminResponse"/>
  <xs:element name="MessageAck" type="MessageAck" />
  <xs:element name="MessageAckResponse" type="MessageAckResponse"/>
  <xs:element name="MessageBoxStatus" type="MessageBoxStatus" />
  <xs:element name="MessageBoxStatusResponse" type="MessageBoxStatusResponse"/>
  <xs:element name="MessageListing" type="MessageListing" />
  <xs:element name="MessageListingResponse" type="MessageListingResponse"/>
  <xs:element name="MessageRead" type="MessageRead" />
  <xs:element name="MessageReadResponse" type="MessageReadResponse"/>
  <xs:element name="MessageRecv" type="MessageRecv" />
  <xs:element name="MessageRecvResponse" type="MessageRecvResponse.Any"/>
  <xs:element name="MessageReset" type="MessageReset" />
  <xs:element name="MessageResetResponse" type="MessageResetResponse"/>
  <xs:element name="MessageSend" type="MessageSend.Any" />
  <xs:element name="MessageSendResponse" type="MessageSendResponse"/>
  <xs:element name="MessageTrack" type="MessageTrack" />
  <xs:element name="MessageTrackResponse" type="MessageTrackResponse"/>
  <xs:element name="OrgQuery" type="OrganizationQuery" />
  <xs:element name="OrgQueryResponse" type="OrgQueryResponse"/>
  <xs:element name="ProviderQuery" type="ProviderQuery" />
  <xs:element name="ProviderQueryResponse" type="ProviderQueryResponse"/>
  <xs:element name="UserQuery" type="UserQuery" />
  <xs:element name="UserQueryResponse" type="UserQueryResponse"/>
</xs:schema>
```

8.2 Data Type Schemas – DataTypes.xsd

The common data types by the eMSEP service definitions are defined in dt/DataTypes.xsd, a WXS schema file that is included in the eMSEP.xsd schema file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1" xmlns="http://www.e-
ms.ca/Schema/v1"
  xmlns:emsep="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:hl7="urn:hl7-org:v3" elementFormDefault="qualified">
  <xs:import namespace="urn:hl7-org:v3"
  schemaLocation="../../eMS_CDA/dt/datatypes.xsd"/>
  <!--
  e-MS Project: DataTypes Schema for eMS Message Protocol.
  Author: Brad Head <mailto:bradhead@sierrasystems.com>
  Date: Jun 04, 2004

  $Revision: 10 $

  Copyright (c) 2004 by Vancouver Island Health Authority. All rights reserved.
  -->
  <xs:complexType name="Country" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="State" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="City" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PostalCode" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="HouseNumber" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="StreetName" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="StreetNameType" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="StreetAddressLine" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="hl7:ADXP"/>
    </xs:complexContent>
  </xs:complexType>
  <!-- Postal Address -->
  <xs:complexType name="PostalAddress" mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="country" type="Country" minOccurs="0"/>
      <xs:element name="state" type="State" minOccurs="0"/>
      <xs:element name="city" type="City" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>
```

```
4372         <xs:element name="postalCode" type="PostalCode" minOccurs="0"/>
4373         <xs:element name="houseNumber" type="HouseNumber" minOccurs="0"/>
4374         <xs:element name="streetName" type="StreetName" minOccurs="0"/>
4375         <xs:element name="streetNameType" type="StreetNameType"
4376 minOccurs="0"/>
4377         <xs:element name="streetAddressLine" type="StreetAddressLine"
4378 minOccurs="0" maxOccurs="4"/>
4379     </xs:choice>
4380     <xs:attribute name="use" type="hl7:set_cs_PostalAddressUse"
4381 use="optional"/>
4382 </xs:complexType>
4383 <xs:complexType name="Telecom">
4384     <xs:complexContent>
4385         <xs:extension base="hl7:TEL"/>
4386     </xs:complexContent>
4387 </xs:complexType>
4388 <xs:complexType name="PersonName" mixed="true">
4389     <xs:complexContent mixed="true">
4390         <xs:extension base="hl7:PN"/>
4391     </xs:complexContent>
4392 </xs:complexType>
4393 <xs:complexType name="AdministrativeGenderCode">
4394     <xs:complexContent>
4395         <xs:extension base="hl7:CE"/>
4396     </xs:complexContent>
4397 </xs:complexType>
4398 <xs:complexType name="OrgName" mixed="true">
4399     <xs:complexContent mixed="true">
4400         <xs:extension base="hl7:ON"/>
4401     </xs:complexContent>
4402 </xs:complexType>
4403 <xs:group name="OrgDetailGrp">
4404     <xs:sequence>
4405         <xs:element name="MessagingAddress"
4406 type="MessagingAddress.Organization" minOccurs="0"/>
4407         <xs:element name="Name" type="OrgName" minOccurs="0" maxOccurs="2"/>
4408         <xs:element name="Telecom" type="Telecom" minOccurs="0"
4409 maxOccurs="unbounded"/>
4410         <xs:element name="PostalAddress" type="PostalAddress" minOccurs="0"/>
4411         <xs:element name="Note" type="xs:string" minOccurs="0"/>
4412     </xs:sequence>
4413 </xs:group>
4414 <xs:complexType name="Organization">
4415     <xs:sequence>
4416         <xs:group ref="OrgDetailGrp"/>
4417     </xs:sequence>
4418 </xs:complexType>
4419 <!-- Data Types -->
4420 <xs:complexType name="EmptyElement" final="extension">
4421     <xs:annotation>
4422         <xs:documentation xml:lang="en"> An empty element. Content must be
4423 empty. This is used
4424         to define a non empty type that does not contain
4425 anything</xs:documentation>
4426     </xs:annotation>
4427     <xs:attribute name="a" use="optional"/>
4428 </xs:complexType>
4429 <xs:complexType name="Coded">
4430     <xs:attribute name="code" type="xs:token" use="optional">
4431         <xs:annotation>
4432             <xs:documentation> The plain code symbol defined by the code
4433 system. For example,
4434             "784.0" is the code symbol of the ICD-9 code "784.0" for
4435 headache.</xs:documentation>
4436         </xs:annotation>
4437     </xs:attribute>
4438     <xs:attribute name="codeSystem" type="xs:string" use="optional">
4439         <xs:annotation>
4440             <xs:documentation> Specifies the code system that defines the
4441 code.</xs:documentation>
```



```
4442     </xs:annotation>
4443   </xs:attribute>
4444   <xs:attribute name="codeSystemName" type="xs:string" use="optional">
4445     <xs:annotation>
4446       <xs:documentation> A common name of the coding
4447 system.</xs:documentation>
4448     </xs:annotation>
4449   </xs:attribute>
4450   <xs:attribute name="codeSystemVersion" type="xs:string" use="optional">
4451     <xs:annotation>
4452       <xs:documentation> If applicable, a version descriptor defined
4453 specifically for the
4454       given code system.</xs:documentation>
4455     </xs:annotation>
4456   </xs:attribute>
4457   <xs:attribute name="displayName" type="xs:string" use="optional">
4458     <xs:annotation>
4459       <xs:documentation> A name or title for the code, under which the
4460 sending system
4461       shows the code value to its users. </xs:documentation>
4462     </xs:annotation>
4463   </xs:attribute>
4464 </xs:complexType>
4465 <xs:simpleType name="URN">
4466   <xs:restriction base="xs:anyURI">
4467     <xs:pattern value="urn:.*"/>
4468   </xs:restriction>
4469 </xs:simpleType>
4470 <xs:simpleType name="URN.UUID">
4471   <xs:annotation>
4472     <xs:documentation>uuid elements are string characters with a maximum
4473 length of 36.</xs:documentation>
4474   </xs:annotation>
4475   <xs:restriction base="URN">
4476     <xs:maxLength value="45"/>
4477     <xs:pattern value="urn:uuid:([0-9]|[a-f]|[A-F]){8}-([0-9]|[a-f]|[A-
4478 F]){4}-([0-9]|[a-f]|[A-F]){4})-([0-9]|[a-f]|[A-F]){4})-([0-9]|[a-f]|[A-
4479 F]){12})"/>
4480   </xs:restriction>
4481 </xs:simpleType>
4482 <xs:simpleType name="URN.Provider">
4483   <xs:annotation>
4484     <xs:documentation>URN for an addressable Provider.</xs:documentation>
4485   </xs:annotation>
4486   <xs:restriction base="URN">
4487     <xs:maxLength value="255"/>
4488     <xs:pattern value="urn:provider:.*"/>
4489   </xs:restriction>
4490 </xs:simpleType>
4491 <xs:simpleType name="URN.Organization">
4492   <xs:annotation>
4493     <xs:documentation>URN for an addressable Provider.</xs:documentation>
4494   </xs:annotation>
4495   <xs:restriction base="URN">
4496     <xs:maxLength value="255"/>
4497     <xs:pattern value="urn:org:.*"/>
4498   </xs:restriction>
4499 </xs:simpleType>
4500 <xs:simpleType name="URN.MessagingAddress">
4501   <xs:union memberTypes=" URN.Provider URN.Organization"/>
4502 </xs:simpleType>
4503 <xs:complexType name="MessagingAddress">
4504   <xs:attribute name="displayName" type="xs:string" use="optional"/>
4505   <xs:attribute name="value" type="URN.MessagingAddress" use="required"/>
4506 </xs:complexType>
4507 <xs:complexType name="MessagingAddress.Provider" final="#all">
4508   <xs:attribute name="displayName" type="xs:string"
4509 use="optional"/>
4510   <xs:attribute name="value" type="URN.Provider" use="required"/>
4511 </xs:complexType>
```

```
4512 <xs:complexType name="MessagingAddress.Organization" final="#all">
4513     <xs:attribute name="displayName" type="xs:string"
4514 use="optional"/>
4515     <xs:attribute name="value" type="URN.Organization"
4516 use="required"/>
4517 </xs:complexType>
4518 <xs:simpleType name="ProviderID">
4519     <xs:restriction base="URN.UUID"/>
4520 </xs:simpleType>
4521 <xs:complexType name="ProviderSpecialty" mixed="true">
4522     <xs:attribute name="sub" type="xs:boolean" use="optional"
4523 default="false"/>
4524 </xs:complexType>
4525 <xs:simpleType name="UserName">
4526     <xs:restriction base="xs:token">
4527         <xs:minLength value="1"/>
4528     </xs:restriction>
4529 </xs:simpleType>
4530 <xs:simpleType name="AccessControl">
4531     <xs:restriction base="xs:NMTOKEN">
4532         <xs:enumeration value="Delegate"/>
4533         <xs:enumeration value="Owner"/>
4534     </xs:restriction>
4535 </xs:simpleType>
4536 <xs:simpleType name="UserRole">
4537     <xs:restriction base="xs:NMTOKEN">
4538         <xs:enumeration value="MOA"/>
4539         <xs:enumeration value="EMR"/>
4540         <xs:enumeration value="OrgAdministrator"/>
4541         <xs:enumeration value="Provider"/>
4542     </xs:restriction>
4543 </xs:simpleType>
4544 <xs:complexType name="Role">
4545     <xs:attribute name="value" type="UserRole" use="required"/>
4546     <xs:attribute name="effectiveFrom" type="xs:dateTime"/>
4547 </xs:complexType>
4548 <xs:simpleType name="Email">
4549     <xs:restriction base="xs:token">
4550         <xs:pattern value="([\.a-zA-Z0-9_\-])+@([\.a-zA-Z0-9_\-])+(([\.a-zA-Z0-9_\-])*\.[a-zA-Z0-9_\-])+"/>
4551     </xs:restriction>
4552 </xs:simpleType>
4553 <xs:simpleType name="StatusType">
4554     <xs:restriction base="xs:token">
4555         <xs:enumeration value="OK"/>
4556         <xs:enumeration value="ERROR"/>
4557     </xs:restriction>
4558 </xs:simpleType>
4559 <xs:complexType name="ResponseBase">
4560     <xs:attribute name="status" use="required" type="StatusType"/>
4561 </xs:complexType>
4562 <xs:simpleType name="BillingNumber">
4563     <xs:restriction base="xs:string"/>
4564 </xs:simpleType>
4565 <xs:complexType name="Organization.Key">
4566     <xs:sequence>
4567         <xs:element name="MessagingAddress"
4568 type="MessagingAddress.Organization"/>
4569     </xs:sequence>
4570 </xs:complexType>
4571 <xs:complexType name="Provider.MemberKey">
4572     <xs:choice>
4573         <xs:element name="MessagingAddress"
4574 type="MessagingAddress.Provider"/>
4575     </xs:choice>
4576 </xs:complexType>
4577 <xs:complexType name="User.Affiliation">
4578     <xs:sequence>
4579         <xs:element name="Role" type="Role" maxOccurs="unbounded"/>
4580         <xs:element name="Organization" type="Organization"/>
4581
```

```
4582         </xs:sequence>
4583     </xs:complexType>
4584     <xs:complexType name="RightsElement">
4585         <xs:attribute name="value" type="AccessControl" use="required"/>
4586     </xs:complexType>
4587     <xs:group name="AccessRightsGrp">
4588         <xs:sequence>
4589             <xs:element name="Rights" type="RightsElement"/>
4590             <xs:element name="EffectiveFrom" type="xs:dateTime"/>
4591             <xs:element name="EffectiveTo" type="xs:dateTime" nillable="true"
4592 minOccurs="0"/>
4593         </xs:sequence>
4594     </xs:group>
4595     <xs:complexType name="AccessRights">
4596         <xs:group ref="AccessRightsGrp"/>
4597     </xs:complexType>
4598     <xs:complexType name="User.AccessRights">
4599         <xs:sequence>
4600             <xs:choice>
4601                 <xs:element name="Provider" type="Provider.MemberKey"/>
4602                 <xs:element name="Organization" type="Organization.Key"/>
4603             </xs:choice>
4604             <xs:group ref="AccessRightsGrp"/>
4605         </xs:sequence>
4606     </xs:complexType>
4607     <xs:complexType name="User">
4608         <xs:sequence>
4609             <xs:element name="UserName" type="UserName"/>
4610             <xs:element name="FullName" type="xs:string" minOccurs="0"/>
4611             <xs:element name="Affiliation" type="User.Affiliation" minOccurs="0"
4612 maxOccurs="unbounded"/>
4613             <xs:element name="Access" type="User.AccessRights" minOccurs="0"
4614 maxOccurs="unbounded"/>
4615         </xs:sequence>
4616     </xs:complexType>
4617     <xs:complexType name="Error" mixed="true">
4618         <xs:sequence>
4619             <xs:element name="ErrorMsg" type="xs:string"/>
4620             <xs:element name="MessageID" type="URN.UUID" minOccurs="0"/>
4621         </xs:sequence>
4622         <xs:attribute name="errorCode" type="xs:integer" use="required"/>
4623         <xs:attribute name="dateTime" type="xs:dateTime" use="optional"/>
4624     </xs:complexType>
4625 </xs:schema>
```

9 SOAP Bindings – Web Services

9.1 Web Service definitions for Exchange Services Protocol

The e-MS Web Services are described using a Web Services Description Language (WSDL) document. A WSDL document is the standard mechanism used to describe the capabilities of a service. The e-MS WSDL files are described in this section. The WSDL location URI and the Binding URI web service addresses are not yet finalized, and depend upon the location of the deployment of this set of services. The **placeholders**, indicated in **bold** in the examples must be substituted with actual values.

The WSDL has been verified using tools published by the Web Services Interoperability Organization as conforming to the WS-I Basic Profile. This conformance test ensures that a higher degree of interoperability is achieved beyond what can be expressed using W3C WSDL specification.

***Note(1):** It is possible that an instance of an e-MS service request to transmit a CDA document successfully validates against the WSDL but its request XML does not validate against the XSD for the specific MessageSend that expects a valid e-MS. This is because the WSDL publishes an unrestricted MessageSend that allows any content (<xs:any/>), whereas e-MS conformance requires that the payload be a valid e-MS CDA document.*

9.1.1 e-MS Web Service Namespace

```
http://www.e-ms.ca/Schema/v1/WS
```

9.2 SOAP Request/Response Example

Example 88 SOAP Request

```
POST /eMSAdminWS/eMSAdminWS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.e-ms.ca/Schema/v1/WS/eMSAdminWS/ProviderAdmin"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1/WS">
      <ProviderAdmin xmlns="http://www.e-ms.ca/Schema/v1">
        <Provider>
          <MessagingAddress value="urn:provider:speters" />
        </Provider>
        <Access>
          <Add>
            <UserName>speters</UserName>
            <Rights value="Owner"/>
          </Add>
        </Access>
      </ProviderAdmin>
```

```
4675     </ProviderAdmin>
4676   </soap:Body>
4677 </soap:Envelope>
4678
```

4679 Example 89 SOAP Response

```
4680 HTTP/1.1 200 OK
4681 Content-Type: text/xml; charset=utf-8
4682 Content-Length: length
4683
4684 <?xml version="1.0" encoding="utf-8"?>
4685 <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4686   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4687   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
4688   <soap:Body>
4689     <ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1/WS">
4690       <ProviderAdminResponse xmlns="http://www.e-ms.ca/Schema/v1">
4691         <Provider>
4692           <ProviderID>urn:uuid:0FB220FF-9D91-43FB-8D77-
4693 E9B3A5EFCEDA</ProviderID>
4694           <MessagingAddress displayName="Dr. Patricia Primary"
4695 value="urn:provider:level7.healthcare.inc:pprimary"/>
4696         </Provider>
4697         <Access>
4698           <Added>
4699             <UserName>pprimary</UserName>
4700           </Added>
4701         </Access>
4702       </ProviderAdminResponse>
4703     </ProviderAdminResponse>
4704   </soap:Body>
4705 </soap:Envelope>
4706
```

4707 9.3 eMSEP Web Services (eMSEP)

4708 9.3.1 Web Services URL

```
4709 https://localhost/eMSEP/eMSEP.asmx
4710
4711
```

4712 9.3.2 Web Service Methods

Web Service Method	SOAP Action
MessageAck	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageAck
MessageBoxStatus	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageBoxStatus
MesssageListing	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageListing
MessageRead	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageRead
MessageRecv	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageRecv
MessageReset	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageReset
MessageSend	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageSend
MessageTrack	http://www.e-ms.ca/Schema/v1/WS/eMSEP/MessageTrack

OrgQuery	http://www.e-ms.ca/Schema/v1/WS/eMSEP/OrgQuery
ProviderQuery	http://www.e-ms.ca/Schema/v1/WS/eMSEP/ProviderQuery
UserQuery	http://www.e-ms.ca/Schema/v1/WS/eMSEP/UserQuery
OrgAdmin	http://www.e-ms.ca/Schema/v1/WS/eMSEP/OrgAdmin
ProviderAdmin	http://www.e-ms.ca/Schema/v1/WS/eMSEP/ProviderAdmin
UserAdmin	http://www.e-ms.ca/Schema/v1/WS/eMSEP/UserAdmin

9.3.3 URL to view the WSDL

The eMSAdminWS WSDL may be viewed at the following URL (replace **localhost** with the host of the e-MS service):

```
http://localhost/eMSEP/eMSEP.wsdl
```

9.4 eMSEP Web Service Description (WSDL)

