



Canada
Health
Infoway

Inforoute
Santé
du Canada

HL7 v3 pan-Canadian Messaging Standards

Data Type Specification

~~September 23~~ December 12, 2013

MR02.05

Document Information

Author:	SCWG 6
Creation Date:	July 29, 2008
Last Updated:	~~September 23~~ December 12, 2013
Language:	English
Document Number:	SC-3002-EN
Document Status	In Progress
Infoway Project	Various
Distribution	Standards Collaborative Working Group (SCWG); Implementers; Other Stakeholders / Reviewers;
Contact Information	Toronto Office: 150 King Street West, Suite 1300 Toronto, Ontario M5H 1J9 Tel.: (416) 979-4606 Toll free: 1-888-733-6462 Fax: (416) 593-5911 http://www.infoway-inforoute.ca

Version Tracking

Version	Author(s)	Change Description	Date
v1.0	SC Standards Team	Initial version, drawn primarily from cross-project specification.	2007-Nov-07
v1.1	SC Standards Team	Minor updates resulting from vendor of V1.0.	2008-Apr-18

Version	Author(s)	Change Description	Date
v1.2	SC Standards Team	Content: - Updated examples for many data types - Updated definition of ED abstract data type and ST.LANG and fixed examples - Added pre-adopted components table to better communicate which components cannot be found in the released schemas (currently) - Added best practices section for data types derivations - Removed ED.IMAGE until an adopter can implement and assist with vetting the stated definition and/or constraints. Formatting: - Added document sections to conform to implementation guide specification documents for consistency. Traceability: - Changes were made based on the following items from the SCWG6 Tracking Log (included in the release package): #26, 27, 46, 47, 50, 53, 54, 55, 57, 74, 81, 107, 108, 109	2008-Jun-17
v1.3	SC Standards Team	Formatting - Accepted all changes tracked in v1.2 Traceability Changes were made based on the following items from the SCWG6 Tracking Log (included in the release package) F-ID: 10, 88, 98, 145, 146, 150, 153, 194, 200, 228, 233, 237, 251, 257, 261	2008-Dec-17
V1.4	SC Standards Team	Formatting - Accepted all changed tracked in v1.3 Traceability Changes were made based on the following items from the SCWG6 Tracking Log (including in the release package) MR2009 ER: 36, 38, 40, 143, 145, 146, 149, 209, 226, 227, 268, Example fragments corrected	2009-Mar-07

Version	Author(s)	Change Description	Date
V1.5	SC Standards Team	Formatting - Accepted all tracked changes in v1.4 Traceability - Added examples to the DT spec for CD.LAB and make translation.originalText not permitted - Changed PQ.DRUG significant digits to 5, so total digits 6 - Pre-adopted PQ.originalText from data types R2 for use in PQ.LAB. Needed for BC implementation. - Added ASGN to x_BasicPersonNameUse for PN.BASIC - Added example for Rural Route to AD.FULL - Updated data type specification to discuss difference between the data type components or properties and their subsequent representation in the ITS (use example of highClosed, lowClosed vs. inclusive indicator). - Changed sentence to be identical between IVL<TS.DATETIME> and IVL<TS.FULLDATE>. IVL<DATETIME> is correct (so updated IVL<TS.FULLDATE>). - Added pertinent verbiage to data types specification regarding rules for inclusion of flavor in message instances. Don't have to declare a specializationType (in a message instance) when the flavor is inclusive in the spec itself. If receive a flavor, you're NOT necessarily expected to validate against the spec. An error can be returned if the received data doesn't validate against the specification. Data type specializationType declarations are not necessary 'over the wire' as they're just a set of extra (local, CA) constraints. - Added examples from the CSI project.	2010-Mar-26
V1.6	SC Standards Team	- Accepted all tracked changes in v1.5 - Changed footer to reflect Copyright 2010 - Formatting changes to fragments for easier readability - Edited section 2.2 to clarify wording - Added additional wording for time zone - Clarified specification for - TS.FULLDATEWITHTIME - IVL< TS.FULLDATEWITHTIME - Additional guidance on conformance for low and high properties of TS flavours - Corrected example fragment for: - IVL<TS.FULLDATE> - IVL.WIDTH<TS.FULLDATE> - GTS.BOUNDEDPIVL - II.OID - Additional guidance for ED.SIGNATURE - Added originalText to PQ.LAB and example fragment as per v1.5 - Added open issue for PIVL<TS.FULLDATETIME>	2010-Aug-31

Version	Author(s)	Change Description	Date
V1.7	SC Standards Team	• Change mask for TS.DATETIME and TS.FULLDATETIME for fraction of seconds • Add an additional example for CD • Correct typos in description of usage for II.OID. II.OID use is mandatory and is fixed to 'BUS'. • Correct typo in the example of CD.LAB. • Increase the name of PN from 50 to 75 chars. • Add a new Data Type flavor: TS.FULLDATEPARTTIME. • Change the name of the lab measurement value set from x_LabUnitsOfMeasure to LaboratoryObservationUnitsOfMeasure in PQ.LAB • Pre-adopted originalText in the URG<x>. • Add more clarification about Charset in the ED data type. • Change the length of the value property in TEL.EMAIL data type from 50 to 128. • Make the wording about PQ.value more clear • Relax the constraint on the 4 address parts ("city", "state", "postalCode" and "country" properties) from Mandatory to Required in AD.FULL • Create a new abstract DT flavor: TEL.ALL • Increase II extension limit from 20 char to 50 char • Updated PQ.BASIC to allow for 6 digits beneath the decimal. • Update copyright and disclaimer	2013-September-23

Formatted: English (Canada)

Acknowledgments

N/A

Formatted: Heading: Sub-head

Copyright Notice

Except for public domain material or material otherwise licensed to Canada Health Infoway Inc., the material in this document, including all content, designs, text, graphics, and the arrangement thereof, are protected by copyright and other intellectual property laws, and are owned or controlled by Canada Health Infoway Inc. or the party credited as the author or provider of the materials. No alterations, deletions or substitutions may be made in it without the prior written consent of the owner. No part of it may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, email or any information storage and retrieval system, without the prior written consent of the owner. No trademark license is granted in connection with the material contained in the document. HL7® is registered trademark of Health Level Seven, Inc. (<http://www.hl7.org>)

This document contains information for which copyright is held by Health Level Seven, Inc. Implementers of the standards (those developing software or otherwise making use of the specification) are required to be members of either Health Level Seven Inc., the HL7 affiliate in Canada or one of the other HL7 affiliates. There is no such membership requirement for individuals and organizations who merely install or use software with built-in HL7 interfaces.

LOINC® is a registered trademark of the Regenstrief Institute, Inc. (<http://loinc.org>)

The LOINC table and LOINC codes are copyright © 1995-2013, Regenstrief Institute, Inc. and the Logical Observation Identifiers Names and Codes (LOINC) Committee.

Read the full LOINC® and RELMA® Terms of Use (<http://loinc.org/terms-of-use>)

This product includes all or a portion of the UCUM table, UCUM codes, and UCUM definitions or is derived from it, subject to a license from Regenstrief Institute, Inc. and The UCUM Organization. Your use of the UCUM table, UCUM codes, UCUM definitions also is subject to this license, a copy of which is available at <http://unitsofmeasure.org>. The current complete UCUM table, UCUM Specification are available for download at <http://unitsofmeasure.org>. The UCUM table and UCUM codes are copyright © 1995-2009, Regenstrief Institute, Inc. and the Unified Codes for Units of Measures (UCUM) Organization. All rights reserved.

THE UCUM TABLE (IN ALL FORMATS), UCUM DEFINITIONS, AND SPECIFICATION ARE PROVIDED "AS IS." ANY EXPRESS OR IMPLIED WARRANTIES ARE DISCLAIMED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Read the full UCUM Terms of Use (<http://unitsofmeasure.org/trac/wiki/TermsOfUse>)

This material includes SNOMED Clinical Terms® (SNOMED CT®) which is used by permission of the

International Health Terminology Standards Development Organisation (IHTSDO). All rights reserved.