9.4.1 eMSEP.wsdl

```
<?xml version="1.0" encoding="utf-8"?>
<wSDL:definitions
  xsi:schemaLocation="http://schemas.xmlsoap.org/wsdl/
http://schemas.xmlsoap.org/wsdl/"
  name="eMSEP" targetNamespace="http://www.e-ms.ca/Schema/v1/WS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:h17="urn:h17-
org:v3"
  xmlns:emsep="http://www.e-ms.ca/Schema/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:ws="http://www.e-ms.ca/Schema/v1/WS"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:b="http://schemas.microsoft.com/BizTalk/2003">
  <wSDL:types>
    <xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1/WS"
      elementFormDefault="qualified">
      <xs:import namespace="http://www.e-ms.ca/Schema/v1"/>
      <xs:element name="MessageAck">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1"
ref="emsep:MessageAck"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="MessageAckResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1"
ref="emsep:MessageAckResponse"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="MessageBoxStatus">
        <xs:complexType>
```

```
4761         <xs:sequence>
4762             <xs:element minOccurs="1" maxOccurs="1"
4763 ref="emsep:MessageBoxStatus"/>
4764         </xs:sequence>
4765     </xs:complexType>
4766 </xs:element>
4767 <xs:element name="MessageBoxStatusResponse">
4768     <xs:complexType>
4769         <xs:sequence>
4770             <xs:element minOccurs="1" maxOccurs="1"
4771 ref="emsep:MessageBoxStatusResponse"/>
4772         </xs:sequence>
4773     </xs:complexType>
4774 </xs:element>
4775 <xs:element name="MessageListing">
4776     <xs:complexType>
4777         <xs:sequence>
4778             <xs:element minOccurs="1" maxOccurs="1"
4779 ref="emsep:MessageListing"/>
4780         </xs:sequence>
4781     </xs:complexType>
4782 </xs:element>
4783 <xs:element name="MessageListingResponse">
4784     <xs:complexType>
4785         <xs:sequence>
4786             <xs:element minOccurs="1" maxOccurs="1"
4787 ref="emsep:MessageListingResponse"/>
4788         </xs:sequence>
4789     </xs:complexType>
4790 </xs:element>
4791 <xs:element name="MessageRead">
4792     <xs:complexType>
4793         <xs:sequence>
4794             <xs:element minOccurs="1" maxOccurs="1"
4795 ref="emsep:MessageRead"/>
4796         </xs:sequence>
4797     </xs:complexType>
4798 </xs:element>
4799 <xs:element name="MessageReadResponse">
4800     <xs:complexType>
4801         <xs:sequence>
4802             <xs:element minOccurs="1" maxOccurs="1"
4803 ref="emsep:MessageReadResponse"/>
4804         </xs:sequence>
4805     </xs:complexType>
4806 </xs:element>
4807 <xs:element name="MessageRecv">
4808     <xs:complexType>
4809         <xs:sequence>
4810             <xs:element minOccurs="1" maxOccurs="1"
4811 ref="emsep:MessageRecv"/>
4812         </xs:sequence>
4813     </xs:complexType>
4814 </xs:element>
4815 <xs:element name="MessageRecvResponse">
4816     <xs:complexType>
4817         <xs:sequence>
4818             <xs:element minOccurs="1" maxOccurs="1"
4819 ref="emsep:MessageRecvResponse"/>
4820         </xs:sequence>
4821     </xs:complexType>
4822 </xs:element>
4823 <xs:element name="MessageReset">
4824     <xs:complexType>
4825         <xs:sequence>
4826             <xs:element minOccurs="1" maxOccurs="1"
4827 ref="emsep:MessageReset"/>
4828         </xs:sequence>
4829     </xs:complexType>
4830 </xs:complexType>
```



```
4831         </xs:element>
4832         <xs:element name="MessageResetResponse">
4833             <xs:complexType>
4834                 <xs:sequence>
4835                     <xs:element minOccurs="1" maxOccurs="1"
4836 ref="emsep:MessageResetResponse"/>
4837                 </xs:sequence>
4838             </xs:complexType>
4839         </xs:element>
4840         <xs:element name="MessageSend">
4841             <xs:complexType>
4842                 <xs:sequence>
4843                     <xs:element minOccurs="1" maxOccurs="1"
4844 ref="emsep:MessageSend"/>
4845                 </xs:sequence>
4846             </xs:complexType>
4847         </xs:element>
4848         <xs:element name="MessageSendResponse">
4849             <xs:complexType>
4850                 <xs:sequence>
4851                     <xs:element minOccurs="1" maxOccurs="1"
4852 ref="emsep:MessageSendResponse"/>
4853                 </xs:sequence>
4854             </xs:complexType>
4855         </xs:element>
4856         <xs:element name="MessageTrack">
4857             <xs:complexType>
4858                 <xs:sequence>
4859                     <xs:element minOccurs="1" maxOccurs="1"
4860 ref="emsep:MessageTrack"/>
4861                 </xs:sequence>
4862             </xs:complexType>
4863         </xs:element>
4864         <xs:element name="MessageTrackResponse">
4865             <xs:complexType>
4866                 <xs:sequence>
4867                     <xs:element minOccurs="1" maxOccurs="1"
4868 ref="emsep:MessageTrackResponse"/>
4869                 </xs:sequence>
4870             </xs:complexType>
4871         </xs:element>
4872         <xs:element name="OrgAdmin">
4873             <xs:complexType>
4874                 <xs:sequence>
4875                     <xs:element minOccurs="1" maxOccurs="1"
4876 ref="emsep:OrgAdmin"/>
4877                 </xs:sequence>
4878             </xs:complexType>
4879         </xs:element>
4880         <xs:element name="OrgAdminResponse">
4881             <xs:complexType>
4882                 <xs:sequence>
4883                     <xs:element minOccurs="1" maxOccurs="1"
4884 ref="emsep:OrgAdminResponse"/>
4885                 </xs:sequence>
4886             </xs:complexType>
4887         </xs:element>
4888         <xs:element name="OrgQuery">
4889             <xs:complexType>
4890                 <xs:sequence>
4891                     <xs:element minOccurs="1" maxOccurs="1"
4892 ref="emsep:OrgQuery"/>
4893                 </xs:sequence>
4894             </xs:complexType>
4895         </xs:element>
4896         <xs:element name="OrgQueryResponse">
4897             <xs:complexType>
4898                 <xs:sequence>
4899                     <xs:element minOccurs="1" maxOccurs="1"
4900 ref="emsep:OrgQueryResponse"/>
```

```
4901         </xs:sequence>
4902     </xs:complexType>
4903 </xs:element>
4904 <xs:element name="ProviderAdmin">
4905     <xs:complexType>
4906         <xs:sequence>
4907             <xs:element minOccurs="1" maxOccurs="1"
4908 ref="emsep:ProviderAdmin"/>
4909         </xs:sequence>
4910     </xs:complexType>
4911 </xs:element>
4912 <xs:element name="ProviderAdminResponse">
4913     <xs:complexType>
4914         <xs:sequence>
4915             <xs:element minOccurs="1" maxOccurs="1"
4916 ref="emsep:ProviderAdminResponse"/>
4917         </xs:sequence>
4918     </xs:complexType>
4919 </xs:element>
4920 <xs:element name="ProviderQuery">
4921     <xs:complexType>
4922         <xs:sequence>
4923             <xs:element minOccurs="1" maxOccurs="1"
4924 ref="emsep:ProviderQuery"/>
4925         </xs:sequence>
4926     </xs:complexType>
4927 </xs:element>
4928 <xs:element name="ProviderQueryResponse">
4929     <xs:complexType>
4930         <xs:sequence>
4931             <xs:element minOccurs="1" maxOccurs="1"
4932 ref="emsep:ProviderQueryResponse"/>
4933         </xs:sequence>
4934     </xs:complexType>
4935 </xs:element>
4936 <xs:element name="UserAdmin">
4937     <xs:complexType>
4938         <xs:sequence>
4939             <xs:element minOccurs="1" maxOccurs="1"
4940 ref="emsep:UserAdmin"/>
4941         </xs:sequence>
4942     </xs:complexType>
4943 </xs:element>
4944 <xs:element name="UserAdminResponse">
4945     <xs:complexType>
4946         <xs:sequence>
4947             <xs:element minOccurs="1" maxOccurs="1"
4948 ref="emsep:UserAdminResponse"/>
4949         </xs:sequence>
4950     </xs:complexType>
4951 </xs:element>
4952 <xs:element name="UserQuery">
4953     <xs:complexType>
4954         <xs:sequence>
4955             <xs:element minOccurs="1" maxOccurs="1"
4956 ref="emsep:UserQuery"/>
4957         </xs:sequence>
4958     </xs:complexType>
4959 </xs:element>
4960 <xs:element name="UserQueryResponse">
4961     <xs:complexType>
4962         <xs:sequence>
4963             <xs:element minOccurs="1" maxOccurs="1"
4964 ref="emsep:UserQueryResponse"/>
4965         </xs:sequence>
4966     </xs:complexType>
4967 </xs:element>
4968 </xs:schema>
4969 <xs:schema targetNamespace="http://www.e-ms.ca/Schema/v1"
4970 elementFormDefault="qualified"
```

```
4971      xmlns="http://www.e-ms.ca/Schema/v1">
4972      <xs:import namespace="urn:hl7-org:v3"/>
4973      <xs:complexType name="Country" mixed="true">
4974        <xs:complexContent mixed="true">
4975          <xs:extension base="hl7:ADXP"/>
4976        </xs:complexContent>
4977      </xs:complexType>
4978      <xs:complexType name="State" mixed="true">
4979        <xs:complexContent mixed="true">
4980          <xs:extension base="hl7:ADXP"/>
4981        </xs:complexContent>
4982      </xs:complexType>
4983      <xs:complexType name="City" mixed="true">
4984        <xs:complexContent mixed="true">
4985          <xs:extension base="hl7:ADXP"/>
4986        </xs:complexContent>
4987      </xs:complexType>
4988      <xs:complexType name="PostalCode" mixed="true">
4989        <xs:complexContent mixed="true">
4990          <xs:extension base="hl7:ADXP"/>
4991        </xs:complexContent>
4992      </xs:complexType>
4993      <xs:complexType name="HouseNumber" mixed="true">
4994        <xs:complexContent mixed="true">
4995          <xs:extension base="hl7:ADXP"/>
4996        </xs:complexContent>
4997      </xs:complexType>
4998      <xs:complexType name="StreetName" mixed="true">
4999        <xs:complexContent mixed="true">
5000          <xs:extension base="hl7:ADXP"/>
5001        </xs:complexContent>
5002      </xs:complexType>
5003      <xs:complexType name="StreetNameType" mixed="true">
5004        <xs:complexContent mixed="true">
5005          <xs:extension base="hl7:ADXP"/>
5006        </xs:complexContent>
5007      </xs:complexType>
5008      <xs:complexType name="StreetAddressLine" mixed="true">
5009        <xs:complexContent mixed="true">
5010          <xs:extension base="hl7:ADXP"/>
5011        </xs:complexContent>
5012      </xs:complexType>
5013      <xs:complexType name="PostalAddress" mixed="true">
5014        <xs:choice minOccurs="0" maxOccurs="unbounded">
5015          <xs:element name="country" type="Country"
5016minOccurs="0"/>
5017          <xs:element name="state" type="State"
5018minOccurs="0"/>
5019          <xs:element name="city" type="City"
5020minOccurs="0"/>
5021          <xs:element name="postalCode" type="PostalCode"
5022minOccurs="0"/>
5023          <xs:element name="houseNumber"
5024type="HouseNumber" minOccurs="0"/>
5025          <xs:element name="streetName" type="StreetName"
5026minOccurs="0"/>
5027          <xs:element name="streetNameType"
5028type="StreetNameType" minOccurs="0"/>
5029          <xs:element name="streetAddressLine"
5030type="StreetAddressLine" minOccurs="0"
5031maxOccurs="4"/>
5032        </xs:choice>
5033        <xs:attribute name="use"
5034type="hl7:set_cs_PostalAddressUse" use="optional"/>
5035      </xs:complexType>
5036      <xs:complexType name="Telecom">
5037        <xs:complexContent>
5038          <xs:extension base="hl7:TEL"/>
5039        </xs:complexContent>
5040      </xs:complexType>
```

```
5041      <xs:complexType name="PersonName" mixed="true">
5042          <xs:complexContent mixed="true">
5043              <xs:extension base="hl7:PN"/>
5044          </xs:complexContent>
5045      </xs:complexType>
5046      <xs:complexType name="AdministrativeGenderCode">
5047          <xs:complexContent>
5048              <xs:extension base="hl7:CE"/>
5049          </xs:complexContent>
5050      </xs:complexType>
5051      <xs:complexType name="OrgName" mixed="true">
5052          <xs:complexContent mixed="true">
5053              <xs:extension base="hl7:ON"/>
5054          </xs:complexContent>
5055      </xs:complexType>
5056      <xs:group name="OrgDetailGrp">
5057          <xs:sequence>
5058              <xs:element name="MessagingAddress"
5059 type="MessagingAddress.Organization"
5060             minOccurs="0"/>
5061              <xs:element name="Name" type="OrgName"
5062 minOccurs="0" maxOccurs="2"/>
5063              <xs:element name="Telecom" type="Telecom"
5064 minOccurs="0" maxOccurs="unbounded"/>
5065              <xs:element name="PostalAddress"
5066 type="PostalAddress" minOccurs="0"/>
5067              <xs:element name="Note" type="xs:string"
5068 minOccurs="0"/>
5069          </xs:sequence>
5070      </xs:group>
5071      <xs:complexType name="Organization">
5072          <xs:sequence>
5073              <xs:group ref="OrgDetailGrp"/>
5074          </xs:sequence>
5075      </xs:complexType>
5076      <xs:simpleType name="EmptyElement">
5077          <xs:annotation>
5078              <xs:documentation xml:lang="en"> An empty
5079 element. Content must be empty. This
5080             is used to define a non empty type that
5081 does not contain
5082             anything</xs:documentation>
5083          </xs:annotation>
5084          <xs:restriction base="xs:string">
5085              <xs:length value="0"/>
5086          </xs:restriction>
5087      </xs:simpleType>
5088      <xs:complexType name="Coded">
5089          <xs:attribute name="code" type="xs:token"
5090 use="optional">
5091              <xs:annotation>
5092                  <xs:documentation> The plain code symbol
5093 defined by the code system. For
5094             example, "784.0" is the code
5095 symbol of the ICD-9 code "784.0" for
5096             headache.</xs:documentation>
5097          </xs:annotation>
5098          </xs:attribute>
5099          <xs:attribute name="codeSystem" type="xs:string"
5100 use="optional">
5101              <xs:annotation>
5102                  <xs:documentation> Specifies the code
5103 system that defines the
5104             code.</xs:documentation>
5105          </xs:annotation>
5106          </xs:attribute>
5107          <xs:attribute name="codeSystemName" type="xs:string"
5108 use="optional">
5109              <xs:annotation>
```

```
5110         <xs:documentation> A common name of the
5111 coding system.</xs:documentation>
5112         </xs:annotation>
5113         </xs:attribute>
5114         <xs:attribute name="codeSystemVersion" type="xs:string"
5115 use="optional">
5116         <xs:annotation>
5117         <xs:documentation> If applicable, a
5118 version descriptor defined specifically
5119 for the given code
5120 system.</xs:documentation>
5121         </xs:annotation>
5122         </xs:attribute>
5123         <xs:attribute name="displayName" type="xs:string"
5124 use="optional">
5125         <xs:annotation>
5126         <xs:documentation> A name or title for
5127 the code, under which the sending
5128 system shows the code value to
5129 its users. </xs:documentation>
5130         </xs:annotation>
5131         </xs:attribute>
5132     </xs:complexType>
5133     <xs:simpleType name="URN">
5134         <xs:restriction base="xs:anyURI">
5135             <xs:pattern value="urn:.*"/>
5136         </xs:restriction>
5137     </xs:simpleType>
5138     <xs:simpleType name="URN.UUID">
5139         <xs:annotation>
5140         <xs:documentation>uuid elements are string
5141 characters with a maximum length of
5142 36.</xs:documentation>
5143         </xs:annotation>
5144         <xs:restriction base="URN">
5145             <xs:maxLength value="45"/>
5146             <xs:pattern
5147 value="urn:uuid:([0-9]|[a-f]|[A-F]){8}(-
5148 ([0-9]|[a-f]|[A-F]){4})?(-([0-9]|[a-f]|[A-F]){4})?(-([0-
5149 9]|[a-f]|[A-F]){12})?"
5150 />
5151         </xs:restriction>
5152     </xs:simpleType>
5153     <xs:simpleType name="URN.Provider">
5154         <xs:annotation>
5155         <xs:documentation>URN for an addressable
5156 Provider.</xs:documentation>
5157         </xs:annotation>
5158         <xs:restriction base="URN">
5159             <xs:maxLength value="255"/>
5160             <xs:pattern value="urn:provider:.*"/>
5161         </xs:restriction>
5162     </xs:simpleType>
5163     <xs:simpleType name="URN.Organization">
5164         <xs:annotation>
5165         <xs:documentation>URN for an addressable
5166 Organization.</xs:documentation>
5167         </xs:annotation>
5168         <xs:restriction base="URN">
5169             <xs:maxLength value="255"/>
5170             <xs:pattern value="urn:org:.*"/>
5171         </xs:restriction>
5172     </xs:simpleType>
5173     <xs:simpleType name="URN.MessagingAddress">
5174         <xs:restriction base="URN">
5175             <xs:maxLength value="255"/>
5176             <xs:pattern
5177 value="urn:org:.*|urn:provider:.*"/>
5178         </xs:restriction>
5179     </xs:simpleType>
```

```
5180         <xs:complexType name="MessagingAddress">
5181             <xs:attribute name="displayName" type="xs:string"
5182 use="optional"/>
5183             <xs:attribute name="value" type="URN.MessagingAddress"
5184 use="required"/>
5185         </xs:complexType>
5186         <xs:complexType name="MessagingAddress.Provider" final="#all">
5187             <xs:attribute name="displayName" type="xs:string"
5188 use="optional"/>
5189             <xs:attribute name="value" type="URN.Provider"
5190 use="required"/>
5191         </xs:complexType>
5192         <xs:complexType name="MessagingAddress.Organization"
5193 final="#all">
5194             <xs:attribute name="displayName" type="xs:string"
5195 use="optional"/>
5196             <xs:attribute name="value" type="URN.Organization"
5197 use="required"/>
5198         </xs:complexType>
5199         <xs:simpleType name="ProviderID">
5200             <xs:restriction base="URN.UUID"/>
5201         </xs:simpleType>
5202         <xs:complexType name="ProviderSpecialty" mixed="true">
5203             <xs:attribute name="sub" type="xs:boolean"
5204 use="optional" default="false"/>
5205         </xs:complexType>
5206         <xs:simpleType name="UserName">
5207             <xs:restriction base="xs:token">
5208                 <xs:minLength value="1"/>
5209             </xs:restriction>
5210         </xs:simpleType>
5211         <xs:simpleType name="AccessControl">
5212             <xs:restriction base="xs:NMTOKEN">
5213                 <xs:enumeration value="Delegate"/>
5214                 <xs:enumeration value="Owner"/>
5215             </xs:restriction>
5216         </xs:simpleType>
5217         <xs:simpleType name="UserRole">
5218             <xs:restriction base="xs:NMTOKEN">
5219                 <xs:enumeration value="MOA"/>
5220                 <xs:enumeration value="EMR"/>
5221                 <xs:enumeration value="OrgAdministrator"/>
5222                 <xs:enumeration value="Provider"/>
5223             </xs:restriction>
5224         </xs:simpleType>
5225         <xs:complexType name="Role">
5226             <xs:attribute name="value" type="UserRole"
5227 use="required"/>
5228             <xs:attribute name="effectiveFrom" type="xs:dateTime"/>
5229         </xs:complexType>
5230         <xs:simpleType name="Email">
5231             <xs:restriction base="xs:token">
5232                 <xs:pattern
5233 value="([\.a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+((([a-zA-Z0-9_-])*\.[a-zA-Z0-9_-])+)?"
5234 />
5235             </xs:restriction>
5236         </xs:simpleType>
5237         <xs:simpleType name="StatusType">
5238             <xs:restriction base="xs:token">
5239                 <xs:enumeration value="OK"/>
5240                 <xs:enumeration value="ERROR"/>
5241             </xs:restriction>
5242         </xs:simpleType>
5243         <xs:complexType name="ResponseBase">
5244             <xs:attribute name="status" type="StatusType"
5245 use="required"/>
5246         </xs:complexType>
5247         <xs:simpleType name="BillingNumber">
5248             <xs:restriction base="xs:string"/>
5249         </xs:simpleType>
```

```
5250         </xs:simpleType>
5251         <xs:complexType name="Organization.Key">
5252             <xs:sequence>
5253                 <xs:element name="MessagingAddress"
5254 type="MessagingAddress.Organization"/>
5255             </xs:sequence>
5256         </xs:complexType>
5257         <xs:complexType name="Provider.MemberKey">
5258             <xs:choice>
5259                 <xs:element name="MessagingAddress"
5260 type="MessagingAddress.Provider"/>
5261             </xs:choice>
5262         </xs:complexType>
5263         <xs:complexType name="User.Affiliation">
5264             <xs:sequence>
5265                 <xs:element name="Role" type="Role"
5266 maxOccurs="unbounded"/>
5267                 <xs:element name="Organization"
5268 type="Organization"/>
5269             </xs:sequence>
5270         </xs:complexType>
5271         <xs:complexType name="RightsElement">
5272             <xs:attribute name="value" type="AccessControl"
5273 use="required"/>
5274         </xs:complexType>
5275         <xs:group name="AccessRightsGrp">
5276             <xs:sequence>
5277                 <xs:element name="Rights"
5278 type="RightsElement"/>
5279                 <xs:element name="EffectiveFrom"
5280 type="xs:dateTime"/>
5281                 <xs:element name="EffectiveTo"
5282 type="xs:dateTime" nillable="true" minOccurs="0"
5283 />
5284             </xs:sequence>
5285         </xs:group>
5286         <xs:complexType name="AccessRights">
5287             <xs:group ref="AccessRightsGrp"/>
5288         </xs:complexType>
5289         <xs:complexType name="User.AccessRights">
5290             <xs:sequence>
5291                 <xs:choice>
5292                     <xs:element name="Provider"
5293 type="Provider.MemberKey"/>
5294                     <xs:element name="Organization"
5295 type="Organization.Key"/>
5296                 </xs:choice>
5297                 <xs:group ref="AccessRightsGrp"/>
5298             </xs:sequence>
5299         </xs:complexType>
5300         <xs:complexType name="User">
5301             <xs:sequence>
5302                 <xs:element name="UserName" type="UserName"/>
5303                 <xs:element name="FullName" type="xs:string"
5304 minOccurs="0"/>
5305                 <xs:element name="Affiliation"
5306 type="User.Affiliation" minOccurs="0"
5307 maxOccurs="unbounded"/>
5308                 <xs:element name="Access"
5309 type="User.AccessRights" minOccurs="0"
5310 maxOccurs="unbounded"/>
5311             </xs:sequence>
5312         </xs:complexType>
5313         <xs:complexType name="Error" mixed="true">
5314             <xs:sequence>
5315                 <xs:element name="ErrorMsg" type="xs:string"/>
5316                 <xs:element name="MessageID" type="URN.UUID"
5317 minOccurs="0"/>
5318             </xs:sequence>
```



```
5319                                     <xs:attribute name="errorCode" type="xs:integer"
5320 use="required"/>
5321                                     <xs:attribute name="dateTime" type="xs:dateTime"
5322 use="optional"/>
5323                                     </xs:complexType>
5324                                     <xs:complexType name="AdminAccess">
5325                                         <xs:choice>
5326                                             <xs:element name="Add">
5327                                                 <xs:complexType>
5328                                                     <xs:sequence>
5329                                                         <xs:element
5330 name="UserName" type="UserName"/>
5331                                                         <xs:element name="Rights"
5332 type="RightsElement"/>
5333                                                         <xs:element
5334 name="EffectiveFrom" type="xs:dateTime" minOccurs="0"/>
5335                                                         <xs:element
5336 name="EffectiveTo" type="xs:dateTime" minOccurs="0"/>
5337                                                     </xs:sequence>
5338                                                 </xs:complexType>
5339                                             </xs:element>
5340                                             <xs:element name="Remove">
5341                                                 <xs:complexType>
5342                                                     <xs:sequence>
5343                                                         <xs:element
5344 name="UserName" type="UserName"/>
5345                                                         </xs:sequence>
5346                                                 </xs:complexType>
5347                                             </xs:element>
5348                                             <xs:element name="List">
5349                                                 <xs:complexType>
5350                                                     <xs:sequence>
5351                                                         <xs:element name="Rights"
5352 type="RightsElement" minOccurs="0"/>
5353                                                         </xs:sequence>
5354                                                 </xs:complexType>
5355                                             </xs:element>
5356                                         </xs:choice>
5357                                     </xs:complexType>
5358                                     <xs:complexType name="AdminPreferences">
5359                                         <xs:choice>
5360                                             <xs:element name="Update">
5361                                                 <xs:complexType>
5362                                                     <xs:group
5363 ref="AdmPreferencesGrp"/>
5364                                             </xs:complexType>
5365                                             </xs:element>
5366                                             <xs:element name="View" type="EmptyElement"/>
5367                                         </xs:choice>
5368                                     </xs:complexType>
5369                                     <xs:complexType name="AdminPreferencesResp">
5370                                         <xs:choice>
5371                                             <xs:element name="Updated">
5372                                                 <xs:complexType>
5373                                                     <xs:group
5374 ref="AdmPreferencesGrp"/>
5375                                             </xs:complexType>
5376                                             </xs:element>
5377                                             <xs:element name="View">
5378                                                 <xs:complexType>
5379                                                     <xs:group
5380 ref="AdmPreferencesGrp"/>
5381                                             </xs:complexType>
5382                                             </xs:element>
5383                                         </xs:choice>
5384                                     </xs:complexType>
5385                                     <xs:simpleType name="NotificationDelivery">
5386                                         <xs:restriction base="xs:NMTOKEN">
5387                                             <xs:enumeration value="Email"/>
5388                                             <xs:enumeration value="NotificationMessage"/>
```

```
5389         </xs:restriction>
5390     </xs:simpleType>
5391     <xs:complexType name="NotificationSetting">
5392         <xs:attribute name="deliveryMethod"
5393 type="NotificationDelivery" use="required"/>
5394     </xs:complexType>
5395     <xs:group name="AdmPreferencesGrp">
5396         <xs:sequence>
5397             <xs:element name="NotificationEmail"
5398 minOccurs="0" type="Email"/>
5399             <xs:element name="ReadReceipt" minOccurs="0">
5400                 <xs:complexType>
5401                     <xs:attribute name="active"
5402 type="xs:boolean" use="required"/>
5403                 </xs:complexType>
5404             </xs:element>
5405             <xs:element name="NotReadNotification"
5406 minOccurs="0">
5407                 <xs:complexType>
5408                     <xs:attribute name="days"
5409 type="xs:integer" use="required"/>
5410                 </xs:complexType>
5411             </xs:element>
5412             <xs:element name="MessageRetention">
5413                 <xs:complexType>
5414                     <xs:attribute name="days"
5415 type="xs:integer" use="required"/>
5416                 </xs:complexType>
5417             </xs:element>
5418         </xs:sequence>
5419     </xs:group>
5420     <xs:complexType name="User.AccessDetail">
5421         <xs:sequence>
5422             <xs:element name="UserName" type="UserName"/>
5423             <xs:element name="FullName" type="xs:string"
5424 minOccurs="0"/>
5425             <xs:element name="Access" type="AccessRights"
5426 maxOccurs="unbounded"/>
5427         </xs:sequence>
5428     </xs:complexType>
5429     <xs:complexType name="AdminAccessResp">
5430         <xs:choice>
5431             <xs:element name="Added">
5432                 <xs:complexType>
5433                     <xs:sequence>
5434                         <xs:element
5435 name="UserName" type="UserName"/>
5436                     </xs:sequence>
5437                 </xs:complexType>
5438             </xs:element>
5439             <xs:element name="Removed">
5440                 <xs:complexType>
5441                     <xs:sequence>
5442                         <xs:element
5443 name="UserName" type="UserName"/>
5444                     </xs:sequence>
5445                 </xs:complexType>
5446             </xs:element>
5447             <xs:element name="Listing">
5448                 <xs:complexType>
5449                     <xs:sequence>
5450                         <xs:element name="User"
5451 type="User.AccessDetail" minOccurs="0"
5452 maxOccurs="unbounded"/>
5453                     </xs:sequence>
5454                 </xs:complexType>
5455             </xs:element>
5456         </xs:choice>
5457     </xs:complexType>
```

```
5459         <xs:complexType name="OrgAdminProviders">
5460             <xs:choice>
5461                 <xs:element name="Add">
5462                     <xs:complexType>
5463                         <xs:sequence>
5464                             <xs:element
5465 name="ProviderID" type="ProviderID"/>
5466                             <xs:element
5467 name="MessagingAddress" type="MessagingAddress.Provider"
5468 />
5469                         </xs:sequence>
5470                     </xs:complexType>
5471                 </xs:element>
5472                 <xs:element name="Remove">
5473                     <xs:complexType>
5474                         <xs:choice>
5475                             <xs:element
5476 name="ProviderID" type="ProviderID"/>
5477                             <xs:element
5478 name="MessagingAddress" type="MessagingAddress.Provider"
5479 />
5480                         </xs:choice>
5481                     </xs:complexType>
5482                 </xs:element>
5483                 <xs:element name="List" type="EmptyElement"/>
5484             </xs:choice>
5485         </xs:complexType>
5486         <xs:complexType name="OrgAdminDetails">
5487             <xs:choice>
5488                 <xs:element name="Update" type="Organization"/>
5489             </xs:choice>
5490         </xs:complexType>
5491         <xs:complexType name="OrgAdminUsers">
5492             <xs:choice>
5493                 <xs:element name="Add">
5494                     <xs:complexType>
5495                         <xs:sequence>
5496                             <xs:element
5497 name="UserName" type="UserName"/>
5498                             <xs:element name="Role"
5499 type="Role" maxOccurs="unbounded"/>
5500                         </xs:sequence>
5501                     </xs:complexType>
5502                 </xs:element>
5503                 <xs:element name="Remove">
5504                     <xs:complexType>
5505                         <xs:sequence>
5506                             <xs:element
5507 name="UserName" type="UserName"/>
5508                             <xs:element name="Role"
5509 type="Role" minOccurs="0"
5510 maxOccurs="unbounded"/>
5511                         </xs:sequence>
5512                     </xs:complexType>
5513                 </xs:element>
5514                 <xs:element name="List">
5515                     <xs:complexType>
5516                         <xs:sequence>
5517                             <xs:element name="Role"
5518 type="Role" minOccurs="0"
5519 maxOccurs="unbounded"/>
5520                         </xs:sequence>
5521                     </xs:complexType>
5522                 </xs:element>
5523             </xs:choice>
5524         </xs:complexType>
5525         <xs:complexType name="OrgAdminResp.User">
5526             <xs:sequence>
```

```
5529         <xs:element name="UserName" type="UserName"/>
5530         <xs:element name="FullName" type="xs:string"/>
5531         <xs:element name="Role" type="Role"
5532 maxOccurs="unbounded"/>
5533     </xs:sequence>
5534 </xs:complexType>
5535 <xs:complexType name="OrgAdminUsersResp">
5536     <xs:choice>
5537         <xs:element name="Added">
5538             <xs:complexType>
5539                 <xs:sequence>
5540                     <xs:element
5541 name="UserName" type="UserName"/>
5542                 </xs:sequence>
5543             </xs:complexType>
5544         </xs:element>
5545         <xs:element name="Removed">
5546             <xs:complexType>
5547                 <xs:sequence>
5548                     <xs:element
5549 name="UserName" type="UserName"/>
5550                 </xs:sequence>
5551             </xs:complexType>
5552         </xs:element>
5553         <xs:element name="Listing">
5554             <xs:complexType>
5555                 <xs:sequence>
5556                     <xs:element name="User"
5557 type="OrgAdminResp.User" minOccurs="0"
5558 maxOccurs="unbounded"/>
5559                 </xs:sequence>
5560             </xs:complexType>
5561         </xs:element>
5562     </xs:choice>
5563 </xs:complexType>
5564 <xs:complexType name="OrgAdmin">
5565     <xs:sequence>
5566         <xs:element name="Organization"
5567 type="Organization.Key"/>
5568     </xs:sequence>
5569     <xs:choice>
5570         <xs:element name="Access"
5571 type="AdminAccess"/>
5572         <xs:element name="Users"
5573 type="OrgAdminUsers"/>
5574         <xs:element name="Providers"
5575 type="OrgAdminProviders"/>
5576         <xs:element name="Preferences"
5577 type="AdminPreferences"/>
5578         <xs:element name="Details"
5579 type="OrgAdminDetails"/>
5580     </xs:choice>
5581 </xs:sequence>
5582 </xs:complexType>
5583 <xs:complexType name="OrgAdminDetailsResp">
5584     <xs:choice>
5585         <xs:element name="Updated"
5586 type="EmptyElement"/>
5587     </xs:choice>
5588 </xs:complexType>
5589 <xs:complexType name="OrgAdminResponse">
5590     <xs:complexContent>
5591         <xs:extension base="ResponseBase">
5592             <xs:sequence>
5593                 <xs:element name="Organization"
5594 type="Organization.Key"/>
5595             </xs:sequence>
5596             <xs:choice>
5597                 <xs:element name="Access"
```

```
5598                                     <xs:element name="Users"
5599 type="OrgAdminUsersResp"/>
5600                                     <xs:element
5601 name="Providers" type="OrgAdminMembersResp"/>
5602                                     <xs:element
5603 name="Preferences" type="AdminPreferencesResp"/>
5604                                     <xs:element
5605 name="Details" type="OrgAdminDetailsResp"/>
5606                                     <xs:sequence>
5607                                         <xs:element
5608 name="Error" type="Error"/>
5609                                     </xs:sequence>
5610                                     </xs:choice>
5611                                     </xs:sequence>
5612                                     </xs:extension>
5613                                     </xs:complexContent>
5614                                     </xs:complexType>
5615                                     <xs:complexType name="Provider.Member">
5616                                         <xs:sequence>
5617                                             <xs:element name="ProviderID"
5618 type="ProviderID"/>
5619                                             <xs:element name="MessagingAddress"
5620 type="MessagingAddress.Provider"/>
5621                                             <xs:element name="EffectiveFrom"
5622 type="xs:dateTime" minOccurs="0"/>
5623                                         </xs:sequence>
5624                                     </xs:complexType>
5625                                     <xs:complexType name="OrgAdminMembersResp">
5626                                         <xs:choice>
5627                                             <xs:element name="Added">
5628                                                 <xs:complexType>
5629                                                     <xs:sequence>
5630                                                         <xs:element
5631 name="ProviderID" type="ProviderID"/>
5632                                                         <xs:element
5633 name="MesssagingAddress"
5634 type="MessagingAddress.Provider"/>
5635                                                     </xs:sequence>
5636                                                 </xs:complexType>
5637                                             </xs:element>
5638                                             <xs:element name="Removed">
5639                                                 <xs:complexType>
5640                                                     <xs:sequence>
5641                                                         <xs:element
5642 name="ProviderID" type="ProviderID"/>
5643                                                         <xs:element
5644 name="Listing">
5645                                                 <xs:complexType>
5646                                                     <xs:sequence>
5647                                                         <xs:element
5648 name="Provider" type="Provider.Member" minOccurs="0"
5649 maxOccurs="unbounded"/>
5650                                                     </xs:sequence>
5651                                                 </xs:complexType>
5652                                             </xs:element>
5653                                         </xs:choice>
5654                                     </xs:complexType>
5655                                     </xs:sequence>
5656                                     </xs:choice>
5657                                     </xs:complexType>
5658                                     <xs:complexType name="ProviderAdmin">
5659                                         <xs:sequence>
5660                                             <xs:element name="Provider"
5661 type="Provider.MemberKey"/>
5662                                         <xs:choice>
5663                                             <xs:element name="Access"
5664 type="AdminAccess"/>
5665                                             <xs:element name="Preferences"
5666 type="AdminPreferences"/>
5667
```

```
5668                                     <xs:element name="MessagingAddress"
5669 type="MessagingAddress.Provider"/>
5670                                     <xs:element name="Note"
5671 type="xs:string"/>
5672                                     </xs:choice>
5673                                     </xs:sequence>
5674                                     </xs:complexType>
5675                                     <xs:complexType name="ProviderAdminResponse">
5676                                     <xs:complexContent>
5677                                     <xs:extension base="ResponseBase">
5678                                     <xs:sequence>
5679                                     <xs:element name="Provider"
5680 type="Provider.Member"/>
5681                                     <xs:choice>
5682                                     <xs:element name="Access"
5683 type="AdminAccessResp"/>
5684                                     <xs:element
5685 name="Preferences" type="AdminPreferencesResp"/>
5686                                     <xs:element
5687 name="MessagingAddress" type="AddressUpdateResp"/>
5688                                     <xs:element name="Note"
5689 type="NoteUpdateResp"/>
5690                                     <xs:sequence>
5691                                     <xs:element
5692 name="Error" type="Error"/>
5693                                     </xs:sequence>
5694                                     </xs:choice>
5695                                     </xs:sequence>
5696                                     </xs:extension>
5697                                     </xs:complexContent>
5698                                     </xs:complexType>
5699                                     <xs:complexType name="AddressUpdateResp">
5700                                     <xs:attribute name="updated" type="xs:boolean"
5701 default="true"/>
5702                                     </xs:complexType>
5703                                     <xs:complexType name="NoteUpdateResp">
5704                                     <xs:attribute name="updated" type="xs:boolean"
5705 default="true"/>
5706                                     </xs:complexType>
5707                                     <xs:complexType name="UserAdmin">
5708                                     <xs:sequence>
5709                                     <xs:element name="UserName" type="UserName"/>
5710                                     <xs:choice>
5711                                     <xs:element name="ListAccess">
5712                                     <xs:complexType>
5713                                     <xs:attribute name="list"
5714 fixed="true"/>
5715                                     </xs:complexType>
5716                                     </xs:element>
5717                                     </xs:choice>
5718                                     </xs:sequence>
5719                                     </xs:complexType>
5720                                     <xs:complexType name="UserAdminResponse">
5721                                     <xs:complexContent>
5722                                     <xs:extension base="ResponseBase">
5723                                     <xs:choice>
5724                                     <xs:element name="User"
5725 type="User"/>
5726                                     <xs:sequence>
5727                                     <xs:element name="Error"
5728 type="Error"/>
5729                                     </xs:sequence>
5730                                     </xs:choice>
5731                                     </xs:extension>
5732                                     </xs:complexContent>
5733                                     </xs:complexType>
5734                                     <xs:simpleType name="ResultViewType">
5735                                     <xs:restriction base="xs:NMTOKEN">
5736                                     <xs:enumeration value="Detail"/>
5737                                     <xs:enumeration value="Summary"/>
```

```
5738         </xs:restriction>
5739     </xs:simpleType>
5740     <xs:group name="OrgSearchParamGrp">
5741         <xs:sequence>
5742             <xs:element name="Name" type="OrgName"
5743 minOccurs="0" maxOccurs="2"/>
5744             <xs:element name="Telecom" type="Telecom"
5745 minOccurs="0" maxOccurs="unbounded"/>
5746             <xs:element name="PostalAddress"
5747 type="PostalAddress" minOccurs="0"/>
5748         </xs:sequence>
5749     </xs:group>
5750     <xs:complexType name="OrganizationQuery">
5751         <xs:sequence>
5752             <xs:choice>
5753                 <xs:element name="MessagingAddress"
5754 type="MessagingAddress.Organization"/>
5755                 <xs:group ref="OrgSearchParamGrp"/>
5756             </xs:choice>
5757         </xs:sequence>
5758     </xs:complexType>
5759     <xs:complexType name="OrgQueryResponse">
5760         <xs:complexContent>
5761             <xs:extension base="ResponseBase">
5762                 <xs:choice>
5763                     <xs:element name="Organization"
5764 type="OrganizationResult" minOccurs="0"
5765                                     maxOccurs="unbounded"/>
5766                     <xs:sequence>
5767                         <xs:element name="Error"
5768 type="Error"/>
5769                     </xs:sequence>
5770                 </xs:choice>
5771             </xs:extension>
5772         </xs:complexContent>
5773     </xs:complexType>
5774     <xs:complexType name="OrganizationResult">
5775         <xs:sequence>
5776             <xs:group ref="OrgDetailGrp"/>
5777         </xs:sequence>
5778         <xs:attribute name="view" type="ResultViewType"/>
5779     </xs:complexType>
5780     <xs:complexType name="ProviderQuery">
5781         <xs:choice>
5782             <xs:element name="ProviderID"
5783 type="ProviderID"/>
5784             <xs:element name="MessagingAddress"
5785 type="MessagingAddress.Provider"/>
5786             <xs:element name="Listing"
5787 type="ProviderListing"/>
5788             <xs:sequence minOccurs="1" maxOccurs="7">
5789                 <xs:group ref="ProviderSearchGrp"/>
5790             </xs:sequence>
5791         </xs:choice>
5792     </xs:complexType>
5793     <xs:group name="ProviderSearchGrp">
5794         <xs:choice>
5795             <xs:element name="Name" type="PersonName"/>
5796             <xs:element name="PostalAddress"
5797 type="PostalAddress"/>
5798             <xs:element name="Telecom" type="Telecom"
5799 maxOccurs="unbounded"/>
5800             <xs:element name="BillingNumber"
5801 type="BillingNumber"/>
5802             <xs:element name="Specialty"
5803 type="ProviderSpecialty" maxOccurs="2"/>
5804             <xs:element name="AdministrativeGenderCode"
5805 type="AdministrativeGenderCode"/>
5806             <xs:element name="Organization"
5807 type="OrganizationQuery"/>
```



```
5808         </xs:choice>
5809     </xs:group>
5810     <xs:complexType name="ProviderListing">
5811         <xs:attribute name="modifiedSince" type="xs:dateTime"
5812 use="optional"/>
5813     </xs:complexType>
5814     <xs:complexType name="ProviderQueryResponse">
5815         <xs:complexContent>
5816             <xs:extension base="ResponseBase">
5817                 <xs:choice>
5818                     <xs:element name="Provider"
5819 type="Provider" minOccurs="0"
5820                                     maxOccurs="unbounded"/>
5821                     <xs:sequence>
5822                         <xs:element name="Error"
5823 type="Error"/>
5824                     </xs:sequence>
5825                 </xs:choice>
5826             </xs:extension>
5827         </xs:complexContent>
5828     </xs:complexType>
5829     <xs:complexType name="ProviderMembership">
5830         <xs:sequence>
5831             <xs:element name="MessagingAddress"
5832 type="MessagingAddress.Provider"/>
5833             <xs:element name="Organization"
5834 type="Organization"/>
5835             <xs:element name="Telecom" type="Telecom"
5836 minOccurs="0" maxOccurs="unbounded"/>
5837             <xs:element name="Note" type="xs:string"
5838 minOccurs="0"/>
5839         </xs:sequence>
5840     </xs:complexType>
5841     <xs:complexType name="Provider">
5842         <xs:sequence>
5843             <xs:element name="ProviderID"
5844 type="ProviderID"/>
5845             <xs:element name="PersonName"
5846 type="PersonName"/>
5847             <xs:element name="BillingNumber"
5848 type="BillingNumber" minOccurs="0"/>
5849             <xs:element name="Specialty"
5850 type="ProviderSpecialty" minOccurs="0"
5851                                     maxOccurs="unbounded"/>
5852             <xs:element name="AdministrativeGenderCode"
5853 type="AdministrativeGenderCode"
5854                                     minOccurs="0"/>
5855             <xs:element name="Note" type="xs:string"
5856 minOccurs="0"/>
5857             <xs:element name="PostalAddress"
5858 type="PostalAddress" minOccurs="0"/>
5859             <xs:element name="Telecom" type="Telecom"
5860 minOccurs="0" maxOccurs="unbounded"/>
5861             <xs:element name="Membership"
5862 type="ProviderMembership" minOccurs="0"
5863                                     maxOccurs="unbounded"/>
5864         </xs:sequence>
5865         <xs:attribute name="view" type="ResultViewType"/>
5866     </xs:complexType>
5867     <xs:complexType name="UserQuery">
5868         <xs:choice>
5869             <xs:element name="UserName" type="UserName"/>
5870             <xs:group ref="UserQueryParamGrp"/>
5871         </xs:choice>
5872     </xs:complexType>
5873     <xs:group name="UserQueryParamGrp">
5874         <xs:sequence>
5875             <xs:element name="FullName" type="xs:string"
5876 minOccurs="0"/>
```

```
5877         <xs:element name="Role" type="UserRole"
5878 minOccurs="0" maxOccurs="unbounded"/>
5879         <xs:element name="Organization"
5880 type="OrganizationQuery" minOccurs="0"/>
5881     </xs:sequence>
5882 </xs:group>
5883 <xs:complexType name="UserQueryResponse">
5884     <xs:complexContent>
5885         <xs:extension base="ResponseBase">
5886             <xs:choice>
5887                 <xs:element name="User"
5888 type="UserResult" minOccurs="0"
5889                                     maxOccurs="unbounded"/>
5890                 <xs:sequence>
5891                     <xs:element name="Error"
5892 type="Error"/>
5893                 </xs:sequence>
5894             </xs:choice>
5895         </xs:extension>
5896     </xs:complexContent>
5897 </xs:complexType>
5898 <xs:complexType name="UserResult">
5899     <xs:complexContent>
5900         <xs:extension base="User">
5901             <xs:attribute name="view"
5902 type="ResultViewType"/>
5903         </xs:extension>
5904     </xs:complexContent>
5905 </xs:complexType>
5906 <xs:simpleType name="MessageID">
5907     <xs:restriction base="URN.UUID"/>
5908 </xs:simpleType>
5909 <xs:simpleType name="PriorityType">
5910     <xs:annotation>
5911         <xs:documentation>Based upon common Mail X-
5912 Priority extension</xs:documentation>
5913     </xs:annotation>
5914     <xs:restriction base="xs:token">
5915         <xs:enumeration value="Highest"/>
5916         <xs:enumeration value="High"/>
5917         <xs:enumeration value="Normal"/>
5918         <xs:enumeration value="Low"/>
5919         <xs:enumeration value="Lowest"/>
5920     </xs:restriction>
5921 </xs:simpleType>
5922 <xs:simpleType name="MessageTypes">
5923     <xs:restriction base="xs:token">
5924         <xs:enumeration value="ClinicalDocument"/>
5925         <xs:enumeration value="Notification"/>
5926         <xs:enumeration value="Other"/>
5927     </xs:restriction>
5928 </xs:simpleType>
5929 <xs:complexType name="MessageType">
5930     <xs:complexContent>
5931         <xs:extension base="Coded">
5932             <xs:attribute name="type"
5933 type="MessageTypes" use="required"/>
5934             <xs:attribute name="value"
5935 type="xs:token" use="required"/>
5936         </xs:extension>
5937     </xs:complexContent>
5938 </xs:complexType>
5939 <xs:group name="MessageHeaderGrp">
5940     <xs:sequence>
5941         <xs:element name="From"
5942 type="MessagingAddress"/>
5943         <xs:element name="To" type="MessagingAddress"/>
5944         <xs:element name="Subject" type="xs:string"
5945 minOccurs="0"/>

```

```
5946         <xs:element name="Priority" type="PriorityType"
5947 minOccurs="0"/>
5948         <xs:element name="MessageID" type="MessageID"/>
5949         </xs:sequence>
5950     </xs:group>
5951     <xs:group name="MessageHeaderRxGrp">
5952         <xs:sequence>
5953             <xs:group ref="MessageHeaderGrp"/>
5954             <xs:element name="MessageType"
5955 type="MessageType" minOccurs="0"/>
5956             <xs:element name="Date" type="xs:dateTime"/>
5957         </xs:sequence>
5958     </xs:group>
5959     <xs:complexType name="MessageRequestBase">
5960         <xs:sequence>
5961             <xs:element name="MessageID" type="MessageID"/>
5962         </xs:sequence>
5963     </xs:complexType>
5964     <xs:complexType name="MessageInfo">
5965         <xs:sequence>
5966             <xs:group ref="MessageHeaderRxGrp"/>
5967             <xs:element name="Read" minOccurs="0">
5968                 <xs:complexType>
5969                     <xs:attribute name="value"
5970 type="xs:boolean" use="required"/>
5971                     <xs:attribute name="dateTime"
5972 type="xs:dateTime" use="optional"/>
5973                 </xs:complexType>
5974             </xs:element>
5975             <xs:element name="Received" minOccurs="0">
5976                 <xs:complexType>
5977                     <xs:attribute name="value"
5978 type="xs:boolean" use="required"/>
5979                     <xs:attribute name="dateTime"
5980 type="xs:dateTime" use="optional"/>
5981                 </xs:complexType>
5982             </xs:element>
5983         </xs:sequence>
5984     </xs:complexType>
5985     <xs:complexType name="MessageRefResponse">
5986         <xs:complexContent>
5987             <xs:extension base="ResponseBase">
5988                 <xs:choice>
5989                     <xs:sequence>
5990                         <xs:element
5991 name="MessageID" type="MessageID"/>
5992                     </xs:sequence>
5993                     <xs:sequence>
5994                         <xs:element name="Error">
5995                             <xs:complexType
5996 mixed="true">
5997                                 <xs:complexContent mixed="true">
5998                                     <xs:extension base="Error"/>
5999                                 </xs:complexContent>
6000                             </xs:complexType>
6001                         </xs:element>
6002                     </xs:sequence>
6003                 </xs:choice>
6004             </xs:extension>
6005         </xs:complexContent>
6006     </xs:complexType>
6007     <xs:simpleType name="AttachmentFormat">
6008         <xs:restriction base="xs:token">
6009             <xs:enumeration value="MIME"/>
6010         </xs:restriction>
6011     </xs:simpleType>
6012     <xs:simpleType name="MIME">
```

```
6016         <xs:restriction base="xs:string">
6017             <xs:whiteSpace value="collapse"/>
6018             <xs:minLength value="15"/>
6019             <xs:pattern value=".*MIME-Version.*"/>
6020         </xs:restriction>
6021     </xs:simpleType>
6022     <xs:complexType name="Attachment">
6023         <xs:simpleContent>
6024             <xs:extension base="MIME">
6025                 <xs:attribute name="fileName"
type="xs:anyURI" use="required"/>
6026             </xs:extension>
6027         </xs:simpleContent>
6028     </xs:complexType>
6029     <xs:simpleType name="MessageEventType">
6030         <xs:restriction base="xs:string">
6031             <xs:enumeration value="NotRead-Purged">
6032                 <xs:annotation/>
6033             </xs:enumeration>
6034             <xs:enumeration value="NotRead-WillPurge">
6035                 <xs:annotation/>
6036             </xs:enumeration>
6037             <xs:enumeration value="NotRead-
WithinExpectedDays">
6038                 <xs:annotation/>
6039             </xs:enumeration>
6040             <xs:enumeration value="ReadConfirmation">
6041                 <xs:annotation/>
6042             </xs:enumeration>
6043         </xs:restriction>
6044     </xs:simpleType>
6045     <xs:complexType name="MessageEvent">
6046         <xs:sequence>
6047             <xs:element name="MessageInfo"
type="MessageInfo"/>
6048         </xs:sequence>
6049         <xs:attribute name="type" type="MessageEventType"
use="required"/>
6050     </xs:complexType>
6051     <xs:complexType name="Notification">
6052         <xs:sequence>
6053             <xs:element name="MessageEvent"
type="MessageEvent" minOccurs="0"/>
6054             <xs:element name="Note"
type="xs:normalizedString"/>
6055         </xs:sequence>
6056     </xs:complexType>
6057     <xs:group name="ContentGrp.Any">
6058         <xs:sequence>
6059             <xs:any namespace="##other"
processContents="skip"/>
6060             <xs:element name="Attachment" type="Attachment"
minOccurs="0"
maxOccurs="unbounded"/>
6061         </xs:sequence>
6062     </xs:group>
6063     <xs:complexType name="ContentRx.Any">
6064         <xs:choice>
6065             <xs:group ref="ContentGrp.Any"/>
6066             <xs:element name="Notification"
type="Notification"/>
6067         </xs:choice>
6068     </xs:complexType>
6069     <xs:complexType name="ContentTx.Any">
6070         <xs:group ref="ContentGrp.Any"/>
6071     </xs:complexType>
6072     <xs:complexType name="MessageRx.Any">
6073         <xs:sequence>
6074             <xs:group ref="MessageHeaderRxGrp"/>
6075         </xs:sequence>
6076     </xs:complexType>
```

```
6085         <xs:element name="Content"
6086 type="ContentRx.Any"/>
6087     </xs:sequence>
6088 </xs:complexType>
6089 <xs:complexType name="MessageTx.Any">
6090     <xs:sequence>
6091         <xs:group ref="MessageHeaderGrp"/>
6092         <xs:element name="Content"
6093 type="ContentTx.Any"/>
6094     </xs:sequence>
6095 </xs:complexType>
6096 <xs:complexType name="MessageAck">
6097     <xs:complexContent>
6098         <xs:extension base="MessageRequestBase"/>
6099     </xs:complexContent>
6100 </xs:complexType>
6101 <xs:complexType name="MessageAckResponse">
6102     <xs:complexContent>
6103         <xs:extension base="MessageRefResponse"/>
6104     </xs:complexContent>
6105 </xs:complexType>
6106 <xs:simpleType name="MessageBoxStatus">
6107     <xs:restriction base="EmptyElement"/>
6108 </xs:simpleType>
6109 <xs:complexType name="MessageBoxStatusResponse">
6110     <xs:complexContent>
6111         <xs:extension base="ResponseBase">
6112             <xs:choice>
6113                 <xs:sequence>
6114                     <xs:element
6115 name="MessageCount" type="xs:long"/>
6116                 </xs:sequence>
6117                 <xs:sequence>
6118                     <xs:element name="Error"
6119 type="Error"/>
6120                 </xs:sequence>
6121             </xs:choice>
6122         </xs:extension>
6123     </xs:complexContent>
6124 </xs:complexType>
6125 <xs:complexType name="MessageListing">
6126     <xs:attribute name="showHeaderInfo" type="xs:boolean"
6127 use="optional" default="false"/>
6128     <xs:attribute name="received" type="xs:boolean"
6129 use="optional" default="false"/>
6130 </xs:complexType>
6131 <xs:complexType name="MessageListingResponse">
6132     <xs:complexContent>
6133         <xs:extension base="ResponseBase">
6134             <xs:choice>
6135                 <xs:sequence>
6136                     <xs:element
6137 name="MessageInfo" type="MessageInfo" minOccurs="0"
6138 maxOccurs="unbounded"/>
6139                 </xs:sequence>
6140                 <xs:sequence>
6141                     <xs:element
6142 name="MessageID" type="MessageID" minOccurs="0"
6143 maxOccurs="unbounded"/>
6144                 </xs:sequence>
6145                 <xs:sequence>
6146                     <xs:element name="Error"
6147 type="Error"/>
6148                 </xs:sequence>
6149             </xs:choice>
6150         </xs:extension>
6151     </xs:complexContent>
6152 </xs:complexType>
```

```
6155         <xs:complexType name="MessageRead">
6156             <xs:complexContent>
6157                 <xs:extension base="MessageRequestBase"/>
6158             </xs:complexContent>
6159         </xs:complexType>
6160         <xs:complexType name="MessageReadResponse">
6161             <xs:complexContent>
6162                 <xs:extension base="MessageRefResponse"/>
6163             </xs:complexContent>
6164         </xs:complexType>
6165         <xs:complexType name="MessageRecv">
6166             <xs:complexContent>
6167                 <xs:extension base="MessageRequestBase"/>
6168             </xs:complexContent>
6169         </xs:complexType>
6170         <xs:complexType name="MessageRecvResponse.Any">
6171             <xs:complexContent>
6172                 <xs:extension base="ResponseBase">
6173                     <xs:choice>
6174                         <xs:element name="Message"
6175 type="MessageRx.Any"/>
6176                         <xs:element name="Error"
6177 type="Error"/>
6178                     </xs:choice>
6179                 </xs:extension>
6180             </xs:complexContent>
6181         </xs:complexType>
6182         <xs:complexType name="MessageReset">
6183             <xs:sequence>
6184                 <xs:choice minOccurs="0">
6185                     <xs:element name="DateTime"
6186 type="xs:dateTime"/>
6187                     <xs:element name="MessageID"
6188 type="MessageID"/>
6189                 </xs:choice>
6190             </xs:sequence>
6191         </xs:complexType>
6192         <xs:complexType name="MessageResetResponse">
6193             <xs:complexContent>
6194                 <xs:extension base="ResponseBase">
6195                     <xs:choice>
6196                         <xs:sequence>
6197                             <xs:element
6198 name="MessageCount" type="xs:long"/>
6199                             <xs:choice minOccurs="0">
6200                                 <xs:element
6201 name="DateTime" type="xs:dateTime"/>
6202                                 <xs:element
6203 name="MessageID" type="MessageID"
6204 maxOccurs="unbounded"/>
6205                             </xs:choice>
6206                         </xs:sequence>
6207                     </xs:choice>
6208                 </xs:extension>
6209             </xs:complexContent>
6210         </xs:complexType>
6211         <xs:sequence>
6212             <xs:choice>
6213                 <xs:extension>
6214                     <xs:complexContent>
6215                         </xs:complexType>
6216                         <xs:complexType name="MessageSend.Any">
6217                             <xs:sequence>
6218                                 <xs:element name="Message"
6219 type="MessageTx.Any"/>
6220                             </xs:sequence>
6221                         </xs:complexType>
6222                         <xs:complexType name="MessageSendResponse">
6223                             <xs:complexContent>
6224                                 <xs:extension base="MessageRefResponse"/>
```

```
6225         </xs:complexContent>
6226     </xs:complexType>
6227     <xs:complexType name="MessageTrack">
6228         <xs:complexContent>
6229             <xs:extension base="MessageRequestBase"/>
6230         </xs:complexContent>
6231     </xs:complexType>
6232     <xs:complexType name="MessageTrackResponse">
6233         <xs:complexContent>
6234             <xs:extension base="ResponseBase">
6235                 <xs:choice>
6236                     <xs:sequence>
6237                         <xs:element
6238 name="MessageInfo" type="MessageInfo"/>
6239                     </xs:sequence>
6240                     <xs:sequence>
6241                         <xs:element name="Error"
6242 type="Error"/>
6243                     </xs:sequence>
6244                 </xs:choice>
6245             </xs:extension>
6246         </xs:complexContent>
6247     </xs:complexType>
6248     <xs:annotation
6249 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6250         <xs:element name="OrgAdmin" type="OrgAdmin"
6251 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6252         <xs:element name="OrgAdminResponse" type="OrgAdminResponse"
6253 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6254             <xs:annotation/>
6255             </xs:element>
6256             <xs:element name="ProviderAdmin" type="ProviderAdmin"
6257 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6258             <xs:element name="ProviderAdminResponse"
6259 type="ProviderAdminResponse"
6260 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6261             <xs:annotation/>
6262             </xs:element>
6263             <xs:element name="UserAdmin" type="UserAdmin"
6264 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6265             <xs:annotation/>
6266             </xs:element>
6267             <xs:element name="UserAdminResponse" type="UserAdminResponse"
6268 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6269             <xs:annotation/>
6270             </xs:element>
6271             <xs:element name="MessageAck" type="MessageAck"
6272 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6273             <xs:annotation/>
6274             </xs:element>
6275             <xs:element name="MessageAckResponse"
6276 type="MessageAckResponse"
6277 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6278             <xs:annotation/>
6279             </xs:element>
6280             <xs:element name="MessageBoxStatus" type="MessageBoxStatus"
6281 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6282             <xs:annotation/>
6283             </xs:element>
6284             <xs:element name="MessageBoxStatusResponse"
6285 type="MessageBoxStatusResponse"
6286 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6287             <xs:annotation/>
6288             </xs:element>
6289             <xs:element name="MessageListing" type="MessageListing"
6290 xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6291             <xs:annotation/>
6292             </xs:element>
6293             <xs:element name="MessageListing" type="MessageListing"
```



```
6294      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6295      <xs:element name="MessageListingResponse"
6296      type="MessageListingResponse"
6297
6298      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6299      <xs:annotation/>
6300      </xs:element>
6301      <xs:element name="MessageRead" type="MessageRead"
6302
6303      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6304      <xs:element name="MessageReadResponse"
6305      type="MessageReadResponse"
6306
6307      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6308      <xs:annotation/>
6309      </xs:element>
6310      <xs:element name="MessageRecv" type="MessageRecv"
6311
6312      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6313      <xs:element name="MessageRecvResponse"
6314      type="MessageRecvResponse.Any"
6315
6316      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6317      <xs:annotation/>
6318      </xs:element>
6319      <xs:element name="MessageReset" type="MessageReset"
6320
6321      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6322      <xs:element name="MessageResetResponse"
6323      type="MessageResetResponse"
6324
6325      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6326      <xs:annotation/>
6327      </xs:element>
6328      <xs:element name="MessageSend" type="MessageSend.Any"
6329
6330      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6331      <xs:element name="MessageSendResponse"
6332      type="MessageSendResponse"
6333
6334      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6335      <xs:annotation/>
6336      </xs:element>
6337      <xs:element name="MessageTrack" type="MessageTrack"
6338
6339      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6340      <xs:element name="MessageTrackResponse"
6341      type="MessageTrackResponse"
6342
6343      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6344      <xs:annotation/>
6345      </xs:element>
6346      <xs:element name="OrgQuery" type="OrganizationQuery"
6347
6348      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6349      <xs:element name="OrgQueryResponse" type="OrgQueryResponse"
6350
6351      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6352      <xs:annotation/>
6353      </xs:element>
6354      <xs:element name="ProviderQuery" type="ProviderQuery"
6355
6356      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
6357      <xs:element name="ProviderQueryResponse"
6358      type="ProviderQueryResponse"
6359
6360      xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
6361      <xs:annotation/>
6362      </xs:element>
6363
```

```

        <xs:element name="UserQuery" type="UserQuery"
xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema"/>
        <xs:element name="UserQueryResponse" type="UserQueryResponse"
xmlns:ns0="http://eMSSchemas.PropertySchema.PropertySchema">
        <xs:annotation/>
    </xs:element>
</xs:schema>
<xs:schema targetNamespace="urn:hl7-org:v3"
elementFormDefault="qualified"
xmlns="urn:hl7-org:v3">
    <xsd:complexType name="ANY" abstract="true"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron">
        <xsd:annotation>
            <xsd:documentation> Defines the basic
properties of every data value. This is an
abstract type, meaning that no value can
be just a data value without
belonging to any concrete type. Every
concrete type is a specialization of
this general abstract DataValue type.
        </xsd:documentation>
        </xsd:annotation>
        <xsd:attribute name="nullFlavor" type="cs_NullFlavor"
use="optional">
            <xsd:annotation>
                <xsd:documentation> An exceptional value
expressing missing information and
possibly the reason why the
information is missing. </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
    </xsd:complexType>
    <xsd:simpleType name="cs_NullFlavor"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron">
        <xsd:restriction base="cs">
            <xsd:enumeration value="NI"/>
            <xsd:enumeration value="NA"/>
            <xsd:enumeration value="UNK"/>
            <xsd:enumeration value="NASK"/>
            <xsd:enumeration value="ASKU"/>
            <xsd:enumeration value="NAV"/>
            <xsd:enumeration value="OTH"/>
            <xsd:enumeration value="PINF"/>
            <xsd:enumeration value="NINF"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="bl"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron">
        <xsd:annotation>
            <xsd:documentation> The Boolean type stands for
the values of two-valued logic.
A Boolean value can be either true or
false, or, as any other value may be
NULL.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:boolean">
            <xsd:pattern value="true|false"/>
        </xsd:restriction>
    </xsd:simpleType>