SNOMED CT®, was originally created by The College of American Pathologists.

“SNOMED” and “SNOMED CT” are registered trademarks of the IHTSDO (www.ihtsdo.org)

Disclaimer

Canada Health Infoway Inc. (Infoway) in its role as strategic investor may require adherence to this implementation guide or similar specification documents as a criteria for receiving funding or payment.

The Standards Collaborative is the pan-Canadian organization responsible for the coordination of health information standards. The Standards Collaborative provides: a single point of contact for coordination of pan-Canadian standards throughout the standards life cycle: development, implementation support, education, maintenance and conformance; a streamlined governance, processes and operations; efficiencies gained by combining administrative services such as communications, website management, event planning/management, education and administrative support; and coordination of development, maintenance and balloting processes, such that they are harmonized in a way that each adds value without duplication.

This document forms part of a standard which was developed through the Standards Collaborative governance processes that are a consensus-based standards development process that brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While Infoway administers the process, establishes rules to promote fairness in the development of consensus, and may engage consultants to facilitate the process and develop the documentation, it does not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments, interpretations, decisions and guidance contained in the standards and guideline publications.

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

Infoway disclaims liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. Infoway disclaims and makes no guaranty or warranty, expressed or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. Infoway does not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, Infoway is not undertaking to render professional or other services for or on behalf of any person or entity, nor is Infoway undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user should consult for additional views or information not covered by this publication.

~~This document is fully copyright protected by the owner. The owner has the exclusive right to make copies of this document. No alterations, deletions or substitutions may be made in it without the prior written consent of the owner. No part of it may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, email or any information storage and retrieval system, without the prior written consent of the owner.~~

~~HL7® is registered trademark of Health Level Seven, Inc. (<http://www.hl7.org>)~~

This document contains information for which copyright is held by Health Level Seven, Inc. Implementers of the standards (those developing software or otherwise making use of the specification) are required to be members of either Health Level Seven Inc., HL7 Canada or one of the other HL7 affiliates. There is no such membership requirement for individuals and organizations which merely install or use software with built-in HL7 interfaces.

LOINC® is a registered trademark of the Regenstrief Institute, Inc. (<http://regenstrief.org>)

This material includes SNOMED Clinical Terms® (SNOMED-CT®) which is used by permission of the International Health Terminology Standards Development Organisation (IHTSDO). All rights reserved. SNOMED-CT®, was originally created by The College of American Pathologists.

"SNOMED" and "SNOMED-CT" are registered trademarks of the IHTSDO. (www.ihtsdo.org)

Disclaimer

Canada Health Infoway Inc. (*Infoway*) in its role as strategic investor may require adherence to this or similar specification documents as a criteria for funding or payment.

This document forms part of a standard which was developed through the EHR Standards Collaboration Process (SCP). The SCP is a consensus-based standards development process that brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While *Infoway* administers the process, establishes rules to promote fairness in the development of consensus, and may engage consultants to facilitate the process and develop the documentation, it does not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in its standards and guideline publications.

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

Infoway disclaims liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. *Infoway* disclaims and makes no guaranty or warranty, expressed or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. *Infoway* does not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, *Infoway* is not undertaking to render professional or other services for or on behalf of any person or entity, nor is *Infoway* undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

PREFACE – QUICK DOCUMENT OVERVIEW

Purpose	The purpose of this document is to provide implementation, compliance and conformance guidelines for use in developing software that conforms to the Message Specifications developed by Infoway through the Standards Collaborative ~~on-Process (SCP)~~ governance process. Specifically this Specification document contains material which defines the data types and data type components.
Audience	The audience for this document includes organizations implementing the Message Specifications contained in the appropriate Implementation Guide Volumes of this maintenance release (e.g. Shared Health Record Medication Management, Laboratory, Provider and Client Registries). It does not include terminology guidance for Laboratory, iEHR and Public Health Surveillance, Pharmacy and Non-Pharmacy Claims. Stakeholders must consult the program specific Implementation Guides. The audience includes technical and business audiences who will use the guide to develop implementations and validate that these implementations conform to the Message Specifications and to the requirements of stakeholder organizations.

Structure	This document includes the following specific sections:	
	Background	Information regarding prior efforts.
	Introduction	General introduction to this document.
	Best Practices	Guidance on usage patterns for data types.
	Timing	General guidance and specifications on date and time data types.
	Identifiers and Codes	General guidance and specifications on identifier and coded data types.
	Text and Documents	General guidance and specifications on text and document data types.
	Numeric	General guidance and specifications on numeric data types.
	Demographic	General guidance and specifications on demographic data types.
	Miscellaneous	General guidance and specifications on other data types.

TABLE OF CONTENTS

1	Introduction.....	1
1.1	Background.....	1
1.2	Pre-Adoption.....	1
1.3	Flavor Declaration.....	2
1.4	XML ITS Representation of Datatype Properties.....	2
1.5	Null Properties.....	2
1.6	Canadian Rules.....	2
1.7	Property Characteristics.....	3
2	Best Practices.....	3
2.1	Health Level 7 Standard.....	3
2.2	Intervals and Single Values.....	3
3	Dates + Timing.....	4
3.1	TS.DATE – Timestamp (Date Only, Partial Date Allowed).....	4
3.2	TS.DATETIME – Timestamp (Date + Time, Partial Date + Time Allowed).....	4
3.3	TS.FULLDATEWITHTIME – Timestamp <Abstract>.....	5
3.4	TS.FULLDATE – Timestamp (Date Only, Partial Date Not Allowed).....	5
3.5	TS.FULLDATETIME – Timestamp (Date + Time, Partial Date + Time Not Allowed).....	5
3.6	TS.FULLDATEPARTTIME – Timestamp (Date + Time, Partial Date Not Allowed, Partial Time allowed).....	6
3.7	IVL<x> - Interval <Abstract>.....	7
3.8	IVL<TS.DATE> – Partial Date Range.....	7
3.9	URG<x> - Interval <Abstract>.....	8
3.10	URG<TS.DATE> – Uncertain Date.....	8
3.11	IVL<TS.DATETIME> – Partial Date + Time Range.....	9
3.12	IVL<TS.FULLDATEWITHTIME> – Precise Date Range – <Abstract>.....	9
3.13	IVL<TS.FULLDATE> – Precise Date Range.....	10
3.14	IVL<TS.FULLDATETIME> – Precise Date-Time Range.....	11
3.15	IVL.LOW<TS.DATE> – Partial Start Date.....	11
3.16	IVL.LOW<TS.FULLDATE> – Precise Start Date.....	12
3.17	IVL.HIGH<TS.FULLDATE> – Precise End Date.....	12
3.18	IVL.WIDTH<TS.FULLDATE> – Duration.....	13
3.19	PIVL<TS.DATETIME> – Periodic Interval of Time.....	13
3.20	PIVL<TS.FULLDATETIME> – Precise Periodic Interval of Time.....	14
3.21	GTS.BOUNDEDPIVL – General Timing Specification (Time-range and Frequency).....	15
4	Identifiers + Codes.....	16
4.1	II – Instance Identifier <Abstract>.....	16
4.2	II.BUSVER – Instance Identifier (Business with Version) - deprecated.....	17
4.3	II.OID – Instance Identifier (OID).....	17
4.4	II.PUBLIC – Instance Identifier (Public).....	18
4.5	II.PUBLICVER – Instance Identifier (Public with Version) - deprecated.....	18
4.6	II.TOKEN – Instance Identifier (Token).....	18
4.7	II.BUS AND VER – Instance Identifier (Business and Version) <Abstract>.....	19
4.8	II.VER – Instance Identifier (Version).....	19
4.9	II.BUS – Instance Identifier (Business).....	20
4.10	CS – Coded Simple.....	20
4.11	CV – Coded Value.....	21
4.12	CE – Coded with Equivalents.....	21
4.13	CD – Concept Descriptor.....	22
4.14	CD.LAB – Lab Concept Descriptor.....	24
5	Text & Documents.....	25
5.1	ST – String.....	25