```

```
6434      <xsd:complexType name="BL"
6435 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6436
6437      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6438      xmlns:sch="http://www.ascc.net/xml/schematron">
6439      <xsd:annotation>
6440      <xsd:documentation> The Boolean type stands for
6441 the values of two-valued logic.
6442
6443      A Boolean value can be either true or
6444 false, or, as any other value may be
6445 NULL.</xsd:documentation>
6446      </xsd:annotation>
6447      <xsd:complexContent>
6448      <xsd:extension base="ANY">
6449      <xsd:attribute name="value" type="bl"
6450 use="optional"/>
6451      </xsd:extension>
6452      </xsd:complexContent>
6453      </xsd:complexType>
6454      <xsd:simpleType name="bn"
6455 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6456
6457      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6458      xmlns:sch="http://www.ascc.net/xml/schematron">
6459      <xsd:annotation>
6460      <xsd:documentation> The BooleanNonNull type is
6461 used where a Boolean cannot have
6462 a null value. A Boolean value can be
6463 either true or
6464 false.</xsd:documentation>
6465      </xsd:annotation>
6466      <xsd:restriction base="bl"/>
6467      </xsd:simpleType>
6468      <xsd:simpleType name="bin"
6469 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6470
6471      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6472      xmlns:sch="http://www.ascc.net/xml/schematron">
6473      <xsd:annotation>
6474      <xsd:documentation> Binary data is a raw block
6475 of bits. Binary data is a
6476 protected type that should not be
6477 declared outside the data type
6478 specification.</xsd:documentation>
6479      </xsd:annotation>
6480      <xsd:restriction base="xsd:base64Binary"/>
6481      </xsd:simpleType>
6482      <xsd:simpleType name="cs_BinaryDataEncoding"
6483 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6484
6485      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6486      xmlns:sch="http://www.ascc.net/xml/schematron">
6487      <xsd:restriction base="xsd:NMTOKEN">
6488      <xsd:enumeration value="B64"/>
6489      <xsd:enumeration value="TXT"/>
6490      </xsd:restriction>
6491      </xsd:simpleType>
6492      <xsd:complexType name="BIN" abstract="true" mixed="true"
6493      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6494
6495      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6496      xmlns:sch="http://www.ascc.net/xml/schematron">
6497      <xsd:annotation>
6498      <xsd:documentation> Binary data is a raw block
6499 of bits. Binary data is a
6500 protected type that should not be
6501 declared outside the data type
6502 specification.</xsd:documentation>
6503      </xsd:annotation>
6504      <xsd:complexContent mixed="true">
```

```

6504         <xsd:extension base="ANY">
6505             <xsd:attribute name="representation"
6506 type="cs_BinaryDataEncoding"
6507                 use="optional" default="TXT">
6508                 <xsd:annotation>
6509                     <xsd:documentation>
6510 Specifies the representation of the binary data
6511                                     that is the
6512 content of the binary data complex XML schema data
6513
6514 type.</xsd:documentation>
6515                 </xsd:annotation>
6516             </xsd:attribute>
6517         </xsd:extension>
6518     </xsd:complexContent>
6519 </xsd:complexType>
6520 <xsd:simpleType name="cs_CompressionAlgorithm"
6521     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6522
6523     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6524     xmlns:sch="http://www.ascc.net/xml/schematron">
6525     <xsd:restriction base="cs">
6526         <xsd:enumeration value="DF"/>
6527         <xsd:enumeration value="GZ"/>
6528         <xsd:enumeration value="ZL"/>
6529         <xsd:enumeration value="Z"/>
6530     </xsd:restriction>
6531 </xsd:simpleType>
6532 <xsd:simpleType name="cs_IntegrityCheckAlgorithm"
6533     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6534
6535     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6536     xmlns:sch="http://www.ascc.net/xml/schematron">
6537     <xsd:restriction base="cs">
6538         <xsd:enumeration value="SHA-1"/>
6539         <xsd:enumeration value="SHA-256"/>
6540     </xsd:restriction>
6541 </xsd:simpleType>
6542 <xsd:complexType name="ED" mixed="true"
6543 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6544
6545     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6546     xmlns:sch="http://www.ascc.net/xml/schematron">
6547     <xsd:annotation>
6548         <xsd:documentation> Data that is primarily
6549 intended for human interpretation or
6550                                     for further machine processing is
6551 outside the scope of HL7. This includes
6552                                     unformatted or formatted written
6553 language, multimedia data, or structured
6554                                     information as defined by a different
6555 standard (e.g., XML-signatures.)
6556                                     Instead of the data itself, an ED may
6557 contain only a reference (see TEL.)
6558                                     Note that the ST data type is a
6559 specialization of the ED data type when the
6560                                     ED media type is
6561 text/plain.</xsd:documentation>
6562     </xsd:annotation>
6563     <xsd:complexContent mixed="true">
6564         <xsd:extension base="BIN">
6565             <xsd:sequence>
6566                 <xsd:element name="reference"
6567 type="TEL" minOccurs="0">
6568                                     <xsd:annotation>
6569
6570         <xsd:documentation> A telecommunication address (TEL), such as a
6571                                     URL for
6572 HTTP or FTP, which will resolve to precisely the
```

```
6573 same
6574 binary data that could as well have been provided as
6575 inline
6576 data.</xsd:documentation>
6577 </xsd:annotation>
6578 </xsd:element>
6579 <xsd:element name="thumbnail"
6580 type="thumbnail" minOccurs="0"/>
6581 </xsd:sequence>
6582 <xsd:attribute name="mediaType"
6583 type="cs" use="optional"
6584 default="text/plain">
6585 <xsd:annotation>
6586 <xsd:documentation>
6587 Identifies the type of the encapsulated data and
6588 identifies a
6589 method to interpret or render the
6590 data.</xsd:documentation>
6591 </xsd:annotation>
6592 </xsd:attribute>
6593 <xsd:attribute name="language" type="cs"
6594 use="optional">
6595 <xsd:annotation>
6596 <xsd:documentation> For
6597 character based information the language
6598 property
6599 specifies the human language of the
6600 text.</xsd:documentation>
6601 </xsd:annotation>
6602 </xsd:attribute>
6603 <xsd:attribute name="compression"
6604 type="cs_CompressionAlgorithm"
6605 use="optional">
6606 <xsd:annotation>
6607 <xsd:documentation>
6608 Indicates whether the raw byte data is
6609 compressed, and
6610 what compression algorithm was
6611 used.</xsd:documentation>
6612 </xsd:annotation>
6613 </xsd:attribute>
6614 <xsd:attribute name="integrityCheck"
6615 type="bin" use="optional">
6616 <xsd:annotation>
6617 <xsd:documentation> The
6618 integrity check is a short binary value
6619 representing a
6620 cryptographically strong checksum that is
6621 calculated over
6622 the binary data. The purpose of this property,
6623 when communicated
6624 with a reference is for anyone to validate
6625 later whether the
6626 reference still resolved to the same data that
6627 the reference
6628 resolved to when the encapsulated data value with
6629 reference was
6630 created.</xsd:documentation>
6631 </xsd:annotation>
6632 </xsd:attribute>
6633 <xsd:attribute
6634 name="integrityCheckAlgorithm"
6635 type="cs_IntegrityCheckAlgorithm" use="optional" default="SHA-1">
6636 <xsd:annotation>
6637 <xsd:documentation>
6638 Specifies the algorithm used to compute the
6639 integrityCheck
6640 value.</xsd:documentation>
6641 </xsd:annotation>
```

```
6643         </xsd:attribute>
6644     </xsd:extension>
6645     </xsd:complexContent>
6646 </xsd:complexType>
6647 <xsd:complexType name="thumbnail"
6648 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6649
6650     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6651     xmlns:sch="http://www.ascc.net/xml/schematron">
6652     <xsd:annotation>
6653         <xsd:documentation> A thumbnail is an
6654 abbreviated rendition of the full data. A
6655 thumbnail requires significantly fewer
6656 resources than the full data, while
6657 still maintaining some distinctive
6658 similarity with the full data. A
6659 thumbnail is typically used with by-
6660 reference encapsulated data. It allows a
6661 user to select data more efficiently
6662 before actually downloading through the
6663 reference.</xsd:documentation>
6664     </xsd:annotation>
6665     <xsd:complexContent>
6666         <xsd:restriction base="ED">
6667             <xsd:sequence>
6668                 <xsd:element name="reference"
6669 type="TEL" minOccurs="0"/>
6670                 <xsd:element name="thumbnail"
6671 type="thumbnail" minOccurs="0"
6672 maxOccurs="0"/>
6673             </xsd:sequence>
6674         </xsd:restriction>
6675     </xsd:complexContent>
6676 </xsd:complexType>
6677 <xsd:simpleType name="st"
6678 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6679
6680     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6681     xmlns:sch="http://www.ascc.net/xml/schematron">
6682     <xsd:annotation>
6683         <xsd:documentation> The character string data
6684 type stands for text data,
6685 primarily intended for machine
6686 processing (e.g., sorting, querying,
6687 indexing, etc.) Used for names, symbols,
6688 and formal
6689 expressions.</xsd:documentation>
6690     </xsd:annotation>
6691     <xsd:restriction base="xsd:string">
6692         <xsd:minLength value="1"/>
6693     </xsd:restriction>
6694 </xsd:simpleType>
6695 <xsd:complexType name="ST" mixed="true"
6696 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6697
6698     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6699     xmlns:sch="http://www.ascc.net/xml/schematron">
6700     <xsd:annotation>
6701         <xsd:documentation> The character string data
6702 type stands for text data,
6703 primarily intended for machine
6704 processing (e.g., sorting, querying,
6705 indexing, etc.) Used for names, symbols,
6706 and formal
6707 expressions.</xsd:documentation>
6708     </xsd:annotation>
6709     <xsd:complexContent mixed="true">
6710         <xsd:restriction base="ED">
6711             <xsd:sequence>
```

```
6712                                     <xsd:element name="reference"
6713 type="TEL" minOccurs="0" maxOccurs="0"/>
6714                                     <xsd:element name="thumbnail"
6715 type="ED" minOccurs="0" maxOccurs="0"/>
6716                                     </xsd:sequence>
6717                                     <xsd:attribute name="representation"
6718 type="cs_BinaryDataEncoding"
6719                                     fixed="TXT"/>
6720                                     <xsd:attribute name="mediaType"
6721 type="cs" fixed="text/plain"/>
6722                                     <xsd:attribute name="language" type="cs"
6723 use="optional"/>
6724                                     <xsd:attribute name="compression"
6725 type="cs_CompressionAlgorithm"
6726                                     use="prohibited"/>
6727                                     <xsd:attribute name="integrityCheck"
6728 type="bin" use="prohibited"/>
6729                                     <xsd:attribute
6730 name="integrityCheckAlgorithm"
6731                                     type="cs_IntegrityCheckAlgorithm" use="prohibited"/>
6732                                     </xsd:restriction>
6733                                     </xsd:complexContent>
6734                                     </xsd:complexType>
6735                                     <xsd:simpleType name="cs"
6736 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6737                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6738                                     xmlns:sch="http://www.ascc.net/xml/schematron">
6739                                     <xsd:annotation>
6740                                     <xsd:documentation> Coded data in its simplest
6741 form, consists of a code and
6742                                     original text. The code system and code
6743 system version is fixed by the
6744                                     context in which the CS value occurs. CS
6745 is used for coded attributes that
6746                                     have a single HL7-defined value
6747                                     set.</xsd:documentation>
6748                                     </xsd:annotation>
6749                                     <xsd:restriction base="xsd:token">
6750                                     <xsd:pattern value="^[\\s]*"/>
6751                                     </xsd:restriction>
6752                                     </xsd:simpleType>
6753                                     <xsd:complexType name="CD"
6754                                     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6755                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6756                                     xmlns:sch="http://www.ascc.net/xml/schematron">
6757                                     <xsd:annotation>
6758                                     <xsd:documentation> A concept descriptor
6759 represents any kind of concept usually
6760                                     by giving a code defined in a code
6761 system. A concept descriptor can contain
6762                                     the original text or phrase that served
6763 as the basis of the coding and one
6764                                     or more translations into different
6765 coding systems. A concept descriptor can
6766                                     also contain qualifiers to describe,
6767 e.g., the concept of a "left foot" as a
6768                                     postcoordinated term built from the
6769 primary code "FOOT" and the qualifier
6770                                     "LEFT". In exceptional cases, the
6771 concept descriptor need not contain a code
6772                                     but only the original text describing
6773 that concept.</xsd:documentation>
6774                                     </xsd:annotation>
6775                                     <xsd:complexContent>
6776                                     <xsd:extension base="ANY">
6777                                     <xsd:sequence>
```



```
6781                                     <xsd:element name="originalText"
6782 type="ED" minOccurs="0">
6783                                     <xsd:annotation>
6784
6785         <xsd:documentation> The text or phrase used as the basis for the
6786
6787         coding.</xsd:documentation>
6788                                     </xsd:annotation>
6789                                     </xsd:element>
6790                                     <xsd:element name="qualifier"
6791 type="CR" minOccurs="0"
6792                                     maxOccurs="unbounded">
6793         <xsd:annotation>
6794
6795         <xsd:documentation> Specifies additional codes that increase the
6796
6797         specificity of the primary code.</xsd:documentation>
6798                                     </xsd:annotation>
6799                                     </xsd:element>
6800                                     <xsd:element name="translation"
6801 type="CD" minOccurs="0"
6802                                     maxOccurs="unbounded">
6803         <xsd:annotation>
6804
6805         <xsd:documentation> A set of other concept descriptors that
6806
6807         this concept descriptor into other code
6808
6809         systems.</xsd:documentation>
6810                                     </xsd:annotation>
6811                                     </xsd:element>
6812         </xsd:sequence>
6813         <xsd:attribute name="code" type="cs"
6814 use="optional">
6815         <xsd:annotation>
6816         <xsd:documentation> The
6817 plain code symbol defined by the code
6818
6819         system. For
6820 example, "784.0" is the code symbol of the ICD-9
6821
6822         code "784.0" for
6823         headache.</xsd:documentation>
6824                                     </xsd:annotation>
6825         </xsd:attribute>
6826         <xsd:attribute name="codeSystem"
6827 type="uid" use="optional">
6828         <xsd:annotation>
6829         <xsd:documentation>
6830
6831         Specifies the code system that defines the
6832
6833         code.</xsd:documentation>
6834                                     </xsd:annotation>
6835         </xsd:attribute>
6836         <xsd:attribute name="codeSystemName"
6837 type="st" use="optional">
6838         <xsd:annotation>
6839         <xsd:documentation> A
6840
6841         common name of the coding
6842
6843         system.</xsd:documentation>
6844                                     </xsd:annotation>
6845         </xsd:attribute>
6846         <xsd:attribute name="codeSystemVersion"
6847 type="st" use="optional">
6848         <xsd:annotation>
6849         <xsd:documentation> If
6850
6851         applicable, a version descriptor defined
6852
6853         specifically for
6854         the given code system.</xsd:documentation>
6855                                     </xsd:annotation>
6856         </xsd:attribute>
```

```
6850                                     <xsd:attribute name="displayName"
6851 type="st" use="optional">
6852                                     <xsd:annotation>
6853                                     <xsd:documentation> A
6854 name or title for the code, under which the
6855                                     sending system
6856 shows the code value to its
6857
6858 users.</xsd:documentation>
6859                                     </xsd:annotation>
6860                                     </xsd:attribute>
6861                                     </xsd:extension>
6862                                     </xsd:complexContent>
6863                                     </xsd:complexType>
6864                                     <xsd:complexType name="CE"
6865 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6866
6867         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6868         xmlns:sch="http://www.ascc.net/xml/schematron">
6869         <xsd:annotation>
6870         <xsd:documentation> Coded data, consists of a
6871 coded value (CV) and, optionally,
6872         coded value(s) from other coding systems
6873 that identify the same concept.
6874         Used when alternative codes may
6875 exist.</xsd:documentation>
6876         </xsd:annotation>
6877         <xsd:complexContent>
6878         <xsd:restriction base="CD">
6879         <xsd:sequence>
6880         <xsd:element name="qualifier"
6881 type="CR" minOccurs="0" maxOccurs="0"/>
6882         <xsd:element name="originalText"
6883 type="ED" minOccurs="0"/>
6884         <xsd:annotation>
6885
6886 <xsd:documentation> The text or phrase used as the basis for the
6887 coding.</xsd:documentation>
6888         </xsd:annotation>
6889         </xsd:element>
6890         <xsd:element name="translation"
6891 type="CD" minOccurs="0"
6892         maxOccurs="unbounded">
6893         <xsd:annotation>
6894
6895 <xsd:documentation> A set of other concept descriptors that
6896 this concept descriptor into other code
6897         translate
6898 systems.</xsd:documentation>
6899         </xsd:annotation>
6900         </xsd:element>
6901         </xsd:sequence>
6902         <xsd:attribute name="code" type="cs"
6903 use="optional">
6904         <xsd:annotation>
6905         <xsd:documentation> The
6906 plain code symbol defined by the code
6907         system. For
6908 example, "784.0" is the code symbol of the ICD-9
6909         code "784.0" for
6910 headache.</xsd:documentation>
6911         </xsd:annotation>
6912         </xsd:attribute>
6913         <xsd:attribute name="codeSystem"
6914 type="uid" use="optional">
6915         <xsd:annotation>
6916         <xsd:documentation>
6917 Specifies the code system that defines the
6918
6919
```

```
6920                                     code.</xsd:documentation>
6921                                     </xsd:annotation>
6922                                     </xsd:attribute>
6923                                     <xsd:attribute name="codeSystemName"
6924 type="st" use="optional">
6925                                     <xsd:annotation>
6926                                     <xsd:documentation> A
6927 common name of the coding
6928 system.</xsd:documentation>
6929                                     </xsd:annotation>
6930                                     </xsd:attribute>
6931                                     <xsd:attribute name="codeSystemVersion"
6932 type="st" use="optional">
6933                                     <xsd:annotation>
6934                                     <xsd:documentation> If
6935 applicable, a version descriptor defined
6936                                     specifically for
6937 the given code system.</xsd:documentation>
6938                                     </xsd:annotation>
6939                                     </xsd:attribute>
6940                                     <xsd:attribute name="displayName"
6941 type="st" use="optional">
6942                                     <xsd:annotation>
6943                                     <xsd:documentation> A
6944 name or title for the code, under which the
6945                                     sending system
6946 shows the code value to its
6947 users.</xsd:documentation>
6948                                     </xsd:annotation>
6949                                     </xsd:attribute>
6950                                     </xsd:restriction>
6951                                     </xsd:complexType>
6952                                     <xsd:complexType name="CV"
6953 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
6954                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
6955                                     xmlns:sch="http://www.ascc.net/xml/schematron">
6956                                     <xsd:annotation>
6957                                     <xsd:documentation> Coded data, consists of a
6958 code, display name, code system,
6959                                     and original text. Used when a single
6960 code value must be
6961                                     sent.</xsd:documentation>
6962                                     </xsd:annotation>
6963                                     <xsd:complexContent>
6964                                     <xsd:restriction base="CE">
6965                                     <xsd:sequence>
6966                                     <xsd:element name="originalText"
6967 type="ED" minOccurs="0">
6968                                     <xsd:annotation>
6969                                     <xsd:documentation> The text or phrase used as the basis for the
6970 coding.</xsd:documentation>
6971                                     </xsd:annotation>
6972                                     </xsd:element>
6973                                     <xsd:element name="translation"
6974 type="CD" minOccurs="0" maxOccurs="0"/>
6975                                     </xsd:sequence>
6976                                     <xsd:attribute name="code" type="cs"
6977 use="optional">
6978                                     <xsd:annotation>
6979                                     <xsd:documentation> The
6980 plain code symbol defined by the code
6981                                     system. For
6982 example, "784.0" is the code symbol of the ICD-9
```

```
6989 code "784.0" for
6990 headache.</xsd:documentation>
6991 </xsd:annotation>
6992 </xsd:attribute>
6993 <xsd:attribute name="codeSystem"
6994 type="uid" use="optional">
6995 <xsd:annotation>
6996 <xsd:documentation>
6997 Specifies the code system that defines the
6998 code.</xsd:documentation>
6999 </xsd:annotation>
7000 </xsd:attribute>
7001 <xsd:attribute name="codeSystemName"
7002 type="st" use="optional">
7003 <xsd:annotation>
7004 <xsd:documentation> A
7005 common name of the coding
7006 system.</xsd:documentation>
7007 </xsd:annotation>
7008 </xsd:attribute>
7009 <xsd:attribute name="codeSystemVersion"
7010 type="st" use="optional">
7011 <xsd:annotation>
7012 <xsd:documentation> If
7013 applicable, a version descriptor defined
7014 specifically for
7015 the given code system.</xsd:documentation>
7016 </xsd:annotation>
7017 </xsd:attribute>
7018 <xsd:attribute name="displayName"
7019 type="st" use="optional">
7020 <xsd:annotation>
7021 <xsd:documentation> A
7022 name or title for the code, under which the
7023 sending system
7024 shows the code value to its
7025 users.</xsd:documentation>
7026 </xsd:annotation>
7027 </xsd:attribute>
7028 </xsd:restriction>
7029 </xsd:complexType>
7030 </xsd:complexType>
7031 <xsd:complexType name="CS"
7032 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7033 xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7034 xmlns:sch="http://www.ascc.net/xml/schematron">
7035 <xsd:annotation>
7036 <xsd:documentation> Coded data, consists of a
7037 code, display name, code system,
7038 and original text. Used when a single
7039 code value must be
7040 sent.</xsd:documentation>
7041 </xsd:annotation>
7042 <xsd:complexType>
7043 <xsd:restriction base="CV">
7044 <xsd:sequence/>
7045 <xsd:attribute name="code" type="cs"
7046 use="optional">
7047 <xsd:annotation>
7048 <xsd:documentation> The
7049 plain code symbol defined by the code
7050 system. For
7051 example, "784.0" is the code symbol of the ICD-9
7052 code "784.0" for
7053 headache.</xsd:documentation>
7054 </xsd:annotation>
7055 </xsd:attribute>
```

```
7059                                     <xsd:attribute name="codeSystem"
7060 type="uid"/>
7061                                     <xsd:attribute name="codeSystemName"
7062 type="st"/>
7063                                     <xsd:attribute name="codeSystemVersion"
7064 type="st"/>
7065                                     <xsd:attribute name="displayName"
7066 type="st" use="prohibited"/>
7067                                     </xsd:restriction>
7068                                     </xsd:complexContent>
7069                                     </xsd:complexType>
7070                                     <xsd:complexType name="CO"
7071 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7072
7073         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7074         xmlns:sch="http://www.ascc.net/xml/schematron">
7075         <xsd:annotation>
7076             <xsd:documentation> Coded data, where the
7077 domain from which the codeset comes is
7078 ordered. The Coded Ordinal data type
7079 adds semantics related to ordering so
7080 that models that make use of such
7081 domains may introduce model elements that
7082 involve statements about the order of
7083 the terms in a domain. The
7084 representation is exactly the same as
7085 CV, but the type still needs to be
7086 defined.</xsd:documentation>
7087         </xsd:annotation>
7088         <xsd:complexContent>
7089             <xsd:extension base="CV"/>
7090         </xsd:complexContent>
7091     </xsd:complexType>
7092     <xsd:complexType name="CR"
7093 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7094
7095         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7096         xmlns:sch="http://www.ascc.net/xml/schematron">
7097         <xsd:annotation>
7098             <xsd:documentation> A concept qualifier code
7099 with optionally named role. Both
7100 qualifier role and value codes must be
7101 defined by the coding system. For
7102 example, if SNOMED RT defines a concept
7103 "leg", a role relation
7104 "has-laterality", and another concept
7105 "left", the concept role relation
7106 allows to add the qualifier "has-
7107 laterality: left" to a primary code "leg"
7108 to construct the meaning "left
7109 leg".</xsd:documentation>
7110         </xsd:annotation>
7111         <xsd:complexContent>
7112             <xsd:extension base="CD">
7113                 <xsd:sequence>
7114                     <xsd:element name="name"
7115 type="CV" minOccurs="0"/>
7116                 </xsd:sequence>
7117             </xsd:extension>
7118             <xsd:documentation> Specifies the manner in which the concept
7119 role value
7120 contributes to the meaning of a code phrase. For
7121 example,
7122 if SNOMED RT defines a concept "leg", a role
7123 relation
7124 "has-laterality", and another concept "left", the
7125 concept
7126 role relation allows to add the qualifier
7127 "has-
7128 laterality: left" to a primary code "leg" to construct
```

```
7129 meaning "left leg". In this example "has-laterality" is the
7130
7131 CR.name.</xsd:documentation> the
7132
7133 </xsd:annotation>
7134 </xsd:element>
7135 <xsd:element name="value"
7136 type="CD" minOccurs="0">
7137 <xsd:annotation>
7138
7139 <xsd:documentation> The concept that modifies the primary code
7140 of a code
7141 phrase through the role relation. For example, if SNOMED RT
7142 defines a concept "leg", a role relation
7143 "has-
7144 laterality", and another concept "left", the concept
7145 role
7146 relation allows adding the qualifier "has-laterality:
7147 left" to a
7148 primary code "leg" to construct the meaning "left
7149 leg". In
7150 this example "left" is the
7151
7152 CR.value.</xsd:documentation>
7153 </xsd:annotation>
7154 </xsd:element>
7155 </xsd:sequence>
7156 <xsd:attribute name="inverted" type="bl"
7157 use="optional" default="false">
7158 <xsd:annotation>
7159 <xsd:documentation>
7160 Indicates if the sense of the role name is
7161 inverted. This
7162 can be used in cases where the underlying code
7163 system defines
7164 inversion but does not provide reciprocal pairs
7165 of role names. By
7166 default, inverted is
7167
7168 false.</xsd:documentation>
7169 </xsd:annotation>
7170 </xsd:attribute>
7171 </xsd:extension>
7172 </xsd:complexContent>
7173 </xsd:complexType>
7174 <xsd:complexType name="SC" mixed="true"
7175 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7176
7177 xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7178 xmlns:sch="http://www.ascc.net/xml/schematron">
7179 <xsd:annotation>
7180 <xsd:documentation> A Character String that
7181 optionally may have a code attached.
7182 The text must always be present if a
7183 code is present. The code is often a
7184 local code. </xsd:documentation>
7185 </xsd:annotation>
7186 <xsd:complexContent mixed="true">
7187 <xsd:extension base="ST">
7188 <xsd:attribute name="code" type="cs"
7189 use="optional">
7190 <xsd:annotation>
7191 <xsd:documentation> The
7192 plain code symbol defined by the code
7193 system. For
7194 example, "784.0" is the code symbol of the ICD-9
7195 code "784.0" for
7196 headache.</xsd:documentation>
7197 </xsd:annotation>
```