5.2	ST.LANG – String with Language	25
5.3	ED – Encapsulated Data <Abstract>	26
5.4	ED.DOC – Encapsulated Data (Document)	27
5.5	ED.DOCORREF – Encapsulated Data (Document or Reference) <Abstract>	27
5.6	ED.DOCREF – Encapsulated Data (Document Reference)	28
5.7	ED.SIGNATURE – Encapsulated Data (Digital Signature)	29
6	Numeric	30
6.1	INT.POS – Positive Integer	30
6.2	INT.NONNEG – Non-negative Integer	30
6.3	REAL.COORD – Real Number (Coordinate)	30
6.4	REAL.CONF – Real Number (Confidence)	30
6.5	MO.CAD – Canadian Money	31
6.6	PQ.BASIC – Basic Physical Quantity	32
6.7	PQ.DRUG – Drug Physical Quantity	32
6.8	PQ.TIME – Time Quantity	32
6.9	PQ.HEIGHTWEIGHT – Height or Weight Quantity	33
6.10	PQ.DISTANCE – Distance Quantity	33
6.11	PQ.LAB – Laboratory Quantity	33
6.12	IVL<PQ.x> – Quantity Range	34
6.13	URG<PQ.x> – Uncertain Quantity	35
6.14	RTO<PQ.DRUG, PQ.TIME> – Drug Quantity per Time Duration	36
6.15	RTO<MO.CAD,PQ.BASIC> – Money per Unit Quantity	37
7	Demographics	38
7.1	PN.BASIC – Basic Person Name	38
7.2	PN.SIMPLE – Simple Person Name	38
7.3	PN.FULL – Full Person Name	39
7.4	PN.SEARCH – Searchable Name	39
7.5	TN – Trivial Name	40
7.6	AD.BASIC – Basic Address	41
7.7	AD.FULL – Full Address	42
7.8	AD.SEARCH – Searchable Address	44
7.9	TEL.ALL – Phone, E-mail or URI <Abstract>	44
7.10	TEL.PHONEMAIL – Phone or E-mail <Abstract>	45
7.11	TEL.PHONE – Phone Number	45
7.11	TEL.EMAIL – Email Address	47
7.12	TEL.URI – Uniform Resource Identifier	47
8	Miscellaneous	48
8.1	ANY – Unrestricted <Abstract>	48
8.2	ANY.LAB – <Abstract>	48
8.3	ANY.CA.IZ – <Abstract>	48
8.4	ANY.PATH – <Abstract>	48
8.5	BL - Boolean	49
8.6	LIST<x> - List	49
8.7	SET<x> - Set	49
1	Introduction	1
1.1	Background	1
1.2	Pre Adoption	1
1.3	Flavor Declaration	2
1.4	XML ITS Representation of Datatype Properties	2
1.5	Null Properties	2
1.6	Canadian Rules	2
1.7	Property Characteristics	3
2	Best Practices	3
2.1	Health Level 7 Standard	3
2.2	Intervals and Single Values	3
3	Dates + Timing	4

3.1	~~TS.DATE~~ Timestamp (Date Only, Partial Date Allowed)	4
3.2	~~TS.DATETIME~~ Timestamp (Date + Time, Partial Date + Time Allowed)	4
3.3	~~TS.FULLDATEWITHTIME~~ Timestamp <Abstract>	5
3.4	~~TS.FULLDATE~~ Timestamp (Date Only, Partial Date Not Allowed)	5
3.5	~~TS.FULLDATETIME~~ Timestamp (Date + Time, Partial Date + Time Not Allowed)	5
3.6	~~IVL<x>~~ Interval <Abstract>	6
3.7	~~IVL<TS.DATE>~~ Partial Date Range	6
3.8	~~URG<x>~~ Interval <Abstract>	7
3.9	~~URG<TS.DATE>~~ Uncertain Date	7
3.10	~~IVL<TS.DATETIME>~~ Partial Date + Time Range	8
3.11	~~IVL<TS.FULLDATEWITHTIME>~~ Precise Date Range <Abstract>	8
3.12	~~IVL<TS.FULLDATE>~~ Precise Date Range	9
3.13	~~IVL<TS.FULLDATETIME>~~ Precise Date Time Range	10
3.14	~~IVL.LOW<TS.DATE>~~ Partial Start Date	10
3.15	~~IVL.LOW<TS.FULLDATE>~~ Precise Start Date	11
3.16	~~IVL.HIGH<TS.FULLDATE>~~ Precise End Date	11
3.17	~~IVL.WIDTH<TS.FULLDATE>~~ Duration	12
3.18	~~PIVL<TS.DATETIME>~~ Periodic Interval of Time	12
3.19	~~PIVL<TS.FULLDATETIME>~~ Precise Periodic Interval of Time	13
3.20	~~GTS.BOUNDEDPIVL~~ General Timing Specification (Time range and Frequency)	14
4	~~Identifiers + Codes~~	15
4.1	~~II~~ Instance Identifier <Abstract>	15
4.2	~~II.BUSVER~~ Instance Identifier (Business with Version) deprecated	16
4.3	~~II.OID~~ Instance Identifier (OID)	16
4.4	~~II.PUBLIC~~ Instance Identifier (Public)	17
4.5	~~II.PUBLICVER~~ Instance Identifier (Public with Version) deprecated	17
4.6	~~II.TOKEN~~ Instance Identifier (Token)	17
4.7	~~II.BUS AND VER~~ Instance Identifier (Business and Version) <Abstract>	18
4.8	~~II.VER~~ Instance Identifier (Version)	18
4.9	~~II.BUS~~ Instance Identifier (Business)	19
4.10	~~CS~~ Coded Simple	19
4.11	~~CV~~ Coded Value	20
4.12	~~CE~~ Coded with Equivalents	20
4.13	~~CD~~ Concept Descriptor	21
4.14	~~CD.LAB~~ Lab Concept Descriptor	23
5	~~Text & Documents~~	24
5.1	~~ST~~ String	24
5.2	~~ST.LANG~~ String with Language	24
5.3	~~ED~~ Encapsulated Data <Abstract>	25
5.4	~~ED.DOC~~ Encapsulated Data (Document)	26
5.5	~~ED.DOCORREF~~ Encapsulated Data (Document or Reference) <Abstract>	26
5.6	~~ED.DOCREF~~ Encapsulated Data (Document Reference)	27
5.7	~~ED.SIGNATURE~~ Encapsulated Data (Digital Signature)	28
6	~~Numeric~~	29
6.1	~~INT.POS~~ Positive Integer	29
6.2	~~INT.NONNEG~~ Non-negative Integer	29
6.3	~~REAL.COORD~~ Real Number (Coordinate)	29
6.4	~~REAL.CONF~~ Real Number (Confidence)	29
6.5	~~MO.CAD~~ Canadian Money	30
6.6	~~PQ.BASIC~~ Basic Physical Quantity	31
6.7	~~PQ.DRUG~~ Drug Physical Quantity	31
6.8	~~PQ.TIME~~ Time Quantity	31
6.9	~~PQ.HEIGHTWEIGHT~~ Height or Weight Quantity	32
6.10	~~PQ.DISTANCE~~ Distance Quantity	32
6.11	~~PQ.LAB~~ Laboratory Quantity	32
6.12	~~IVL<PQ.x>~~ Quantity Range	33