Page 135 of 171


```

7269         <xsd:documentation/>
7270     </xsd:annotation>
7271     <xsd:restriction base="xsd:string">
7272         <xsd:pattern
7273             value="[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}-[0-
7274 9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{12}"
7275         />
7276     </xsd:restriction>
7277 </xsd:simpleType>
7278 <xsd:simpleType name="ruid"
7279 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7280
7281     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7282     xmlns:sch="http://www.ascc.net/xml/schematron">
7283     <xsd:annotation>
7284         <xsd:documentation/>
7285     </xsd:annotation>
7286     <xsd:restriction base="xsd:string">
7287         <xsd:pattern value="[A-Za-z][A-Za-z0-9\-\_]*"/>
7288     </xsd:restriction>
7289 </xsd:simpleType>
7290 <xsd:complexType name="II"
7291 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7292
7293     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7294     xmlns:sch="http://www.ascc.net/xml/schematron">
7295     <xsd:annotation>
7296         <xsd:documentation> An identifier that uniquely
7297 identifies a thing or object.
7298
7299             Examples are object identifier for HL7
7300 RIM objects, medical record number,
7301             order id, service catalog item id,
7302 Vehicle Identification Number (VIN), etc.
7303             Instance identifiers are defined based
7304 on ISO object
7305             identifiers.</xsd:documentation>
7306     </xsd:annotation>
7307     <xsd:complexContent>
7308         <xsd:extension base="ANY">
7309             <xsd:attribute name="root" type="uid"
7310 use="optional">
7311                 <xsd:annotation>
7312                     <xsd:documentation> A
7313 unique identifier that guarantees the global
7314 uniqueness of the
7315 instance identifier. The root alone may be the
7316 entire instance
7317 identifier.</xsd:documentation>
7318                 </xsd:annotation>
7319             </xsd:attribute>
7320             <xsd:attribute name="extension"
7321 type="st" use="optional">
7322                 <xsd:annotation>
7323                     <xsd:documentation> A
7324 character string as a unique identifier within
7325 the scope of the
7326 identifier root.</xsd:documentation>
7327                 </xsd:annotation>
7328             </xsd:attribute>
7329             <xsd:attribute
7330 name="assigningAuthorityName" type="st" use="optional">
7331                 <xsd:annotation>
7332                     <xsd:documentation> A
7333 human readable name or mnemonic for the
7334 assigning
7335 authority. This name may be provided solely for the
7336 convenience of
7337 unaided humans interpreting an II value and can
7338 have no
7339 computational meaning. Note: no automated processing

```

```
7339                                     must depend on
7340 the assigning authority name to be present in any
7341
7342     form.</xsd:documentation>
7343                                     </xsd:annotation>
7344                                     </xsd:attribute>
7345                                     <xsd:attribute name="displayable"
7346 type="bl" use="optional">
7347                                     <xsd:annotation>
7348                                     <xsd:documentation>
7349 Specifies if the identifier is intended for
7350                                     human display and
7351 data entry (displayable = true) as opposed to
7352                                     pure machine
7353 interoperation (displayable =
7354 false).</xsd:documentation>
7355                                     </xsd:annotation>
7356                                     </xsd:attribute>
7357                                     </xsd:extension>
7358                                     </xsd:complexContent>
7359                                     </xsd:complexType>
7360                                     <xsd:simpleType name="url"
7361 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7362
7363     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7364     xmlns:sch="http://www.ascc.net/xml/schematron">
7365     <xsd:annotation>
7366     <xsd:documentation> A telecommunications
7367 address specified according to Internet
7368 standard RFC 1738
7369 [http://www.ietf.org/rfc/rfc1738.txt]. The URL specifies
7370 the protocol and the contact point
7371 defined by that protocol for the
7372 resource. Notable uses of the
7373 telecommunication address data type are for
7374 telephone and telefax numbers, e-mail
7375 addresses, Hypertext references, FTP
7376 references, etc.</xsd:documentation>
7377     </xsd:annotation>
7378     <xsd:restriction base="xsd:anyURI"/>
7379     </xsd:simpleType>
7380     <xsd:complexType name="URL"
7381 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7382
7383     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7384     xmlns:sch="http://www.ascc.net/xml/schematron">
7385     <xsd:annotation>
7386     <xsd:documentation> A telecommunications
7387 address specified according to Internet
7388 standard RFC 1738
7389 [http://www.ietf.org/rfc/rfc1738.txt]. The URL specifies
7390 the protocol and the contact point
7391 defined by that protocol for the
7392 resource. Notable uses of the
7393 telecommunication address data type are for
7394 telephone and telefax numbers, e-mail
7395 addresses, Hypertext references, FTP
7396 references, etc.</xsd:documentation>
7397     </xsd:annotation>
7398     <xsd:complexContent>
7399     <xsd:extension base="ANY">
7400     <xsd:attribute name="value" type="url"
7401 use="optional"/>
7402     </xsd:extension>
7403     </xsd:complexContent>
7404     </xsd:complexType>
7405     <xsd:simpleType name="ts"
7406 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
```

```
7408      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7409      xmlns:sch="http://www.ascc.net/xml/schematron">
7410      <xsd:annotation>
7411      <xsd:documentation> A quantity specifying a
7412      point on the axis of natural time. A
7413      point in time is most often represented
7414      as a calendar
7415      expression.</xsd:documentation>
7416      </xsd:annotation>
7417      <xsd:restriction base="xsd:string">
7418      <xsd:pattern
7419      value="[0-9]{1,8}|([0-9]{9,14}|[0-
7420      9]{14,14})\.[0-9]+)([+|-][0-9]{1,4})?">/>
7421      </xsd:restriction>
7422      </xsd:simpleType>
7423      <xsd:complexType name="TS"
7424      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7425      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7426      xmlns:sch="http://www.ascc.net/xml/schematron">
7427      <xsd:annotation>
7428      <xsd:documentation> A quantity specifying a
7429      point on the axis of natural time. A
7430      point in time is most often represented
7431      as a calendar
7432      expression.</xsd:documentation>
7433      </xsd:annotation>
7434      <xsd:complexContent>
7435      <xsd:extension base="QTY">
7436      <xsd:attribute name="value" type="ts"
7437      use="optional"/>
7438      </xsd:extension>
7439      </xsd:complexContent>
7440      </xsd:complexType>
7441      <xsd:simpleType name="cs_TelecommunicationAddressUse"
7442      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7443      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7444      xmlns:sch="http://www.ascc.net/xml/schematron">
7445      <xsd:restriction base="cs">
7446      <xsd:enumeration value="H"/>
7447      <xsd:enumeration value="HP"/>
7448      <xsd:enumeration value="HV"/>
7449      <xsd:enumeration value="WP"/>
7450      <xsd:enumeration value="AS"/>
7451      <xsd:enumeration value="EC"/>
7452      <xsd:enumeration value="PG"/>
7453      <xsd:enumeration value="MC"/>
7454      </xsd:restriction>
7455      </xsd:simpleType>
7456      <xsd:complexType name="TEL"
7457      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7458      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7459      xmlns:sch="http://www.ascc.net/xml/schematron">
7460      <xsd:annotation>
7461      <xsd:documentation> A telephone number (voice
7462      or fax), e-mail address, or other
7463      locator for a resource (information or
7464      service) mediated by
7465      telecommunication equipment. The address
7466      is specified as a Universal
7467      Resource Locator (URL) qualified by time
7468      specification and use codes that
7469      help in deciding which address to use
7470      for a given time and
7471      purpose.</xsd:documentation>
7472      </xsd:annotation>
7473      <xsd:complexContent>
```

```
7478         <xsd:extension base="URL">
7479             <xsd:sequence>
7480                 <xsd:element
7481 name="useablePeriod" type="SXCM_TS" minOccurs="0"
7482                                     maxOccurs="unbounded">
7483                     <xsd:annotation>
7484
7485         <xsd:documentation> Specifies the periods of time during which
7486                                     the
7487 telecommunication address can be used. For a telephone
7488                                     number,
7489 this can indicate the time of day in which the party
7490                                     can be
7491 reached on that telephone. For a web address, it may
7492                                     specify a
7493 time range in which the web content is promised to
7494                                     be
7495 available under the given address.</xsd:documentation>
7496                                     </xsd:annotation>
7497                                     </xsd:element>
7498             </xsd:sequence>
7499             <xsd:attribute name="use"
7500 type="set_cs_TelecommunicationAddressUse"
7501                                     use="optional">
7502                 <xsd:annotation>
7503                     <xsd:documentation> One
7504 or more codes advising a system or user
7505                                     which
7506 telecommunication address in a set of like addresses to
7507                                     select for a
7508 given telecommunication need.</xsd:documentation>
7509                     </xsd:annotation>
7510                 </xsd:attribute>
7511             </xsd:extension>
7512         </xsd:complexContent>
7513     </xsd:complexType>
7514     <xsd:simpleType name="cs_AddressPartType"
7515 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7516
7517     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7518     xmlns:sch="http://www.ascc.net/xml/schematron">
7519     <xsd:restriction base="cs">
7520         <xsd:enumeration value="DEL"/>
7521         <xsd:enumeration value="CNT"/>
7522         <xsd:enumeration value="STA"/>
7523         <xsd:enumeration value="CPA"/>
7524         <xsd:enumeration value="CTY"/>
7525         <xsd:enumeration value="ZIP"/>
7526         <xsd:enumeration value="SAL"/>
7527         <xsd:enumeration value="BNR"/>
7528         <xsd:enumeration value="BNN"/>
7529         <xsd:enumeration value="DIR"/>
7530         <xsd:enumeration value="STR"/>
7531         <xsd:enumeration value="STB"/>
7532         <xsd:enumeration value="STTYP"/>
7533         <xsd:enumeration value="ADL"/>
7534         <xsd:enumeration value="UNID"/>
7535         <xsd:enumeration value="UNIT"/>
7536         <xsd:enumeration value="CAR"/>
7537         <xsd:enumeration value="CEN"/>
7538     </xsd:restriction>
7539     </xsd:simpleType>
7540     <xsd:complexType name="ADXP" mixed="true"
7541 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
7542
7543     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
7544     xmlns:sch="http://www.ascc.net/xml/schematron">
7545     <xsd:annotation>
7546         <xsd:documentation> A character string that may
7547 have a type-tag signifying its
```