6.13	URG<PQ.x> Uncertain Quantity	34
6.14	RTO<PQ.DRUG, PQ.TIME> Drug Quantity per Time Duration	34
6.15	RTO<MO.CAD,PQ.BASIC> Money per Unit Quantity	36
7	Demographics	37
7.1	PN.BASIC Basic Person Name	37
7.2	PN.SIMPLE Simple Person Name	37
7.3	PN.FULL Full Person Name	38
7.4	PN.SEARCH Searchable Name	38
7.5	TN Trivial Name	39
7.6	AD.BASIC Basic Address	40
7.7	AD.FULL Full Address	41
7.8	AD.SEARCH Searchable Address	43
7.9	TEL.PHONEMAIL Phone or E mail <Abstract>	43
7.10	TEL.PHONE Phone Number	44
7.11	TEL.EMAIL Email Address	45
7.12	TEL.URI Uniform Resource Identifier	45
8	Miscellaneous	46
8.1	ANY Unrestricted <Abstract>	46
8.2	ANY.LAB <Abstract>	46
8.3	ANY.CA.IZ <Abstract>	46
8.4	ANY.PATH <Abstract>	46
8.5	BL Boolean	47
8.6	LIST<x> List	47
8.7	SET<x> Set	47

1 Introduction

1.1 Background

This document contains the pan-Canadian Data Types specifications, for data types to be used in HL7 version 3 messages in Canada. For more information on the abstract data types referenced in this document, please consult the HL7 v3 International specification.

1.2 Pre-Adoption

Some of the proposed data type flavors below require adoption of data type properties that do not exist in release 1 of the HL7 v3 International data types specification. However, they are expected to be introduced as part of release 2 (R2) of the specification, currently under ballot through HL7 Inc and co-balloted by ISO. These properties are considered to be “pre-adopted” for use in Canada. Upon completion of universal balloting, a new version of the Canadian data type specifications will be balloted in Canada to align labels, vocabulary and other characteristics with the international standard. Where these properties appear within this specification, they will be flagged in **green** and noted in the table below.

Data Type	Component Name	Component Definition	More Information
ANY	specializationType	Declaration of data type flavour.	Section 1.3
II	use	Denotes the usage pattern for the identifier	Section 4.1
PIVL	frequency	Conveys how often the occurrence should or did happen	Section 3.15
AD	state.code	Coded state/province	Section 7.6
AD	country.code	Coded country	Section 7.6
PQ	originalText	Data before encoding	Section 6.11

R02.04.00 Note: Although R2 of the universal data types specification was balloted in September 2008, it was decided to adopt only minor R2 changes in the R02.04.00 release of the data types specifications noted as follows:

- Remove CD qualifier
- Deprecate II.BUSVER and II.PUBVER, Use II.BUS_AND_VER
- HC changed to OR for PN (PersonName) use attribute

This decision was made in consideration for the R02.04.00 timelines and human resource constraints. It is expected that more R2 changes will be adopted into future releases of this specification. As indicated above, a further reconciliation of R2 changes to this data type specifications document is expected as part of future updates.

1.3 Flavor Declaration

In general, an instance does not declare the data type being used for a particular attribute or property. The data type is assumed based on the RIM and the data type flavor is assumed based on the specification. These flavors are defined in this document. However, in some cases, data type substitution may allow the use of different data types and some specifications may allow for a choice of flavors. In these cases, the data type and/or flavor may need to be declared in the instance. For data type, this is done by using W3C's "xsi:type" capability.

For data type flavors, this is done by using the HL7's 'specializationType' property.

Example: <value xsi:type="TS" specializationType="TS.FULLDATETIME"/>

This means of conveying data type flavors as well as all flavors declared in this document are anticipated to be in release 2 of the universal data types specification.

If a data type flavor is declared, the receiver is not necessarily expected to validate that specific flavor against the specification. If validation is done, then an error can be returned if the received data does not validate against the datatype flavor specification.

For flavors that are specified in this document with a specific domain in their name, i.e. ANY.LAB, ANY.CA.IZ, PQ.LAB, etc., the specified domain should be considered the owner of the requirements for that datatype flavor. As such, they should be contacted if a change is required. NOTE: This does not stop any other domain from using that flavor unless otherwise noted. But such use should take into account the original purpose and requirements of the flavor.

1.4 XML ITS Representation of Datatype Properties

For some of the datatypes, currently IVL<x> and URG<x>, the properties listed in this specification are different than what would be found in an XML representation of the datatype. While the specification uses the business names for the properties, i.e. ~~lowClosed~~lowClosed, these names will not be found in the XML representation, i.e. <low inclusive="true"/>..

1.5 Null Properties

Generally, data type properties do not support null flavors, however, null flavors can be sent for any non-mandatory data type property. For the list of supported null flavor values, see the pan-Canadian Terminology Worksheet, **NullFlavor** concept domain.

1.6 Canadian Rules

In some cases, the data type flavor name is the same as the name of the universal data type (e.g. CS, ST, etc.). In these cases, the rules stated are considered to be generic constraints which apply to all uses of that data type within Canada, including in universal specifications. In other cases, the rules are outlined in the balance of this document.

1.7 Property Characteristics

Each data type property identifies whether and how that property is expected to be supported. Possible values, noted in *italics* in the data type specifications, are:

- **Not Permitted** The property must never appear in an instance. Applications may raise an error if they receive it.
- **Optional** The element may be supported by applications if they choose (and they must declare their decision in their conformance statement). If not supported, the element must be ignored if received. I.e. Applications **MUST NOT** raise an error upon receiving an unsupported optional element.
- **Required** The property must be supported by all applications (they are able to capture and/or display it), but won't necessarily always be present in an instance.
- **Populated** The property must be supported (able to be captured and/or displayed) and must always appear in the instance, but may appear with a flavor of null
- **Mandatory** The property must be supported and must always be present with a non-null value.

In general, if an attribute (or data type property) is not available (i.e. cannot be valued), then the rest of the data type properties (mandatory or not) should not be present. Exceptions to this rule will be documented in affected data types (e.g. CD).

In addition, data type properties may indicate maximum lengths. Systems will be able to capture and display an element as long as the maximum length, but it will not be able to send an element that is longer. Therefore a sending application must never send more than allowed by the specification and receiving applications. If longer data is truncated (within the maximum length) a warning message should be provided.

2 Best Practices

2.1 Health Level 7 Standard

This specification is based on the HL7 v3 standard, specifically the HL7 v3 data types R1 standard with some HL7 v3 data types R2 artifacts pre-adopted. All specifications and standards from the HL7 v3 UV data types standards have not been re-created in this document. It is expected that an implementer has an understanding of the HL7 UV materials along with this specification.

In this section, best practices will be documented which assist with the implementation of the data types specification and their use in message instances.

2.2 Intervals and Single Values

To ensure the models are reusable, some attributes are set to support an interval of values. This also provides support for a single value.

For any attribute set to IVL<some data type>, use xsi:type in the message instance set to the single data type.

For example, if an attribute is set to IVL<TS.DATETIME> in the message model definition, then in the message instance a single value (only) can be populated as follows:

```
<effectiveTime xsi:type="TS" specializationType="TS.DATETIME" value="200508021730-0800"/>
```

3 Dates + Timing

3.1 TS.DATE – Timestamp (Date Only, Partial Date Allowed)

Description	Specification
Only permits the “date” portion of the timestamp to be specified. Allows for partial dates.	This constraint on the timestamp data type only permits the “date” portion of the timestamp to be specified. Partial dates may be specified. The grammar for the data type is: YYYY[MM[DD]] (4-8 characters). Dates (or portions) specified must be valid dates.