role in the address. Typical parts that exist in about every address are street, house number, or post box, postal code, city, country but other roles may be defined regionally, nationally, or on an enterprise level (e.g. in military addresses). Addresses are usually broken up into lines, which are indicated by special line-breaking delimiter elements (e.g.,

```
DEL).</xsd:documentation>
</xsd:annotation>
<xsd:complexContent mixed="true">
  <xsd:extension base="ST">
    <xsd:attribute name="partType"
type="cs_AddressPartType">
      <xsd:annotation>
        <xsd:documentation>
Specifies whether an address part names the street, city,
country, postal code, post box, etc. If the type
is NULL the
address part is unclassified and would simply appear on an address
label as is.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="cs_PostalAddressUse"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
  xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron">
  <xsd:restriction base="cs">
    <xsd:enumeration value="PHYS"/>
    <xsd:enumeration value="PST"/>
    <xsd:enumeration value="TMP"/>
    <xsd:enumeration value="BAD"/>
    <xsd:enumeration value="H"/>
    <xsd:enumeration value="HP"/>
    <xsd:enumeration value="HV"/>
    <xsd:enumeration value="WP"/>
    <xsd:enumeration value="ABC"/>
    <xsd:enumeration value="SYL"/>
    <xsd:enumeration value="IDE"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AD" mixed="true"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
  xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron">
  <xsd:annotation>
    <xsd:documentation> Mailing and home or office
addresses. A sequence of address
      parts, such as street or post office
Box, city, postal code, country,
      etc.</xsd:documentation>
    </xsd:annotation>
  <xsd:complexContent mixed="true">
    <xsd:extension base="ANY">
      <xsd:sequence>
        <xsd:choice minOccurs="0"
maxOccurs="unbounded">
          <xsd:element
name="delimiter">
            <xsd:complexType
mixed="true">
```

```
7618
7619     <xsd:complexContent mixed="true">
7620
7621     <xsd:restriction base="ADXP">
7622
7623     <xsd:attribute name="partType"
7624
7625         type="cs_AddressPartType" fixed="DEL"/>
7626
7627     </xsd:restriction>
7628
7629     </xsd:complexContent>
7630
7631     </xsd:complexType>
7632
7633                                     </xsd:element>
7634     name="country">
7635                                     <xsd:element
7636     mixed="true">
7637                                     <xsd:complexType
7638
7639         <xsd:complexContent mixed="true">
7640
7641         <xsd:restriction base="ADXP">
7642
7643         <xsd:attribute name="partType"
7644
7645             type="cs_AddressPartType" fixed="CNT"/>
7646
7647         </xsd:restriction>
7648
7649         </xsd:complexContent>
7650
7651         </xsd:complexType>
7652
7653                                     </xsd:element>
7654     name="state">
7655                                     <xsd:element
7656     mixed="true">
7657                                     <xsd:complexType
7658
7659         <xsd:complexContent mixed="true">
7660
7661         <xsd:restriction base="ADXP">
7662
7663         <xsd:attribute name="partType"
7664
7665             type="cs_AddressPartType" fixed="STA"/>
7666
7667         </xsd:restriction>
7668
7669         </xsd:complexContent>
7670
7671         </xsd:complexType>
7672
7673                                     </xsd:element>
7674     name="county">
7675                                     <xsd:element
7676     mixed="true">
7677                                     <xsd:complexType
7678
7679         <xsd:complexContent mixed="true">
7680
7681         <xsd:restriction base="ADXP">
7682
7683         <xsd:attribute name="partType"
7684
7685             type="cs_AddressPartType" fixed="CPA"/>
7686
7687         </xsd:restriction>
7688
7689         </xsd:complexContent>
```

```
7687
7688     </xsd:complexType>
7689
7690                                     </xsd:element>
7691                                     <xsd:element name="city">
7692                                     <xsd:complexType
mixed="true">
7693
7694         <xsd:complexContent mixed="true">
7695         <xsd:restriction base="ADXP">
7696         <xsd:attribute name="partType"
7697         <xsd:attribute name="partType"
7698         <xsd:attribute name="partType"
7699         type="cs_AddressPartType" fixed="CTY"/>
7700
7701         </xsd:restriction>
7702
7703     </xsd:complexContent>
7704
7705 </xsd:complexType>
7706
7707                                     </xsd:element>
7708                                     <xsd:element
name="postalCode">
7709
7710                                     <xsd:complexType
mixed="true">
7711
7712         <xsd:complexContent mixed="true">
7713         <xsd:restriction base="ADXP">
7714         <xsd:restriction base="ADXP">
7715         <xsd:attribute name="partType"
7716         <xsd:attribute name="partType"
7717         <xsd:attribute name="partType"
7718         type="cs_AddressPartType" fixed="ZIP"/>
7719
7720         </xsd:restriction>
7721
7722     </xsd:complexContent>
7723
7724 </xsd:complexType>
7725
7726                                     </xsd:element>
7727                                     <xsd:element
name="streetAddressLine">
7728
7729                                     <xsd:complexType
mixed="true">
7730
7731         <xsd:complexContent mixed="true">
7732         <xsd:restriction base="ADXP">
7733         <xsd:restriction base="ADXP">
7734         <xsd:attribute name="partType"
7735         <xsd:attribute name="partType"
7736         <xsd:attribute name="partType"
7737         type="cs_AddressPartType" fixed="SAL"/>
7738
7739         </xsd:restriction>
7740
7741     </xsd:complexContent>
7742
7743 </xsd:complexType>
7744
7745                                     </xsd:element>
7746                                     <xsd:element
name="houseNumber">
7747
7748                                     <xsd:complexType
mixed="true">
7749
7750         <xsd:complexContent mixed="true">
7751         <xsd:restriction base="ADXP">
7752         <xsd:restriction base="ADXP">
7753         <xsd:attribute name="partType"
7754         <xsd:attribute name="partType"
7755         <xsd:attribute name="partType"
```



```
7756         type="cs_AddressPartType" fixed="BNR"/>
7757     </xsd:restriction>
7758 </xsd:complexContent>
7759 </xsd:complexType>
7760                                     </xsd:element>
7761                                     <xsd:element
7762 name="houseNumberNumeric">
7763                                     <xsd:complexType
7764 mixed="true">
7765         <xsd:complexContent mixed="true">
7766         <xsd:restriction base="ADXP">
7767         <xsd:attribute name="partType"
7768         type="cs_AddressPartType" fixed="BNN"/>
7769         </xsd:restriction>
7770         </xsd:complexContent>
7771         </xsd:complexType>
7772                                     </xsd:element>
7773                                     <xsd:element
7774 name="direction">
7775                                     <xsd:complexType
7776 mixed="true">
7777         <xsd:complexContent mixed="true">
7778         <xsd:restriction base="ADXP">
7779         <xsd:attribute name="partType"
7780         type="cs_AddressPartType" fixed="DIR"/>
7781         </xsd:restriction>
7782         </xsd:complexContent>
7783         </xsd:complexType>
7784                                     </xsd:element>
7785                                     <xsd:element
7786 name="streetName">
7787                                     <xsd:complexType
7788 mixed="true">
7789         <xsd:complexContent mixed="true">
7790         <xsd:restriction base="ADXP">
7791         <xsd:attribute name="partType"
7792         type="cs_AddressPartType" fixed="STR"/>
7793         </xsd:restriction>
7794         </xsd:complexContent>
7795         </xsd:complexType>
7796                                     </xsd:element>
7797                                     <xsd:element
7798 name="streetNameBase">
7799                                     <xsd:complexType
7800 mixed="true">
```

```
7826
7827     <xsd:complexContent mixed="true">
7828
7829         <xsd:restriction base="ADXP">
7830
7831             <xsd:attribute name="partType"
7832
7833                 type="cs_AddressPartType" fixed="STB"/>
7834
7835         </xsd:restriction>
7836
7837     </xsd:complexContent>
7838
7839 </xsd:complexType>
7840
7841                                     </xsd:element>
7842                                     <xsd:element
7843                                     <xsd:complexType
7844                                     mixed="true">
7845
7846         <xsd:complexContent mixed="true">
7847
7848             <xsd:restriction base="ADXP">
7849
7850                 <xsd:attribute name="partType"
7851
7852                     type="cs_AddressPartType" fixed="STYP"/>
7853
7854             </xsd:restriction>
7855
7856         </xsd:complexContent>
7857
7858     </xsd:complexType>
7859
7860                                     </xsd:element>
7861                                     <xsd:element
7862                                     <xsd:complexType
7863                                     mixed="true">
7864
7865         <xsd:complexContent mixed="true">
7866
7867             <xsd:restriction base="ADXP">
7868
7869                 <xsd:attribute name="partType"
7870
7871                     type="cs_AddressPartType" fixed="ADL"/>
7872
7873             </xsd:restriction>
7874
7875         </xsd:complexContent>
7876
7877     </xsd:complexType>
7878
7879                                     </xsd:element>
7880                                     <xsd:element
7881                                     <xsd:complexType
7882                                     mixed="true">
7883
7884         <xsd:complexContent mixed="true">
7885
7886             <xsd:restriction base="ADXP">
7887
7888                 <xsd:attribute name="partType"
7889
7890                     type="cs_AddressPartType" fixed="UNID"/>
7891
7892             </xsd:restriction>
7893
7894         </xsd:complexContent>
```

```
7895     </xsd:complexType>
7896
7897     </xsd:element>
7898     <xsd:element
7899     name="unitType">
7900     <xsd:complexType
7901     mixed="true">
7902
7903         <xsd:complexContent mixed="true">
7904
7905             <xsd:restriction base="ADXP">
7906
7907                 <xsd:attribute name="partType"
7908
7909                 type="cs_AddressPartType" fixed="UNIT"/>
7910
7911             </xsd:restriction>
7912
7913         </xsd:complexContent>
7914
7915     </xsd:complexType>
7916
7917     </xsd:element>
7918     <xsd:element
7919     name="carrier">
7920     <xsd:complexType
7921     mixed="true">
7922
7923         <xsd:complexContent mixed="true">
7924
7925             <xsd:restriction base="ADXP">
7926
7927                 <xsd:attribute name="partType"
7928
7929                 type="cs_AddressPartType" fixed="CAR"/>
7930
7931             </xsd:restriction>
7932
7933         </xsd:complexContent>
7934
7935     </xsd:complexType>
7936
7937     </xsd:element>
7938     <xsd:element
7939     name="censusTract">
7940     <xsd:complexType
7941     mixed="true">
7942
7943         <xsd:complexContent mixed="true">
7944
7945             <xsd:restriction base="ADXP">
7946
7947                 <xsd:attribute name="partType"
7948
7949                 type="cs_AddressPartType" fixed="CEN"/>
7950
7951             </xsd:restriction>
7952
7953         </xsd:complexContent>
7954
7955     </xsd:complexType>
7956
7957     </xsd:element>
7958     </xsd:choice>
7959     <xsd:element
7960     name="useablePeriod" type="SXCM_TS" minOccurs="0"
7961
7962     maxOccurs="unbounded">
7963     <xsd:annotation>
7964
7965         <xsd:documentation> A General Timing Specification (GTS)
7966
7967         specifying
7968         the periods of time during which the address can
```