Example:

```
<effectiveTime xsi:type="TS" specializationType="TS.DATE" value="200508"/>
```

3.2 TS.DATETIME – Timestamp (Date + Time, Partial Date + Time Allowed)

Description	Specification
Only permits the "date" and "time" portion of the timestamp to be specified. Does allow for partial dates and partial time, but timezone is mandatory if time is specified.	This constraint on the timestamp data type only permits the "date", "time" and "timezone" portions of the timestamp to be specified. Partial dates may be specified only if no time component is specified. Time may be omitted, and partial times may be specified. If a time is specified, the "timezone" portion of the timestamp must be specified. Local time is calculated as an offset of GMT (e.g. "-0800" for Pacific Time Zone, -0500 for Eastern Time Zone). The grammar for the date/time is: <u>YYYY[MM][DD][HH][MM][SS].[S][S][S] </u>[+/-ZZZZ] (24 characters). Dates and times (or portions) specified must be valid dates and times.

Example:

```
<effectiveTime xsi:type="TS" specializationType="TS. DATETIME" value="200508021730-0800"/>
```

Example for Find Candidates Query – deceased date of Jan 7th 2010

```
<deceasedTime><value xsi:type="TS" specializationType="TS.DATETIME"
value="20100107"/></deceasedTime>
```

3.3 TS.FULLDATEWITHTIME – Timestamp <Abstract>

Description	Specification
Requires at least the “date” portion of the timestamp to be fully specified. Allows for flexibility when specifying ControlAct.effectiveTime in wrappers and CMETs. If this data type is used in a message model, any conformant message instance must declare one of the constrained versions of TS.FULLDATEWITHTIME (TS.FULLDATE, TS.FULLDATETIME and TS.FULLDATEPARTTIME) for each repetition.	Each value sent over the wire must correspond to one of the following non-abstract data type flavor specifications defined in this document: TS.FULLDATE, TS.FULLDATETIME The specific data type flavour must be declared using specializationType.

3.4 TS.FULLDATE – Timestamp (Date Only, Partial Date Not Allowed)

Description	Specification
Only permits the “date” portion of the timestamp to be specified. Does not allow for partial dates.	This constraint on the timestamp data type only permits the “date” portions of the timestamp to be specified. The grammar for the data type is: YYYYMMDD (8 characters). Dates specified must be valid dates.

Example:

```
<effectiveTime xsi:type="TS" specializationType="TS.FULLDATE" value="20050802"/>
```

3.5 TS.FULLDATETIME – Timestamp (Date + Time, Partial Date + Time Not Allowed)

Description	Specification
Requires both the “date” and “time” portion of the timestamp to be fully specified, including timezone.	This constraint on the timestamp data type requires the “date”, “time”, and “timezone” portion of the timestamp to be fully specified and each is, therefore, <i>mandatory</i> . Partial dates may not be specified. Partial times may not be specified, except for precision component of time. The “timezone” portion of the timestamp must be specified in local time . Local time is calculated as an offset of GMT (e.g. “-0800” for Pacific Time Zone, “-0500” for Eastern Time Zone). The grammar for the date/time is: <u>YYYYMMDDHHMMSS[.S[S[S]]]]+/-ZZZZ</u> (19 to 24 characters). Dates and times specified must be valid dates and times.

Example:

```
<effectiveTime xsi:type="TS" specializationType="TS.FULLDATETIME" value="20050802173000-0800"/>
```

```
<effectiveTime xsi:type="TS" specializationType="TS.FULLDATETIME" value="20050802173000.1630-0800"/>
```

3.6 TS.FULLDATEPARTTIME – Timestamp (Date + Time, Partial Date Not Allowed, Partial Time allowed)

<u>Description</u>	<u>Specification</u>
Requires that the "date" portion of the timestamp to be fully specified. The "time" portion is optional.	<u>This constraint on the timestamp data type requires the "date". The Partial dates may NOT be specified. Time may be omitted, and partial times may be specified.</u> <u>If a time is specified, the "timezone" portion of the timestamp must be specified.</u> <u>The "timezone" portion of the timestamp must be specified in local time. Local time is calculated as an offset of GMT (e.g. "-0800" for Pacific Time Zone, "-0500" for Eastern Time Zone.</u> <u>The grammar for the date/time is:</u> <u>YYYYMMDD[HH[MM][SS[.S[S[S[S]]]]]] [+/- ZZZZ]</u> <u>(8 to 24 characters).</u> <u>Dates and times specified must be valid dates and times.</u>

Examples:

```
<effectiveTime xsi:type="TS" specializationType="TS.FULLDATEPARTTIME" value="200508021730-0800"/>
```

```
<effectiveTime xsi:type="TS" specializationType="TS.FULLDATEPARTTIME" value="20100102"/>
```

3.63.7 IVL<x> - Interval <Abstract>

Description	Specification
This data type is used when a continuous range needs to be expressed.	There are no generic constraints for the IVL datatype. Note: In the XML ITS, the “lowClosed” and “highClosed” properties are actually implemented with an “inclusive” attribute on the low and high attributes. Thus lowClosed equates to low/@inclusive and highClosed equates to high/@inclusive.

3.73.8 IVL<TS.DATE> – Partial Date Range

Description	Specification
This data type is used when a date range needs to be conveyed where neither the start nor end date will always be known to the complete date.	The properties “lowClosed”, “center” and “highClosed” are <i>not permitted</i> . “low” property identifies the start for the range, is <i>required</i> and must follow the TS.DATE specification. “high” property identifies the end for the range, is <i>required</i> and must follow the TS.DATE specification. “width” property identifies the duration of the range, is <i>required</i> and must follow the PQ.TIME specification. Furthermore, units should be limited to days, weeks, months or years. For a non-null attribute, two (and only two) of the identified properties must be <i>populated</i> . That is, implementations must support all of the high, low and width properties; and in a message instance each repetition must include exactly two of these, though either or both of them can be a nullFlavor. Null flavors are permitted for “low”, “high” and “width”, but are prohibited from being set to PINF or NINF

Example:

“from March 01, 2000 to July 2000” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">  
  <low specializationType="TS.DATE" value="20000301"/>  
  <high specializationType="TS.DATE" value="200007"/>  
</effectiveTime>
```

Example:

“Beginning March, 2000, not yet ended” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">  
  <low specializationType="TS.DATE" value="200003"/>  
  <high specializationType="TS.DATE" nullFlavor="NA"/>  
</effectiveTime>
```

Formatted: Bullets and Numbering

3.83.9 URG<x> - Interval <Abstract>

Description	Specification
This data type is used when a continuous range needs to be expressed.	There are no generic constraints for the URG datatype. Note: In the XML ITS, the “lowClosed” and “highClosed” properties are actually implemented with an “inclusive” attribute on the low and high attributes. Thus lowClosed equates to low/@inclusive and highClosed equates to high/@inclusive. originalText is a pre-adopted data type component of type ST.