be used.
This is used to specify different addresses for
different
times of the year or to refer to historical
addresses.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="use"
type="set_cs_PostalAddressUse" use="optional">
<xsd:annotation>
<xsd:documentation> A set
of codes advising a system or user which
address in a set
of like addresses to select for a given
purpose.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="isNotOrdered"
type="bl" use="optional">
<xsd:annotation>
<xsd:documentation> A
boolean value specifying whether the order of
the address parts
is known or not. While the address parts are
always a
Sequence, the order in which they are presented may or
may not be known.
Where this matters, the isNotOrdered property
can be used to
convey this information. </xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="cs_EntityNamePartQualifier"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron">
<xsd:restriction base="cs">
<xsd:enumeration value="BR"/>
<xsd:enumeration value="SP"/>
<xsd:enumeration value="VV"/>
<xsd:enumeration value="AC"/>
<xsd:enumeration value="PR"/>
<xsd:enumeration value="NB"/>
<xsd:enumeration value="LS"/>
<xsd:enumeration value="CL"/>
<xsd:enumeration value="IN"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ENXP" mixed="true"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron">
<xsd:annotation>
<xsd:documentation> A character string token
representing a part of a name. May
have a type code signifying the role of
the part in the whole entity name,
and a qualifier code for more detail
about the name part type. Typical name
parts for person names are given names,
and family names, titles,
etc.</xsd:documentation>

```
8034         </xsd:annotation>
8035         <xsd:complexContent mixed="true">
8036             <xsd:extension base="ST">
8037                 <xsd:attribute name="partType"
8038 type="cs_EntityNamePartType">
8039                     <xsd:annotation>
8040                         <xsd:documentation>
8041 Indicates whether the name part is a given name,
8042                                     family name,
8043 prefix, suffix, etc.</xsd:documentation>
8044                     </xsd:annotation>
8045                 </xsd:attribute>
8046                 <xsd:attribute name="qualifier"
8047 type="set_cs_EntityNamePartQualifier"
8048                     use="optional">
8049                     <xsd:annotation>
8050                         <xsd:documentation> The
8051 qualifier is a set of codes each of which
8052                                     specifies a
8053 certain subcategory of the name part in addition to
8054                                     the main name
8055 part type. For example, a given name may be
8056                                     flagged as a
8057 nickname, a family name may be a pseudonym or a
8058                                     name of public
8059 records.</xsd:documentation>
8060                     </xsd:annotation>
8061                 </xsd:attribute>
8062             </xsd:extension>
8063         </xsd:complexContent>
8064     </xsd:complexType>
8065     <xsd:simpleType name="cs_EntityNamePartType"
8066 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8067
8068     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8069     xmlns:sch="http://www.ascc.net/xml/schematron">
8070     <xsd:restriction base="cs">
8071         <xsd:enumeration value="FAM"/>
8072         <xsd:enumeration value="GIV"/>
8073         <xsd:enumeration value="PFX"/>
8074         <xsd:enumeration value="SFX"/>
8075         <xsd:enumeration value="DEL"/>
8076     </xsd:restriction>
8077 </xsd:simpleType>
8078 <xsd:complexType name="en.delimiter" mixed="true"
8079     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8080
8081     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8082     xmlns:sch="http://www.ascc.net/xml/schematron">
8083 <xsd:complexContent mixed="true">
8084     <xsd:restriction base="ENXP">
8085         <xsd:attribute name="partType"
8086 type="cs_EntityNamePartType" fixed="DEL"/>
8087     </xsd:restriction>
8088 </xsd:complexContent>
8089 </xsd:complexType>
8090 <xsd:complexType name="en.family" mixed="true"
8091     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8092
8093     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8094     xmlns:sch="http://www.ascc.net/xml/schematron">
8095 <xsd:complexContent mixed="true">
8096     <xsd:restriction base="ENXP">
8097         <xsd:attribute name="partType"
8098 type="cs_EntityNamePartType" fixed="FAM"/>
8099     </xsd:restriction>
8100 </xsd:complexContent>
8101 </xsd:complexType>
8102 <xsd:complexType name="en.given" mixed="true"
8103     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
```

```
8104      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8105      xmlns:sch="http://www.ascc.net/xml/schematron">
8106      <xsd:complexContent mixed="true">
8107          <xsd:restriction base="ENXP">
8108              <xsd:attribute name="partType"
8109 type="cs_EntityNamePartType" fixed="GIV"/>
8110          </xsd:restriction>
8111      </xsd:complexContent>
8112  </xsd:complexType>
8113  <xsd:complexType name="en.prefix" mixed="true"
8114      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8115      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8116      xmlns:sch="http://www.ascc.net/xml/schematron">
8117      <xsd:complexContent mixed="true">
8118          <xsd:restriction base="ENXP">
8119              <xsd:attribute name="partType"
8120 type="cs_EntityNamePartType" fixed="PFX"/>
8121          </xsd:restriction>
8122      </xsd:complexContent>
8123  </xsd:complexType>
8124  <xsd:complexType name="en.suffix" mixed="true"
8125      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8126      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8127      xmlns:sch="http://www.ascc.net/xml/schematron">
8128      <xsd:complexContent mixed="true">
8129          <xsd:restriction base="ENXP">
8130              <xsd:attribute name="partType"
8131 type="cs_EntityNamePartType" fixed="SFX"/>
8132          </xsd:restriction>
8133      </xsd:complexContent>
8134  </xsd:complexType>
8135  <xsd:simpleType name="cs_EntityNameUse"
8136      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8137      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8138      xmlns:sch="http://www.ascc.net/xml/schematron">
8139      <xsd:restriction base="cs">
8140          <xsd:enumeration value="L"/>
8141          <xsd:enumeration value="A"/>
8142          <xsd:enumeration value="I"/>
8143          <xsd:enumeration value="R"/>
8144          <xsd:enumeration value="ABC"/>
8145          <xsd:enumeration value="SYL"/>
8146          <xsd:enumeration value="IDE"/>
8147      </xsd:restriction>
8148  </xsd:simpleType>
8149  <xsd:complexType name="EN" mixed="true"
8150      xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8151      xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8152      xmlns:sch="http://www.ascc.net/xml/schematron">
8153      <xsd:annotation>
8154          <xsd:documentation> A name for a person,
8155 organization, place or thing. A
8156 sequence of name parts, such as given
8157 name or family name, prefix, suffix,
8158 etc. Examples for entity name values are
8159 "Jim Bob Walton, Jr.", "Health
8160 Level Seven, Inc.", "Lake Tahoe", etc.
8161 An entity name may be as simple as a
8162 character string or may consist of
8163 several entity name parts, such as,
8164 "Jim", "Bob", "Walton", and "Jr.",
8165 "Health Level Seven" and "Inc.", "Lake
8166 and "Tahoe".</xsd:documentation>
8167 </xsd:annotation>
8168 <xsd:complexContent mixed="true">
```

```
8174         <xsd:extension base="ANY">
8175             <xsd:sequence>
8176                 <xsd:choice minOccurs="0"
8177 maxOccurs="unbounded">
8178                     <xsd:element
8179 name="delimiter" type="en.delimiter"/>
8180                     <xsd:element
8181 name="family" type="en.family"/>
8182                     <xsd:element name="given"
8183 type="en.given"/>
8184                     <xsd:element
8185 name="prefix" type="en.prefix"/>
8186                     <xsd:element
8187 name="suffix" type="en.suffix"/>
8188                 </xsd:choice>
8189                 <xsd:element name="validTime"
8190 type="IVL_TS" minOccurs="0">
8191                     <xsd:annotation>
8192
8193         <xsd:documentation> An interval of time specifying the time
8194                                     during
8195 which the name is or was used for the entity. This
8196
8197         accomodates the fact that people change names for people,
8198                                     places and
8199 things. </xsd:documentation>
8200                     </xsd:annotation>
8201                 </xsd:element>
8202             </xsd:sequence>
8203             <xsd:attribute name="use"
8204 type="set_cs_EntityNameUse" use="optional">
8205                 <xsd:annotation>
8206                     <xsd:documentation> A set
8207 of codes advising a system or user which
8208                                     name in a set of
8209 like names to select for a given purpose. A
8210                                     name without
8211 specific use code might be a default name useful
8212                                     for any purpose,
8213 but a name with a specific use code would be
8214                                     preferred for
8215 that respective purpose.</xsd:documentation>
8216                     </xsd:annotation>
8217                 </xsd:attribute>
8218             </xsd:extension>
8219         </xsd:complexContent>
8220     </xsd:complexType>
8221     <xsd:complexType name="PN" mixed="true"
8222 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8223
8224     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8225     xmlns:sch="http://www.ascc.net/xml/schematron">
8226         <xsd:annotation>
8227             <xsd:documentation> A name for a person. A
8228 sequence of name parts, such as given
8229                                     name or family name, prefix, suffix,
8230 etc. PN differs from EN because the
8231                                     qualifier type cannot include LS (Legal
8232 Status).</xsd:documentation>
8233             </xsd:annotation>
8234             <xsd:complexContent mixed="true">
8235                 <xsd:extension base="EN"/>
8236             </xsd:complexContent>
8237         </xsd:complexType>
8238         <xsd:complexType name="ON" mixed="true"
8239 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8240
8241     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8242     xmlns:sch="http://www.ascc.net/xml/schematron">
8243             <xsd:annotation>
```



```
8244         <xsd:documentation> A name for an organization.
8245 A sequence of name
8246         parts.</xsd:documentation>
8247     </xsd:annotation>
8248     <xsd:complexContent mixed="true">
8249         <xsd:restriction base="EN">
8250             <xsd:sequence>
8251                 <xsd:choice minOccurs="0"
8252 maxOccurs="unbounded">
8253                     <xsd:element
8254 name="delimiter" type="en.delimiter"/>
8255                     <xsd:element
8256 name="prefix" type="en.prefix"/>
8257                     <xsd:element
8258 name="suffix" type="en.suffix"/>
8259                 </xsd:choice>
8260                 <xsd:element name="validTime"
8261 type="IVL_TS" minOccurs="0">
8262                     <xsd:annotation>
8263         <xsd:documentation> An interval of time specifying the time
8264 during
8265 which the name is or was used for the entity. This
8266 accomodates the fact that people change names for people,
8267 places and
8268 things. </xsd:documentation>
8269                     </xsd:annotation>
8270                 </xsd:element>
8271             </xsd:sequence>
8272             <xsd:attribute name="use"
8273 type="set_cs_EntityNameUse" use="optional">
8274                 <xsd:annotation>
8275         <xsd:documentation> A set
8276 of codes advising a system or user which
8277 name in a set of
8278 like names to select for a given purpose. A
8279 name without
8280 specific use code might be a default name useful
8281 for any purpose,
8282 but a name with a specific use code would be
8283 preferred for
8284 that respective purpose.</xsd:documentation>
8285                 </xsd:annotation>
8286             </xsd:attribute>
8287         </xsd:restriction>
8288     </xsd:complexContent>
8289     </xsd:complexType>
8290     <xsd:complexType name="TN" mixed="true"
8291 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8292     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8293     xmlns:sch="http://www.ascc.net/xml/schematron">
8294         <xsd:annotation>
8295             <xsd:documentation> A restriction of entity
8296 name that is effectively a simple
8297 string used for a simple name for things
8298 and places.</xsd:documentation>
8299         </xsd:annotation>
8300         <xsd:complexContent mixed="true">
8301             <xsd:restriction base="EN">
8302                 <xsd:sequence/>
8303             </xsd:restriction>
8304         </xsd:complexContent>
8305     </xsd:complexType>
8306     <xsd:complexType name="QTY" abstract="true"
8307 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8308     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8309     xmlns:sch="http://www.ascc.net/xml/schematron">
```

```
8314         <xsd:annotation>
8315             <xsd:documentation> The quantity data type is
8316 an abstract generalization for all
8317                                     data types (1) whose value set has an
8318 order relation (less-or-equal) and (2)
8319                                     where difference is defined in all of
8320 the data type's totally ordered value
8321                                     subsets. The quantity type abstraction
8322 is needed in defining certain other
8323                                     types, such as the interval and the
8324 probability
8325                                     distribution.</xsd:documentation>
8326         </xsd:annotation>
8327         <xsd:complexContent>
8328             <xsd:extension base="ANY"/>
8329         </xsd:complexContent>
8330     </xsd:complexType>
8331     <xsd:simpleType name="int"
8332 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8333
8334         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8335         xmlns:sch="http://www.ascc.net/xml/schematron">
8336         <xsd:annotation>
8337             <xsd:documentation> Integer numbers (-1,0,1,2,
8338 100, 3398129, etc.) are precise
8339                                     numbers that are results of counting and
8340 enumerating. Integer numbers are
8341                                     discrete, the set of integers is
8342 infinite but countable. No arbitrary limit
8343                                     is imposed on the range of integer
8344 numbers. Two NULL flavors are defined for
8345                                     the positive and negative
8346 infinity.</xsd:documentation>
8347         </xsd:annotation>
8348         <xsd:restriction base="xsd:integer"/>
8349     </xsd:simpleType>
8350     <xsd:complexType name="INT"
8351 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8352
8353         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8354         xmlns:sch="http://www.ascc.net/xml/schematron">
8355         <xsd:annotation>
8356             <xsd:documentation> Integer numbers (-1,0,1,2,
8357 100, 3398129, etc.) are precise
8358                                     numbers that are results of counting and
8359 enumerating. Integer numbers are
8360                                     discrete, the set of integers is
8361 infinite but countable. No arbitrary limit
8362                                     is imposed on the range of integer
8363 numbers. Two NULL flavors are defined for
8364                                     the positive and negative
8365 infinity.</xsd:documentation>
8366         </xsd:annotation>
8367         <xsd:complexContent>
8368             <xsd:extension base="QTY">
8369                 <xsd:attribute name="value" type="int"
8370 use="optional"/>
8371             </xsd:extension>
8372         </xsd:complexContent>
8373     </xsd:complexType>
8374     <xsd:simpleType name="real"
8375 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8376
8377         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8378         xmlns:sch="http://www.ascc.net/xml/schematron">
8379         <xsd:annotation>
8380             <xsd:documentation> Fractional numbers.
8381 Typically used whenever quantities are
8382                                     measured, estimated, or computed from
8383 other real numbers. The typical
```

```
8384                                     representation is decimal, where the
8385 number of significant decimal digits is
8386                                     known as the precision. Real numbers are
8387 needed beyond integers whenever
8388                                     quantities of the real world are
8389 measured, estimated, or computed from other
8390                                     real numbers. The term "Real number" in
8391 this specification is used to mean
8392                                     that fractional values are covered
8393 without necessarily implying the full set
8394                                     of the mathematical real
8395 numbers.</xsd:documentation>
8396                                     </xsd:annotation>
8397                                     <xsd:union memberTypes="xsd:decimal xsd:double"/>
8398                                     </xsd:simpleType>
8399                                     <xsd:complexType name="REAL"
8400 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8401
8402         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8403         xmlns:sch="http://www.ascc.net/xml/schematron">
8404         <xsd:annotation>
8405         <xsd:documentation> Fractional numbers.
8406 Typically used whenever quantities are
8407                                     measured, estimated, or computed from
8408 other real numbers. The typical
8409                                     representation is decimal, where the
8410 number of significant decimal digits is
8411                                     known as the precision. Real numbers are
8412 needed beyond integers whenever
8413                                     quantities of the real world are
8414 measured, estimated, or computed from other
8415                                     real numbers. The term "Real number" in
8416 this specification is used to mean
8417                                     that fractional values are covered
8418 without necessarily implying the full set
8419                                     of the mathematical real
8420 numbers.</xsd:documentation>
8421                                     </xsd:annotation>
8422                                     <xsd:complexContent>
8423                                     <xsd:extension base="QTY">
8424                                     <xsd:attribute name="value" type="real"
8425 use="optional"/>
8426                                     </xsd:extension>
8427                                     </xsd:complexType>
8428                                     <xsd:complexType name="PQR"
8429 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8430
8431         xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8432         xmlns:sch="http://www.ascc.net/xml/schematron">
8433         <xsd:annotation>
8434         <xsd:documentation> A representation of a
8435 physical quantity in a unit from any
8436                                     code system. Used to show alternative
8437 representation for a physical
8438                                     quantity.</xsd:documentation>
8439                                     </xsd:annotation>
8440                                     <xsd:complexContent>
8441                                     <xsd:extension base="CV">
8442                                     <xsd:attribute name="value" type="real"
8443 use="optional">
8444                                     <xsd:annotation>
8445                                     <xsd:documentation> The
8446 magnitude of the measurement value in terms
8447                                     of the unit
8448 specified in the code.</xsd:documentation>
8449                                     </xsd:annotation>
8450                                     </xsd:attribute>
8451                                     </xsd:extension>
8452                                     </xsd:complexContent>
```

```
8454         </xsd:complexType>
8455         <xsd:complexType name="PQ"
8456 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8457
8458     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8459     xmlns:sch="http://www.ascc.net/xml/schematron">
8460         <xsd:annotation>
8461             <xsd:documentation> A dimensioned quantity
8462 expressing the result of a
8463             measurement act. </xsd:documentation>
8464         </xsd:annotation>
8465         <xsd:complexContent>
8466             <xsd:extension base="QTY">
8467                 <xsd:sequence>
8468                     <xsd:element name="translation"
8469 type="PQR" minOccurs="0"
8470
8471                     maxOccurs="unbounded">
8472                 <xsd:annotation>
8473
8474                 <xsd:documentation> An alternative representation of the same
8475 quantity expressed in a different unit, of a
8476                                     physical
8477                                     different
8478 unit code system and possibly with a different
8479 value.</xsd:documentation>
8480             </xsd:annotation>
8481         </xsd:element>
8482     </xsd:sequence>
8483     <xsd:attribute name="value" type="real"
8484 use="optional">
8485         <xsd:annotation>
8486             <xsd:documentation> The
8487 magnitude of the quantity measured in terms
8488                                     of the
8489 unit.</xsd:documentation>
8490         </xsd:annotation>
8491     </xsd:attribute>
8492     <xsd:attribute name="unit" type="cs"
8493 use="optional" default="1">
8494         <xsd:annotation>
8495             <xsd:documentation> The
8496 unit of measure specified in the Unified
8497                                     Code for Units of
8498 Measure (UCUM)
8499 [http://aurora.rg.iupui.edu/UCUM].</xsd:documentation>
8500         </xsd:annotation>
8501     </xsd:attribute>
8502 </xsd:extension>
8503 </xsd:complexContent>
8504 </xsd:complexType>
8505 <xsd:complexType name="MO"
8506 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8507
8508     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8509     xmlns:sch="http://www.ascc.net/xml/schematron">
8510         <xsd:annotation>
8511             <xsd:documentation> A monetary amount is a
8512 quantity expressing the amount of
8513                                     money in some currency. Currencies are
8514 the units in which monetary amounts
8515                                     are denominated in different economic
8516 regions. While the monetary amount is
8517                                     a single kind of quantity (money) the
8518 exchange rates between the different
8519                                     units are variable. This is the
8520 principle difference between physical
8521                                     quantity and monetary amounts, and the
8522 reason why currency units are not
```

```
physical units.</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
  <xsd:extension base="QTY">
    <xsd:attribute name="value" type="real"
use="optional">
      <xsd:annotation>
        <xsd:documentation> The
magnitude of the monetary amount in terms of
      the currency
unit.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="currency" type="cs"
use="optional">
      <xsd:annotation>
        <xsd:documentation> The
currency unit as defined in ISO
      4217.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RTO"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
  xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron">
  <xsd:annotation>
    <xsd:documentation> A quantity constructed as
the quotient of a numerator
    quantity divided by a denominator
quantity. Common factors in the numerator
    and denominator are not automatically
cancelled out. The data type supports
    titers (e.g., "1:128") and other
quantities produced by laboratories that
    truly represent ratios. Ratios are not
simply "structured numerics",
    particularly blood pressure measurements
(e.g. "120/60") are not ratios. In
    many cases the REAL should be used
instead of the . </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="RTO_QTY_QTY"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="cs_SetOperator"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
  xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron">
  <xsd:restriction base="cs">
    <xsd:enumeration value="I"/>
    <xsd:enumeration value="E"/>
    <xsd:enumeration value="A"/>
    <xsd:enumeration value="H"/>
    <xsd:enumeration value="P"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="cs_CalendarCycle"
xmlns:xlink="http://www.w3.org/TR/WD-xlink"
  xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron">
  <xsd:restriction base="cs">
    <xsd:enumeration value="CY"/>
    <xsd:enumeration value="MY"/>
```

```
8594         <xsd:enumeration value="CM"/>
8595         <xsd:enumeration value="CW"/>
8596         <xsd:enumeration value="WY"/>
8597         <xsd:enumeration value="DM"/>
8598         <xsd:enumeration value="CD"/>
8599         <xsd:enumeration value="DY"/>
8600         <xsd:enumeration value="DW"/>
8601         <xsd:enumeration value="HD"/>
8602         <xsd:enumeration value="CH"/>
8603         <xsd:enumeration value="NH"/>
8604         <xsd:enumeration value="CN"/>
8605         <xsd:enumeration value="SN"/>
8606         <xsd:enumeration value="CS"/>
8607     </xsd:restriction>
8608 </xsd:simpleType>
8609 <xsd:simpleType name="cs_TimingEvent"
8610 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8611
8612     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8613     xmlns:sch="http://www.ascc.net/xml/schematron">
8614     <xsd:restriction base="cs">
8615         <xsd:enumeration value="AC"/>
8616         <xsd:enumeration value="ACD"/>
8617         <xsd:enumeration value="ACM"/>
8618         <xsd:enumeration value="ACV"/>
8619         <xsd:enumeration value="HS"/>
8620         <xsd:enumeration value="IC"/>
8621         <xsd:enumeration value="ICD"/>
8622         <xsd:enumeration value="ICM"/>
8623         <xsd:enumeration value="ICV"/>
8624         <xsd:enumeration value="PC"/>
8625         <xsd:enumeration value="PCD"/>
8626         <xsd:enumeration value="PCM"/>
8627         <xsd:enumeration value="PCV"/>
8628     </xsd:restriction>
8629 </xsd:simpleType>
8630 <xsd:simpleType name="cs_ProbabilityDistributionType"
8631     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8632
8633     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8634     xmlns:sch="http://www.ascc.net/xml/schematron">
8635     <xsd:restriction base="cs">
8636         <xsd:enumeration value="U"/>
8637         <xsd:enumeration value="N"/>
8638         <xsd:enumeration value="LN"/>
8639         <xsd:enumeration value="G"/>
8640         <xsd:enumeration value="E"/>
8641         <xsd:enumeration value="X2"/>
8642         <xsd:enumeration value="T"/>
8643         <xsd:enumeration value="F"/>
8644         <xsd:enumeration value="B"/>
8645     </xsd:restriction>
8646 </xsd:simpleType>
8647 <xsd:complexType name="SXCM_TS"
8648 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8649
8650     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8651     xmlns:sch="http://www.ascc.net/xml/schematron">
8652     <xsd:complexContent>
8653         <xsd:extension base="TS">
8654             <xsd:attribute name="operator"
8655 type="cs_SetOperator" use="optional"
8656
8657                 default="I">
8658                 <xsd:annotation>
8659                     <xsd:documentation> A
8660 code specifying whether the set component is
8661                                     included (union)
8662 or excluded (set-difference) from the set, or
8663                                     other set
8664 operations with the current set component and the set
```

```
8664 from the representation stream up to the current as constructed
8665
8666 point.</xsd:documentation>
8667
8668                                     </xsd:annotation>
8669                                     </xsd:attribute>
8670                                     </xsd:extension>
8671                                     </xsd:complexContent>
8672                                     </xsd:complexType>
8673                                     <xsd:simpleType name="set_cs_TelecommunicationAddressUse"
8674                                     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8675
8676                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8677                                     xmlns:sch="http://www.ascc.net/xml/schematron">
8678                                     <xsd:list itemType="cs_TelecommunicationAddressUse"/>
8679                                     </xsd:simpleType>
8680                                     <xsd:simpleType name="set_cs_PostalAddressUse"
8681                                     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8682
8683                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8684                                     xmlns:sch="http://www.ascc.net/xml/schematron">
8685                                     <xsd:list itemType="cs_PostalAddressUse"/>
8686                                     </xsd:simpleType>
8687                                     <xsd:simpleType name="set_cs_EntityNamePartQualifier"
8688                                     xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8689
8690                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8691                                     xmlns:sch="http://www.ascc.net/xml/schematron">
8692                                     <xsd:list itemType="cs_EntityNamePartQualifier"/>
8693                                     </xsd:simpleType>
8694                                     <xsd:complexType name="IVL_TS"
8695 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8696
8697                                     xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8698                                     xmlns:sch="http://www.ascc.net/xml/schematron">
8699                                     <xsd:complexContent>
8700                                     <xsd:extension base="SXCM_TS">
8701                                     <xsd:choice minOccurs="0">
8702                                     <xsd:sequence>
8703                                     <xsd:element name="low"
8704 type="IVXB_TS">
8705                                     <xsd:annotation>
8706
8707                                     <xsd:documentation> The low limit of the
8708
8709                                     interval.</xsd:documentation>
8710                                     </xsd:annotation>
8711                                     </xsd:element>
8712                                     <xsd:choice
8713 minOccurs="0">
8714                                     <xsd:element
8715 name="width" type="PQ" minOccurs="0">
8716                                     <xsd:annotation>
8717
8718                                     <xsd:documentation> The difference between high and low
8719
8720                                     boundary. The purpose of distinguishing a width
8721
8722                                     property is to handle all cases of incomplete
8723
8724                                     information symmetrically. In any interval
8725
8726                                     representation only two of the three properties
8727
8728                                     high, low, and width need to be stated and the third
8729
8730                                     can be derived.</xsd:documentation>
8731                                     </xsd:annotation>
```



```
8734                                     </xsd:element>
8735                                     <xsd:element
8736 name="high" type="IVXB_TS" minOccurs="0">
8737
8738     <xsd:annotation>
8739
8740     <xsd:documentation> The high limit of the
8741
8742     interval.</xsd:documentation>
8743
8744     </xsd:annotation>
8745                                     </xsd:element>
8746                                     </xsd:choice>
8747                                     </xsd:sequence>
8748                                     <xsd:element name="high"
8749 type="IVXB_TS">
8750                                     <xsd:annotation>
8751
8752     <xsd:documentation/>
8753                                     </xsd:annotation>
8754                                     </xsd:element>
8755                                     <xsd:sequence>
8756                                     <xsd:element name="width"
8757 type="PQ">
8758                                     <xsd:annotation>
8759
8760     <xsd:documentation> The difference between high and low
8761
8762     boundary. The purpose of distinguishing a width property
8763                                     is
8764 to handle all cases of incomplete information
8765
8766     symmetrically. In any interval representation only two
8767                                     of
8768 the three properties high, low, and width need to be
8769
8770     stated and the third can be derived.</xsd:documentation>
8771                                     </xsd:annotation>
8772                                     </xsd:element>
8773                                     <xsd:element name="high"
8774 type="IVXB_TS" minOccurs="0">
8775                                     <xsd:annotation>
8776
8777     <xsd:documentation> The high limit of the
8778
8779     interval.</xsd:documentation>
8780                                     </xsd:annotation>
8781                                     </xsd:element>
8782                                     </xsd:sequence>
8783                                     <xsd:sequence>
8784                                     <xsd:element
8785 name="center" type="TS">
8786                                     <xsd:annotation>
8787
8788     <xsd:documentation> The arithmetic mean of the interval (low
8789
8790     plus high divided by 2). The purpose of distinguishing
8791
8792     the center as a semantic property is for conversions of
8793
8794     intervals from and to point values.</xsd:documentation>
8795                                     </xsd:annotation>
8796                                     </xsd:element>
8797                                     <xsd:element name="width"
8798 type="PQ" minOccurs="0">
8799                                     <xsd:annotation>
8800
8801     <xsd:documentation> The difference between high and low
8802
8803     boundary. The purpose of distinguishing a width property
```



```
8804 is
8805 to handle all cases of incomplete information
8806
8807 symmetrically. In any interval representation only two
8808 of
8809 the three properties high, low, and width need to be
8810
8811 stated and the third can be derived.</xsd:documentation>
8812 </xsd:annotation>
8813 </xsd:element>
8814 </xsd:sequence>
8815 </xsd:choice>
8816 </xsd:extension>
8817 </xsd:complexContent>
8818 </xsd:complexType>
8819 <xsd:complexType name="IVXB_TS"
8820 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8821
8822 xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8823 xmlns:sch="http://www.ascc.net/xml/schematron">
8824 <xsd:complexContent>
8825 <xsd:extension base="TS">
8826 <xsd:attribute name="inclusive"
8827 type="bl" use="optional" default="true">
8828 <xsd:annotation>
8829 <xsd:documentation>
8830 Specifies whether the limit is included in the
8831 interval
8832 (interval is closed) or excluded from the interval
8833 (interval is
8834 open).</xsd:documentation>
8835 </xsd:annotation>
8836 </xsd:attribute>
8837 </xsd:extension>
8838 </xsd:complexContent>
8839 </xsd:complexType>
8840 <xsd:simpleType name="set_cs_EntityNameUse"
8841 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8842
8843 xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8844 xmlns:sch="http://www.ascc.net/xml/schematron">
8845 <xsd:list itemType="cs_EntityNameUse"/>
8846 </xsd:simpleType>
8847 <xsd:complexType name="RTO_QTY_QTY"
8848 xmlns:xlink="http://www.w3.org/TR/WD-xlink"
8849
8850 xmlns:gsd="http://aurora.regenstrief.org/GenericXMLSchema"
8851 xmlns:sch="http://www.ascc.net/xml/schematron">
8852 <xsd:annotation/>
8853 <xsd:complexContent>
8854 <xsd:extension base="QTY">
8855 <xsd:sequence>
8856 <xsd:element name="numerator">
8857 <xsd:annotation>
8858
8859 <xsd:documentation> The quantity that is being divided in the
8860 ratio. The
8861 default is the integer number 1
8862
8863 (one.)</xsd:documentation>
8864 </xsd:annotation>
8865 </xsd:complexType>
8866 <xsd:complexContent>
8867 <xsd:extension base="QTY">
8868
8869 <xsd:attribute name="value" default="1"/>
8870
8871 </xsd:extension>
```

```
8874 </xsd:complexContent>
8875
8876 </xsd:complexType>
8877 </xsd:element>
8878 <xsd:element name="denominator">
8879 <xsd:annotation>
8880
8881 <xsd:documentation> The quantity that devides the numerator in
8882 the ratio.
8883 The default is the integer number 1 (one.) The
8884
8885 denominator must not be zero.</xsd:documentation>
8886 </xsd:annotation>
8887 </xsd:complexType>
8888
8889 <xsd:complexContent>
8890
8891 <xsd:extension base="QTY">
8892
8893 <xsd:attribute name="value" default="1"/>
8894
8895 </xsd:extension>
8896
8897 </xsd:complexContent>
8898 </xsd:complexType>
8899 </xsd:element>
8900 </xsd:sequence>
8901 </xsd:extension>
8902 </xsd:complexContent>
8903 </xsd:complexType>
8904 </xs:schema>
8905 </wsdl:types>
8906 <wsdl:message name="MessageAckSoapIn">
8907 <wsdl:part element="ws:MessageAck" name="parameters"/>
8908 </wsdl:message>
8909 <wsdl:message name="MessageAckSoapOut">
8910 <wsdl:part element="ws:MessageAckResponse" name="parameters"/>
8911 </wsdl:message>
8912 <wsdl:message name="MessageBoxStatusSoapIn">
8913 <wsdl:part element="ws:MessageBoxStatus" name="parameters"/>
8914 </wsdl:message>
8915 <wsdl:message name="MessageBoxStatusSoapOut">
8916 <wsdl:part element="ws:MessageBoxStatusResponse" name="parameters"/>
8917 </wsdl:message>
8918 <wsdl:message name="MessageListingSoapIn">
8919 <wsdl:part element="ws:MessageListing" name="parameters"/>
8920 </wsdl:message>
8921 <wsdl:message name="MessageListingSoapOut">
8922 <wsdl:part element="ws:MessageListingResponse" name="parameters"/>
8923 </wsdl:message>
8924 <wsdl:message name="MessageReadSoapIn">
8925 <wsdl:part element="ws:MessageRead" name="parameters"/>
8926 </wsdl:message>
8927 <wsdl:message name="MessageReadSoapOut">
8928 <wsdl:part element="ws:MessageReadResponse" name="parameters"/>
8929 </wsdl:message>
8930 <wsdl:message name="MessageRecvSoapIn">
8931 <wsdl:part element="ws:MessageRecv" name="parameters"/>
8932 </wsdl:message>
8933 <wsdl:message name="MessageRecvSoapOut">
8934 <wsdl:part element="ws:MessageRecvResponse" name="parameters"/>
8935 </wsdl:message>
8936 <wsdl:message name="MessageResetSoapIn">
8937 <wsdl:part element="ws:MessageReset" name="parameters"/>
8938 </wsdl:message>
8939 <wsdl:message name="MessageResetSoapOut">
8940 <wsdl:part element="ws:MessageResetResponse" name="parameters"/>
8941 </wsdl:message>
8942 <wsdl:message name="MessageSendSoapIn">
8943 <wsdl:part element="ws:MessageSend" name="parameters"/>
```

```
8944 </wsdl:message>
8945 <wsdl:message name="MessageSendSoapOut">
8946   <wsdl:part element="ws:MessageSendResponse" name="parameters"/>
8947 </wsdl:message>
8948 <wsdl:message name="MessageTrackSoapIn">
8949   <wsdl:part element="ws:MessageTrack" name="parameters"/>
8950 </wsdl:message>
8951 <wsdl:message name="MessageTrackSoapOut">
8952   <wsdl:part element="ws:MessageTrackResponse" name="parameters"/>
8953 </wsdl:message>
8954 <wsdl:message name="OrgAdminSoapIn">
8955   <wsdl:part element="ws:OrgAdmin" name="parameters"/>
8956 </wsdl:message>
8957 <wsdl:message name="OrgAdminSoapOut">
8958   <wsdl:part element="ws:OrgAdminResponse" name="parameters"/>
8959 </wsdl:message>
8960 <wsdl:message name="OrgQuerySoapIn">
8961   <wsdl:part element="ws:OrgQuery" name="parameters"/>
8962 </wsdl:message>
8963 <wsdl:message name="OrgQuerySoapOut">
8964   <wsdl:part element="ws:OrgQueryResponse" name="parameters"/>
8965 </wsdl:message>
8966 <wsdl:message name="ProviderAdminSoapIn">
8967   <wsdl:part element="ws:ProviderAdmin" name="parameters"/>
8968 </wsdl:message>
8969 <wsdl:message name="ProviderAdminSoapOut">
8970   <wsdl:part element="ws:ProviderAdminResponse" name="parameters"/>
8971 </wsdl:message>
8972 <wsdl:message name="ProviderQuerySoapIn">
8973   <wsdl:part element="ws:ProviderQuery" name="parameters"/>
8974 </wsdl:message>
8975 <wsdl:message name="ProviderQuerySoapOut">
8976   <wsdl:part element="ws:ProviderQueryResponse" name="parameters"/>
8977 </wsdl:message>
8978 <wsdl:message name="UserQuerySoapIn">
8979   <wsdl:part element="ws:UserQuery" name="parameters"/>
8980 </wsdl:message>
8981 <wsdl:message name="UserQuerySoapOut">
8982   <wsdl:part element="ws:UserQueryResponse" name="parameters"/>
8983 </wsdl:message>
8984 <wsdl:message name="UserAdminSoapIn">
8985   <wsdl:part element="ws:UserAdmin" name="parameters"/>
8986 </wsdl:message>
8987 <wsdl:message name="UserAdminSoapOut">
8988   <wsdl:part element="ws:UserAdminResponse" name="parameters"/>
8989 </wsdl:message>
8990 <wsdl:portType name="eMSEPSOap">
8991   <wsdl:operation name="MessageAck">
8992     <wsdl:input message="ws:MessageAckSoapIn"/>
8993     <wsdl:output message="ws:MessageAckSoapOut"/>
8994   </wsdl:operation>
8995   <wsdl:operation name="MessageBoxStatus">
8996     <wsdl:input message="ws:MessageBoxStatusSoapIn"/>
8997     <wsdl:output message="ws:MessageBoxStatusSoapOut"/>
8998   </wsdl:operation>
8999   <wsdl:operation name="MessageListing">
9000     <wsdl:input message="ws:MessageListingSoapIn"/>
9001     <wsdl:output message="ws:MessageListingSoapOut"/>
9002   </wsdl:operation>
9003   <wsdl:operation name="MessageRead">
9004     <wsdl:input message="ws:MessageReadSoapIn"/>
9005     <wsdl:output message="ws:MessageReadSoapOut"/>
9006   </wsdl:operation>
9007   <wsdl:operation name="MessageRecv">
9008     <wsdl:input message="ws:MessageRecvSoapIn"/>
9009     <wsdl:output message="ws:MessageRecvSoapOut"/>
9010   </wsdl:operation>
9011   <wsdl:operation name="MessageReset">
9012     <wsdl:input message="ws:MessageResetSoapIn"/>
9013     <wsdl:output message="ws:MessageResetSoapOut"/>
```

```
9014     </wsdl:operation>
9015     <wsdl:operation name="MessageSend">
9016         <wsdl:input message="ws:MessageSendSoapIn"/>
9017         <wsdl:output message="ws:MessageSendSoapOut"/>
9018     </wsdl:operation>
9019     <wsdl:operation name="MessageTrack">
9020         <wsdl:input message="ws:MessageTrackSoapIn"/>
9021         <wsdl:output message="ws:MessageTrackSoapOut"/>
9022     </wsdl:operation>
9023     <wsdl:operation name="OrgAdmin">
9024         <wsdl:input message="ws:OrgAdminSoapIn"/>
9025         <wsdl:output message="ws:OrgAdminSoapOut"/>
9026     </wsdl:operation>
9027     <wsdl:operation name="ProviderAdmin">
9028         <wsdl:input message="ws:ProviderAdminSoapIn"/>
9029         <wsdl:output message="ws:ProviderAdminSoapOut"/>
9030     </wsdl:operation>
9031     <wsdl:operation name="UserAdmin">
9032         <wsdl:input message="ws:UserAdminSoapIn"/>
9033         <wsdl:output message="ws:UserAdminSoapOut"/>
9034     </wsdl:operation>
9035     <wsdl:operation name="OrgQuery">
9036         <wsdl:input message="ws:OrgQuerySoapIn"/>
9037         <wsdl:output message="ws:OrgQuerySoapOut"/>
9038     </wsdl:operation>
9039     <wsdl:operation name="ProviderQuery">
9040         <wsdl:input message="ws:ProviderQuerySoapIn"/>
9041         <wsdl:output message="ws:ProviderQuerySoapOut"/>
9042     </wsdl:operation>
9043     <wsdl:operation name="UserQuery">
9044         <wsdl:input message="ws:UserQuerySoapIn"/>
9045         <wsdl:output message="ws:UserQuerySoapOut"/>
9046     </wsdl:operation>
9047 </wsdl:portType>
9048 <wsdl:binding name="eMSEPSoap" type="ws:eMSEPSoap">
9049     <soap:binding style="document"
9050 transport="http://schemas.xmlsoap.org/soap/http"/>
9051     <wsdl:operation name="MessageAck">
9052         <soap:operation soapAction="http://www.e-
9053 ms.ca/Schema/v1/WS/eMSEP/MessageAck"
9054             style="document"/>
9055         <wsdl:input>
9056             <soap:body use="literal"/>
9057         </wsdl:input>
9058         <wsdl:output>
9059             <soap:body use="literal"/>
9060         </wsdl:output>
9061     </wsdl:operation>
9062     <wsdl:operation name="MessageBoxStatus">
9063         <soap:operation soapAction="http://www.e-
9064 ms.ca/Schema/v1/WS/eMSEP/MessageBoxStatus"
9065             style="document"/>
9066         <wsdl:input>
9067             <soap:body use="literal"/>
9068         </wsdl:input>
9069         <wsdl:output>
9070             <soap:body use="literal"/>
9071         </wsdl:output>
9072     </wsdl:operation>
9073     <wsdl:operation name="MessageListing">
9074         <soap:operation soapAction="http://www.e-
9075 ms.ca/Schema/v1/WS/eMSEP/MessageListing"
9076             style="document"/>
9077         <wsdl:input>
9078             <soap:body use="literal"/>
9079         </wsdl:input>
9080         <wsdl:output>
9081             <soap:body use="literal"/>
9082         </wsdl:output>
9083     </wsdl:operation>
```

```
9084         <wsdl:operation name="MessageRead">
9085             <soap:operation soapAction="http://www.e-
9086 ms.ca/Schema/v1/WS/eMSEP/MessageRead"
9087                 style="document"/>
9088             <wsdl:input>
9089                 <soap:body use="literal"/>
9090             </wsdl:input>
9091             <wsdl:output>
9092                 <soap:body use="literal"/>
9093             </wsdl:output>
9094         </wsdl:operation>
9095         <wsdl:operation name="MessageRecv">
9096             <soap:operation soapAction="http://www.e-
9097 ms.ca/Schema/v1/WS/eMSEP/MessageRecv"
9098                 style="document"/>
9099             <wsdl:input>
9100                 <soap:body use="literal"/>
9101             </wsdl:input>
9102             <wsdl:output>
9103                 <soap:body use="literal"/>
9104             </wsdl:output>
9105         </wsdl:operation>
9106         <wsdl:operation name="MessageReset">
9107             <soap:operation soapAction="http://www.e-
9108 ms.ca/Schema/v1/WS/eMSEP/MessageReset"
9109                 style="document"/>
9110             <wsdl:input>
9111                 <soap:body use="literal"/>
9112             </wsdl:input>
9113             <wsdl:output>
9114                 <soap:body use="literal"/>
9115             </wsdl:output>
9116         </wsdl:operation>
9117         <wsdl:operation name="MessageSend">
9118             <soap:operation soapAction="http://www.e-
9119 ms.ca/Schema/v1/WS/eMSEP/MessageSend"
9120                 style="document"/>
9121             <wsdl:input>
9122                 <soap:body use="literal"/>
9123             </wsdl:input>
9124             <wsdl:output>
9125                 <soap:body use="literal"/>
9126             </wsdl:output>
9127         </wsdl:operation>
9128         <wsdl:operation name="MessageTrack">
9129             <soap:operation soapAction="http://www.e-
9130 ms.ca/Schema/v1/WS/eMSEP/MessageTrack"
9131                 style="document"/>
9132             <wsdl:input>
9133                 <soap:body use="literal"/>
9134             </wsdl:input>
9135             <wsdl:output>
9136                 <soap:body use="literal"/>
9137             </wsdl:output>
9138         </wsdl:operation>
9139         <wsdl:operation name="OrgAdmin">
9140             <soap:operation soapAction="http://www.e-
9141 ms.ca/Schema/v1/WS/eMSEP/OrgAdmin"
9142                 style="document"/>
9143             <wsdl:input>
9144                 <soap:body use="literal"/>
9145             </wsdl:input>
9146             <wsdl:output>
9147                 <soap:body use="literal"/>
9148             </wsdl:output>
9149         </wsdl:operation>
9150         <wsdl:operation name="ProviderAdmin">
9151             <soap:operation soapAction="http://www.e-
9152 ms.ca/Schema/v1/WS/eMSEP/ProviderAdmin"
9153                 style="document"/>
```

```
9154         <wsdl:input>
9155             <soap:body use="literal"/>
9156         </wsdl:input>
9157         <wsdl:output>
9158             <soap:body use="literal"/>
9159         </wsdl:output>
9160     </wsdl:operation>
9161     <wsdl:operation name="UserAdmin">
9162         <soap:operation soapAction="http://www.e-
9163 ms.ca/Schema/v1/WS/eMSEP/UserAdmin"
9164             style="document"/>
9165         <wsdl:input>
9166             <soap:body use="literal"/>
9167         </wsdl:input>
9168         <wsdl:output>
9169             <soap:body use="literal"/>
9170         </wsdl:output>
9171     </wsdl:operation>
9172     <wsdl:operation name="OrgQuery">
9173         <soap:operation soapAction="http://www.e-
9174 ms.ca/Schema/v1/WS/eMSEP/OrgQuery"
9175             style="document"/>
9176         <wsdl:input>
9177             <soap:body use="literal"/>
9178         </wsdl:input>
9179         <wsdl:output>
9180             <soap:body use="literal"/>
9181         </wsdl:output>
9182     </wsdl:operation>
9183     <wsdl:operation name="ProviderQuery">
9184         <soap:operation soapAction="http://www.e-
9185 ms.ca/Schema/v1/WS/eMSEP/ProviderQuery"
9186             style="document"/>
9187         <wsdl:input>
9188             <soap:body use="literal"/>
9189         </wsdl:input>
9190         <wsdl:output>
9191             <soap:body use="literal"/>
9192         </wsdl:output>
9193     </wsdl:operation>
9194     <wsdl:operation name="UserQuery">
9195         <soap:operation soapAction="http://www.e-
9196 ms.ca/Schema/v1/WS/eMSEP/UserQuery"
9197             style="document"/>
9198         <wsdl:input>
9199             <soap:body use="literal"/>
9200         </wsdl:input>
9201         <wsdl:output>
9202             <soap:body use="literal"/>
9203         </wsdl:output>
9204     </wsdl:operation>
9205 </wsdl:binding>
9206 <wsdl:service name="eMSEP">
9207     <documentation/>
9208     <wsdl:port binding="ws:eMSEPSoap" name="eMSEPSoap">
9209         <soap:address location="http://localhost/eMSEP/eMSEP.asmx"/>
9210     </wsdl:port>
9211 </wsdl:service>
9212 </wsdl:definitions>
```

9213

10References

10.1 Normative

- [RFC2141] R. Moats, *URN Syntax*, <http://www.ietf.org/rfc/rfc2141.txt>, IETF RFC 2141, May 1997.
- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [RFC1939] J. Myers, M. Rose, *Post Office Protocol – Version 3*, <http://www.ietf.org/rfc/rfc1939.txt>, IETF RFC 1939, May 1996.
- [RFC2616] R. Fielding, et al., *Hypertext Transfer Protocol -- HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, IETF RFC 2616, June 1999.
- [RFC2045] N. Freed, N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, <http://www.ietf.org/rfc/rfc2045.txt>, IETF RFC 2045, November 1996.
(Related: RFC2046, RFC2047, RFC2048, RFC2049)
- [HL7-CDA] R. Dolin, et al., *HL7 Clinical Document Architecture – Release 2.0, Committee Ballot #03*, <http://www.hl7.org/Library/Committees/structure/CDA.ReleaseTwo.CommitteeBallot03.Aug.2004.zip>, August 1, 2004.
- [HL7-V3-RIM] Health Level 7 Reference Information Model, <http://www.hl7.org/library/data-model/RIM/C30124/rim.htm>
- [W3C-WS] W3C Web Services Activity – SOAP Recommendations, <http://www.w3.org/2002/ws/>
- [WS-I] Web Services Interoperability Organization – Basic Profile Working Group, <http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile>

10.2 Non-Normative

- [EMS-BRD] *Electronic Medical Summary Project: Business Requirements Document – Phase 2*, VIHA, May 20, 2004.
- [EMS-SRS] *Electronic Medical Summary Project: System Requirements Specification – Phase 2*, VIHA, June 8, 2004.
- [EMS-PIA] P. Smith, F. Kaiser, *Vancouver Island Health Authority – Privacy Impact Assessment Tool for Electronic Medical Summary Project – Phase 2*, VIHA, November 23, 2004.

9257

9258 **[EMS-CDA]** *Electronic Medical Summary Project: e-MS Clinical Document Architecture*
9259 Implementation Guide, VIHA, December, 2004.

9260

Appendix A. Acknowledgments

9261

9262 The following individuals were members of the e-MS project technical committee during the
9263 development of this specification:

- 9264 • Karen Kuhn (e-MS Project Manager), Vancouver Island Health Authority
- 9265 • Brad Head (e-MS Technical Architect), Sierra Systems Group, Inc.
- 9266 • Pat Willmott, Sierra Systems Group, Inc.
- 9267 • Mike Leavy, BC Ministry of Health
- 9268 • Robert Grant, BC Ministry of Health
- 9269 • George Earl, Microsoft
- 9270 • Sam Fisher, Vancouver Island Health Authority
- 9271 • Lloyd Loisel, Vancouver Island Health Authority
- 9272 • David Booth, Vancouver Island Health Authority
- 9273 • Dennis Neibergal, CLINICARE
- 9274 • Brendan Byrne, Wolf Medical Systems

9275 In addition, the following people made contributions to this specification:

- 9276 • Herman Gocher, Sierra Systems Group, Inc.
- 9277 • Pamela Smith, Sierra Systems Group, Inc.
- 9278 • James Angus, Vancouver Island Health Authority
- 9279 • Jame Healy, Sunaptic Solutions
- 9280 • James Roland, Sunaptic Solutions
- 9281 • Kevin Routley, Vancouver Island Health Authority
- 9282 • Frieda Kaiser, Vancouver Island Health Authority

9283

9284

Appendix B. e-MS Clinical Document Architecture Implementation Guide

9285

9286

This appendix references document **[EMS-CDA]**

9287

Appendix C. Revision History

Rev	Date	By Whom	What
WD-01	2004-06-18	Brad Head	Initial draft version
WD-02	2004-06-24	Brad Head	Filled Query and Administrative. Incorporated feedback from initial draft.
WD-03	2004-06-30	Brad Head	Updated after comments from review of WD-02 with e-MS core team. Added WSDL section.
WD-04	2004-07-02	Brad Head	Added SOAP examples, and Command Pattern description. Renamed protocol to e-MSEP.
WD-05	2004-08-13	Brad Head	Updated protocol schema (syntax) to conform to naming conventions. Updated ProviderQuery to support retrieving updates from a certain timestamp. Updated examples to match new schemas. Incorporated feedback from technical committee regarding Working Draft 04. Removed the graph diagram representations of the schema models. This will be added back once the status of the document is beyond Draft status.
WD-06	2004-08-18	Brad Head	Updated Figures 2 and 3 to reflect new message names. Updated MessagingDataTypes.xsd to workaround a schema issue with BizTalk web services wizard – does not impact the syntax. 100% compatible with WD-05 specifications. Corrected some grammar errors and spelling. Filled in the SOAP Binding section with the WSDL specifications and their descriptions. Added Normative reference to MIME RFC. Updated e-MS CDA Implementation Guide reference.
WD-07	2004-10-15	Brad Head	Modified the eMSEP.xsd schema, areas affected: MessageSend, MessageRecvResponse. Of particular note is new Message/Content/ClinicalDocument structure in the MessageSend and MessageRecvResponse. Fixed some bugs in cardinality to Query related to Telecom type. Restructured complexTypes to improve modeling. Added new section on Notifications to detail them more. Tested the samples for validation.
WD-08	2004-10-25	Brad Head	Added MessageType element to MessageInfo complexType. This provides the message type {ClinicalDocument Notification Other} when

Rev	Date	By Whom	What
			<p>responding to a MessageListing request with showHeaderInfo="true". It also affects MessageRecvResponse, MessageTrackResponse, which could optionally include this new element in the header of the response.</p> <p>Added new Example 42 showing ProviderQuery requesting all Providers modified since a date/time value.</p> <p>Fixed some typos and spelling errors.</p> <p>Modified WXS complexType MessageResponseBase in DataTypes.xsd by creating a simpleType for the status attribute. This was necessary to workaround an incorrect interpretation of the WXS Facet rules by BizTalk related to restriction facets and required attributes. This fix does not affect the definition of the schema. It only affects DataTypes.xsd and the implementation of the status simpleType.</p> <p>Removed comments about Organization identifiers as UUID. Organizations are identified by their messaging address.</p> <p>Further constrained the PostalAddress complex data type – removing HL7 fields that are not supported in our Provider data model.</p>
WD-09	2004-11-16	Brad Head	<p>[1] Merged HL7Flavors.xsd into DataTypes.xsd due to reuse of the HL7 derived types in common types.</p> <p>[2] Modified MessagingAddress type to be an empty element with two attributes: displayName and value.</p> <p>[3] Restricted the MessagingAddress values to be either a Provider or Organization address: urn:provider:... or urn:org:...</p> <p>[4] To, From elements are of type MessagingAddress.</p> <p>[5] Corrected the cardinality of Membership to be [0...unbounded] in the ProviderQueryResponse. Previously the cardinality was [1.. unbounded] which prevented a Provider from being listed that wasn't already a member of an organization.</p> <p>[6] Removed superfluous Organization entries in OrgAdminResponse for XPath /OrgAdminResponse/Access/Listing/User, and /OrgAdminResponse/Users/Listing/User.</p>

Rev	Date	By Whom	What
			<p>[7] Removed superfluous Access entry for ProviderAdminResponse for XPath: /ProviderAdminResponse/Access/Listing</p> <p>[8] Changed Organization type in UserAdminResponse at XPath /UserAdminResponse/User/Affiliation/Organization to be a full Organization type with PostalAddress, Telecom, etc.</p> <p>[9] Modified samples contained in this document and bundled with the source files to reflect changes made.</p> <p>[10] Modified EmptyElement type to use phantom prohibited attribute, rather than phantom child element which was causing issues with BizTalk.</p> <p>[11] Refined wording around notifications to align with business requirements.</p>
WD-10	2004-12-07	Brad Head	<p>[1] Fixed a cardinality issue with Organization Details. The messaging address was required during an update. This was meant to be optional along with all the other settings. This has been fixed.</p> <p>[2] The WSDL now matches exactly the eMSEP schemas after abandoning the WSDL generator that the service used. The WSDL now contains an exact copy of the eMSEP schemas, complete with data types and cardinality preserved. We've also automated its conformance validation against the Web Service Interoperability Organization (WS-I) Basic Profile.</p>
WD-11	2005-03-28	Brad Head	<p>[1] In MessageSend, section 4.2.7 'Body' element should have been 'Content' element when mentioning the structure of the Message child elements.</p> <p>[2] Updated MessageSend to have mandatory MessageID to support reliable messaging. Added Figure 4 to illustrate reliable delivery.</p> <p>[3] Updated eMSEP.wsdl to reflect its use of newest datatypes.xsd from HL7</p>

9288

Appendix D. Notices

9289

9290 Copyright © 2004, 2005 by Vancouver Island Health Authority (VIHA). *All Rights Reserved.*

9291 This document and translations of it may be copied and furnished to others, and derivative works
9292 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
9293 published and distributed, in whole or in part, without restriction of any kind, provided that the
9294 above copyright notice and this paragraph are included on all such copies and derivative works.
9295 However, this document itself does not be modified in any way, such as by removing the
9296 copyright notice or references to VIHA, except as needed for the purpose of developing VIHA
9297 specifications, in which case the procedures for copyrights defined in the VIHA Intellectual
9298 Property Rights document must be followed, or as required to translate it into languages other
9299 than English.

9300 The limited permissions granted above are perpetual and will not be revoked by VIHA or its
9301 successors or assigns.

9302 This document and the information contained herein is provided on an "AS IS" basis and VIHA
9303 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
9304 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
9305 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
9306 PARTICULAR PURPOSE.