3.93.10 URG<TS.DATE> – Uncertain Date

Description	Specification
This data type is used when an occurrence is tied to a specific date, but the actual date is not known, merely the range of dates within which the date falls. This differs from IVL<TS.DATE> in that it refers to a single occurrence rather than a period covering multiple days.	The properties “lowClosed” and “highClosed” are <i>not permitted</i>. “low” property identifies the earliest possible date that could be a match, is <i>required</i> and must follow the TS.DATE specification. “high” property identifies the latest possible date that could be a match, is <i>required</i> and must follow the TS.DATE specification. “center” property is <i>required</i>, identifies the mid-point within the range and must follow the TS.DATE specification. “width” property identifies the size of the uncertain period, is <i>required</i> and must follow the PQ.TIME specification. For a non-null attribute, two (and only two) of the identified properties must be <i>populated</i>. That is, implementations must support all of the high, low and width properties; and in a message instance each repetition must include exactly two of these, though either or both of them can be a nullFlavor. Null flavors are permitted for “low”, “high”, “center” and “width”, but are prohibited from being set to PINF or NINF

Example:

“April, 2004 +/- 2 months” would be specified as:

```
<effectiveTime xsi:type="URG_TS"><center xsi:type="TS" specializationType="TS.DATE"
value="200404"/><width xsi:type="PQ" specializationType="PQ.TIME" value="4" unit="m"/>
</effectiveTime>
```

3.103.11 **IVL<TS.DATETIME> – Partial Date + Time Range**

Description	Specification
Used when a time range needs to be conveyed where the boundaries of the range can be imprecise.	The properties “lowClosed”, “center” and “highClosed” are <i>not permitted</i>. “low” property identifies the start for the range, is <i>required</i> and must follow the TS.DATETIME specification. “high” property identifies the start for the range, is <i>required</i> and must follow the TS.DATETIME specification. “width” property identifies the duration of the range, is <i>required</i> and must follow the PQ.TIME specification. For a non-null attribute, two (and only two) of the identified properties must be <i>populated</i>. That is, implementations must support all of the high, low and width properties; and in a message instance each repetition must include exactly two of these, though either or both of them can be a nullFlavor. Null flavors are permitted for “low”, “high” and “width”, but are prohibited from being set to PINF or NINF

Example:

“from March 01, 2000 @ 3:00 PM to July 15, 2000 @ 4:00 AM (Alberta time)” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">
  <low specializationType="TS.DATETIME" value="200003011500-0700"/>
  <high specializationType="TS.DATETIME" value="200007150400-0700"/>
</effectiveTime>
```

3.113.12 **IVL<TS.FULLDATEWITHTIME> – Precise Date Range – <Abstract>**

Description	Specification
Requires at least the “date” portion of the timestamp to be fully specified. Allows for flexibility when specifying ControlAct.effectiveTime in wrappers and CMETs This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the constrained version of IVL<TS.FULLDATEWITHTIME>: IVL<TS.FULLDATE> IVL<TS.FULLDATETIME> for each repetition.	Each value sent over the wire must correspond to one of the following non-abstract data type flavor specifications defined in this document: IVL<TS.FULLDATE> and IVL<TS.FULLDATETIME> The specific data type flavour must be declared using specializationType. The constraints for the specific declared version must be followed.

3.123.13 IVL<TS.FULLDATE> – Precise Date Range

Description	Specification
This data type is used when a date range needs to be conveyed with precise start and/or end dates.	The properties “lowClosed”, “highClosed” and “center” are <i>not permitted</i>. “low” property identifies the start for the time-period, is <i>required</i> and must follow the TS.FULLDATE specification. “high” property identifies the end date for the time-period, is <i>required</i> and must follow the TS.FULLDATE specification. “width” property identifies the duration of the range, is <i>required</i> and must follow the PQ.TIME specification. Furthermore, units should be limited to days, weeks, months or years. For a non-null attribute, two (and only two) of the identified properties must be <i>populated</i>. Implementations must support all of the high, low and width properties; and in a message instance each repetition must include exactly two of these, though either or both of them can be a nullFlavor. Null flavors are permitted for “low”, “high” and “width”, but are prohibited from being set to PINF or NINF

Example:

“from March 01, 2000 to July 15, 2000” would be specified as:

```
<effectiveTime xsi:type="IVL_TS" specializationType="IVL.TS.FULLDATE">  
  <low specializationType="TS.FULLDATE" value="20000301"/>  
  <high specializationType="TS.FULLDATE" value="20000715"/>  
</effectiveTime>
```

3.133.14 **IVL<TS.FULLDATETIME>** – Precise Date-Time Range

Description	Specification
Used when a time range needs to be conveyed with precise start and/or end times.	The properties “lowClosed”, “highClosed”, “center” and “width” are <i>not permitted</i> . “low” property identifies the start for the time-period, is <i>populated</i> and must follow the TS.FULLDATETIME specification. “high” property identifies the end date for the time-period, is <i>populated</i> and must follow the TS.FULLDATETIME specification. Null flavors are permitted for “low” and “high”, but are prohibited from being set to PINF or NINF

Example:

“from March 01, 2000 @ 3:30:00 PM to July 15, 2000 @ 4:30:00 AM, Ontario time” would be specified as:

```
<effectiveTime xsi:type="IVL_TS" specializationType="IVL.TS.FULLDATETIME">  
  <low specializationType="TS.FULLDATETIME" value="20000301153000-0500"/>  
  <high specializationType="TS.FULLDATETIME" value="20000715043000-0500"/>  
</effectiveTime>
```

3.143.15 **IVL.LOW<TS.DATE>** – Partial Start Date

Description	Specification
Used to convey the start date of a range only. Allows for partial dates.	The properties “high”, “center”, “width”, “lowClosed”, “highClosed” are <i>not permitted</i> . “low” property identifies the ‘start date’, is <i>mandatory</i> and must follow the TS.DATE specification.

Example:

“Began July, 2005” would be specified as:

```
<effectiveTime xsi:type="IVL_TS" specializationType="IVL.LOW">  
  <low specializationType="TS.DATE" value="200507"/>  
</effectiveTime>
```

3.153.16 IVL.LOW<TS.FULLDATE> – Precise Start Date

Description	Specification
Used to convey the start date of a range only. Does not allow for partial dates.	The properties “high”, “center”, “width”, “lowClosed”, “highClosed” are <i>not permitted</i> . “low” property identifies the ‘start date’, is <i>mandatory</i> and must follow the TS.FULLDATE specification.

Example:

“Begin July 6, 2005” would be specified as:

```
<effectiveTime xsi:type="IVL_TS" specializationType="IVL.LOW_TS.FULLDATE">  
  <low specializationType="TS.FULLDATE" value="20050706"/>  
</effectiveTime>
```

3.163.17 IVL.HIGH<TS.FULLDATE> – Precise End Date

Description	Specification
This data type is used when both a start and end date may exist, but only the end date should be captured. Does not allow for partial dates.	The properties “low”, “center”, “width”, “lowClosed”, “highClosed” are <i>not permitted</i> . “high” property identifies the ‘end date’, is <i>mandatory</i> and must follow the TS.FULLDATE specification.

Example:

“End, September 15, 2003” would be specified as:

```
<effectiveTime xsi:type="IVL_TS" specializationType="IVL.HIGH_TS.FULLDATE">  
  <high specializationType="TS.FULLDATE" value="20030915"/>  
</effectiveTime>
```

3.173.18 IVL.WIDTH<TS.FULLDATE> – Duration

Description	Specification
This data type is used to convey duration where the specific start and end days are not needed.	The properties “low”, “high”, “center”, “lowClosed”, “highClosed” are <i>not permitted</i> . “width” property identifies the ‘duration’, is <i>mandatory</i> and must follow the PQ.TIME specification.

Example:

“23 days” would be specified as:

```
<effectiveTime xsi:type="IVL_TS" specializationType="IVL.WIDTH_TS.FULLDATE">  
  <width specializationType="PQ.TIME" value="23" unit="d"/>  
</effectiveTime>
```

3.183.19 PIVL<TS.DATETIME> – Periodic Interval of Time

Description	Specification
Allows communication of a frequency.	The “phase”, “period”, “alignment” and “institutionSpecified” properties are <i>not permitted</i> . The frequency property conveys how often the occurrence should or did happen and is <i>mandatory</i> . It contains two <i>mandatory</i> properties: Numerator is <i>mandatory</i> , identifies the number of repetitions within a given time period and must follow the INT.NONNEG specification. Denominator is also <i>mandatory</i> , identifies the time period and must follow the PQ.TIME specification.

Example:

“3 times per day” would be specified as:

```
<effectiveTime xsi:type="PIVL_TS">  
  <frequency>  
    <numerator xsi:type="INT" specializationType="INT.NONNEG" value="3"/>  
    <denominator xsi:type="PQ" specializationType="PQ.TIME" value="1" unit="d"/>  
  </frequency>  
</effectiveTime>
```

3.193.20 *PIVL<TS.FULLDATETIME> – Precise Periodic Interval of Time*

Description	Specification
OPEN ISSUE: This datatype is currently used in SDL models and is intended to portray an informal representation of timing. E.g. "What hours a facility is open until" or "What days is a service offered" In this release of the Data type specification, the description, specification and examples along with implementation guidance are not addressed. This topic will be discussed in the appropriate SCWGs to ensure that the use cases are properly documented and that the requirements are clearly understood prior to defining the next steps. One example of a next step is, based on current understanding; this may require the use of a new flavour GTS.TEXT instead of PIVL<TS.FULLDATETIME>. Implementers are encouraged to bring use cases and requirements to the SCWG forum in the meanwhile.	

3.203.21 **GTS.BOUNDEDPIVL – General Timing Specification (Time-range and Frequency)**

Description	Specification
This data type is used to convey two pieces of information: The overall time-period when something occurred (or is to occur), as well as how often it should/did occur within that time-period	This data type is transmitted via two mandatory repetitions. The first <i>mandatory</i> repetition must comply with the IVL<TS.FULLDATE> specification and indicates the overall time-period. The operator is not permitted in the first repetition. The second <i>mandatory</i> repetition must comply with the PIVL<TS.DATETIME> specification and indicates the frequency. It must have an additional <i>mandatory</i> “operator” property with a value of “I” for intersection, indicating that the time of occurrence is the combination of both the outer time interval and the inner frequency Note: SXPR_TS is the mechanism for implementing the GTS data type. Value is <i>not permitted</i>. Operator = “I”

Example:

“3 times/day beginning August 3rd 2005 for 3 weeks” would be specified as:

```
<effectiveTime xsi:type="SXPR_TS" specializationType="GTS.BOUNDEDPIVL">
  <comp xsi:type="IVL_TS" specializationType="IVL_TS.FULLDATE">
    <low specializationType="TS.FULLDATE" value="20050803"/>
    <width specializationType="PQ.TIME" value="3" unit="wk"/>
  </comp>
  <comp xsi:type="PIVL_TS" operator="I">
    <frequency>
      <numerator xsi:type="INT" specializationType="INT.NONNEG" value="3"/>
      <denominator xsi:type="PQ" specializationType="PQ.TIME" value="1" unit="d"/>
    </frequency>
  </comp>
</effectiveTime>
```

4 Identifiers + Codes

4.1 II – Instance Identifier <Abstract>

Description	Specification
The instance identifier data type is used to communicate all "identifiers", including identifiers for people, records, versions, messages, etc. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the constrained version of II (e.g. II.OID) for each repetition.	The rules stated here for II apply to all flavors of II. II.root is <i>mandatory</i> and is constrained to either an OID or a UUID (GUID). It has a maximum length of 200 characters. II.extension has a maximum of ~~520~~ characters. It is considered case-sensitive for comparison purposes. Non-case-sensitive identifiers should be exchanged in upper-case. There are no further restrictions on the extension component. II.assigningAuthorityName is <i>not permitted</i>. AssigningAuthorityName is not needed for processing purposes. It is also not needed when performing identifier matches, which is the sole purpose of the II data type. II.use is a pre-adopted data type component of type ST and has a maximum of 6 characters. A list of values permitted for II.use can be found in the Terminology Worksheet, IdentifierUse concept domain. Note that each Data type flavour defines which, if any, values for II.use are permitted for that flavour.

4.2 II.BUSVER – Instance Identifier (Business with Version) - deprecated



Note to Reader: The use of this data type has been reviewed and the recommendation is to use II.BUS_AND_VER instead. Therefore, the use of this data type is being *deprecated*.

Description	Specification
Communicates a “business” identifier as well as a “version” of that identifier to assist in collision detection. Business context: This is an identifier which remains consistent as an object evolves. i.e. The identifier does not change when the object undergoes state transitions. Examples include prescription numbers, encounter numbers, lab result numbers, etc. Version context: This is used when referencing specific versions of a particular record.	“displayable” property is <i>not permitted</i> “root” property is <i>mandatory</i> and may contain an OID or a GUID “extension” property is <i>required</i> , and if present, represents the ‘human-readable’ portion of the identifier. The extension may only be present if the root is an OID. “use” property is mandatory and must be “BUS”

Example:

“The second time a prescription number of R092377 issued by a pharmacy with the OID 2.16.840.1.113883.19.335.72 is updated” might look like this:

```
<id xsi:type="II" specializationType="II.BUSVER" root="2.16.840.1.113883.19.335.72.7"
extension="R092377" use="BUS" version="2"/>
```

4.3 II.OID – Instance Identifier (OID)

Description	Specification
Used where an identifier is known to be constrained to only be an OID. E.g. Conformance profile identifiers	“use” property is mandatory and must be “BUS” “extension” and “displayable” properties are <i>not permitted</i> . “root” property is <i>mandatory</i> and constrained to contain only an OID.

Example:

“A profile id representing a jurisdictional conformance profile” might look like this:

```
<id xsi:type="II" specializationType="II.OID" root="2.16.840.1.113883.19.335.15.8885.21197"
use="BUS"/>
```

4.4 II.PUBLIC – Instance Identifier (Public)

Description	Specification
This is used to convey common public identifiers such as provider license numbers, jurisdictional PHNs, etc.	"use" property is mandatory and must be "BUS" "root" property is <i>mandatory</i> and contains the OID for the identifier type. It must be an OID. "extension" property is <i>mandatory</i> and contains the human-readable identifier. "displayable" property is <i>mandatory</i> and must have a value of "true".

Example:

"The Alberta PHN (ULI) 12356-7890" would be specified as:

```
<id xsi:type="II" specializationType="II.PUBLIC" root="2.16.840.1.113883.4.20"
extension="123567890" use="BUS" displayable="true"/>
```

4.5 II.PUBLICVER – Instance Identifier (Public with Version) - deprecated

	Note to Reader: This datatype is not currently being used by any models therefore; the use of this data type is being <i>deprecated</i> .
---	--

Description	Specification
This is used to convey common public identifiers such as provider license numbers, jurisdictional PHNs, etc. The identifier also contains a "version" of that identifier to assist in collision detection.	"use" property is <i>not permitted</i> "root" property is <i>mandatory</i> and contains the OID for the identifier type. It must be an OID. "extension" property is <i>mandatory</i> and contains the human-readable identifier. "displayable" property is <i>mandatory</i> and must have a value of "true". "version" property is <i>mandatory</i> and must be unique for each change in the underlying record. No specific guidance on numbering scheme is provided.

4.6 II.TOKEN – Instance Identifier (Token)

Description	Specification
This is used to convey "internal" computer-generated on-the-fly identifiers, such as security tokens. These identifiers are not intended for human display.	"extension", "use" and "displayable" properties are <i>not permitted</i> . "root" property is <i>mandatory</i> and constrained to contain only a UUID (GUID).

Example:

"A temporary resolved-patient identifier" might look like this:

```
<id xsi:type="II" specializationType="II.TOKEN" root="64AC5370-D851-2D2F-2A18-
78FBB73BFED9"/>
```

4.7 II.BUS_AND_VER – Instance Identifier (Business and Version)

<Abstract

Description	Specification
This type is used when an attribute is intended to be used to convey two types of identifiers – a business identifier and a version identifier. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the constrained version of II.BUS_AND_VER (e.g. II.VER and II.BUS) for each repetition	Each value sent over the wire for the two repetitions must correspond to each of the non-abstract data type flavor specifications defined in this document - II.BUS and II.VER

4.8 II.VER – Instance Identifier (Version)

Description	Specification
This is used when referencing specific versions of a particular record.	“displayable” and “extension” properties are <i>not permitted</i>. “root” property is <i>mandatory</i> and must contain a GUID “use” property is <i>mandatory</i> and must be “VER”

Example:

“The id of a particular allergy record version” might look like this:

```
<id xsi:type="II" specializationType="II.VER" root="64AC5370-D851-2D2F-2A18-78FBB73BFED9" use="VER"/>
```

4.9 II.BUS – Instance Identifier (Business)

Description	Specification
Communicates a “business” identifier. This is an identifier which remains consistent as an object evolves. i.e. The identifier does not change when the object undergoes state transitions. Examples include prescription numbers, encounter numbers, lab result numbers, etc.	“displayable” property is <i>not permitted</i> “root” property is <i>mandatory</i> and may contain an OID or a GUID “extension” property is <i>required</i> , and if present, represents the ‘human-readable’ portion of the identifier. The extension may only be present if the root is an OID. “use” property is mandatory and must be “BUS”

Example:

“A prescription number R092377 issued by a pharmacy with the OID 2.16.840.1.113883.19.335.72” might look like this:

```
<id xsi:type="II" specializationType="II.BUS" root="2.16.840.1.113883.19.335.72.7"
extension="R092377" use="BUS"/>
```

4.10 CS – Coded Simple

Description	Specification
As per the HL7 data type	Constraints are the same as for the CD data type, with only the ‘code’ attribute permitted, which is <i>mandatory</i> . Note: This data type should only appear in specifications where the corresponding RIM attribute or Abstract Data type property also has a data type of CS. i.e. Other coded data types should never be constrained to CS.

Example:

“A mood code indicating an event” would be specified as:

```
<... moodCode="EVN" ...>
```

Example for Find Candidates Response Identified Entity.statusCode of Active

A status code of active

```
<statusCode code="active"/>
```

4.11 CV – Coded Value

Description	Specification
As per the HL7 data type	Constraints are the same as for the CD data type, with only the following properties permitted: “code”, “codeSystem” and “originalText”

Example:

“a gender of female” would be specified as:

```
<administrativeGenderCode code="F" codeSystem="2.16.840.1.113883.5.1"/>
```

Example for Find Candidates Query

```
<administrativeGender><value code="F" codeSystem="2.16.840.1.113883.5.1"/>
</administrativeGender>
```

4.12 CE – Coded with Equivalents

	Note to Reader: The use of this data type has been reviewed and the recommendation is to use CD instead. Namely because the only difference between CD and CE was the use of the qualifier property, which is no longer being used for this release.
---	---

Description	Specification
As per the HL7 data type	Constraints are the same as for the CD data type, except the “qualifier” property is <i>not permitted</i> .

Example:

“A diagnosis of SARs (prior to the issuing of an ICD10 code), using a local code” would be specified as:

```
<value nullFlavor="OTH"><originalText>SARs</originalText><translation code="AB99123788"
codeSystem="2.16.840.1.113883.19.881.57"></translation></value>
```

4.13 CD – Concept Descriptor

Description	Specification
As per the HL7 data type	If a CD instance is null, the codeSystem and originalText properties are still permitted, but not <i>mandatory</i>. The “codeSystemName”, “codeSystemVersion”, “qualifier” and “displayName” properties are <i>not permitted</i>. “code” property is constrained to ST with a maximum length of 200 characters. (For elements with value-sets denoted as “fixed at ballot”, implementers may constrain themselves to only supporting the length of the longest code within that value-set). “codeSystem” is constrained to a length of 100 characters and must correspond to a codeSystem OID registered in the HL7 OID registry. “originalText” property is constrained to ST with a maximum length of 150 characters The rules for supporting the above properties vary depending on whether the element has a coding strength of CWE (Coded with extensibility) or CNE (Coded with no extensibility). For CWE, “code”, “codeSystem” and “originalText” properties are <i>required</i>. A constraint exists that at least one of “code” and “originalText” must be present and non-null, and that if “code” is present, then “codeSystem” is <i>mandatory</i>. For CNE, “code” and “codeSystem” are <i>mandatory</i> (meaning they must be non-null if the overall attribute or property is non-null). If the element is null with a flavor of OTH (Other), then the “originalText” property is <i>mandatory</i> (and no other properties may be specified). The “translation” property is <i>required</i> and has the same data type as the root data type. I.e. CD may have CD translations. CE may only have CE translations. Translations are limited to one level of nesting. I.e. Tracking translations of translations will not be supported. The number of translations is limited to 10. Translations are intended to give the translation, not the fact that translation occurred. Therefore, it would not make sense to have a NullFlavor for this property/component. Qualifiers are permitted. There are no constraints on translations. Fundamentally either the base code or one of the translations must be from the approved value set, or the whole data type must have a null flavor of OTH or the coding strength must be CWE.

Example:

"A diagnosis of moderate to severe psoriasis of the face with sudden onset" would be specified as:

```
<value code="402320002" codeSystem="2.16.840.1.113883.6.96"><originalText>Acute, moderate facial psoriasis</originalText></value>
```

A post-coordinated coding of "procedure to replace a left hip by inserting a prosthesis" would be:

```
<value  
code="397956004:363704007=24136001:272741003=7771000{363699004=304120007,260686004=  
257867005}" codeSystem="2.16.840.1.113883.6.96" originalText="procedure to replace a left hip by  
inserting a prosthesis"/>
```

397956004|prosthetic arthroplasty of the hip

363704007|Procedure site

24136001|Hip joint structure

272741003|laterality

7771000|left

363699004|direct device

304120007|Total hip replacement prosthesis

260686004|method

257867005|insertion-action

A post-coordinated UCUM would be:

```
<value code="[iU]/d" codeSystem="2.16.840.1.113883.6.8" originalText="iU/day"/>
```

```
<value code="g/12.h" codeSystem="2.16.840.1.113883.6.8" originalText="g/12 hr"/>
```

Example of null flavor of UNC (unencoded):

```
<value nullFlavor="UNC">
```

```
<originalText>disorder of respiratory system</originalText>
```

```
</value>
```

4.14 CD.LAB – Lab Concept Descriptor

Description	Specification
As per the CD data type.	All constraints specified for CD applies except where noted in the following: The “displayName” property is <i>permitted</i> . Only CNE coding strength if permitted. CWE is not permitted; therefore code and codeSystem are Mandatory. The “translation” property is <i>Optional</i> and has the same data type as the root data type. i.e. CD.LAB. Only code and codeSystem are permitted and are all Optional. All other attributes are not permitted. All other rules and constraints as specified by the CD datatype apply.

Example:

A diagnosis of “moderate to severe psoriasis of the face with sudden onset” would be specified as:

```
<value code="402320002" codeSystem="2.16.840.1.113883.6.96" displayName=" moderate to severe psoriasis of the face with sudden onset"></value>
```

Or, if the text that was displayed to the user was different:

```
<value code="402320002" codeSystem="2.16.840.1.113883.6.96" displayName=" moderate to severe psoriasis of the face with sudden onset"><originalText>Acute, moderate facial psoriasis</originalText></value>
```

If the original code had been translated into another coding system, that would be specified as:

```
<value code="11084-1" codeSystem="2.16.840.1.113883.6.1" displayName="Reagin antibody; Serum"><originalText>VDRL</originalText><translation code="ABCD" codeSystem="2.16.840.1.113883.6.96"/></value>
```

5 Text & Documents

5.1 ST – String

Description	Specification
As per the HL7 data type	"language" property is <i>not permitted</i> Maximum length of the string is determined by the message, document or other (e.g. data type) specification that includes this data type for a specified attribute or property.

Example:

```
<name>Royal Alexandra Hospital</name>
```

5.2 ST.LANG – String with Language

Description	Specification
A string where there is a requirement to expose the language of the string. Effectively this means that attributes intended to convey the language of text must be explicitly declared as such	"language" property is <i>mandatory</i> For definition of "language" property, see ED – Encapsulated Data data type. Maximum length is determined by the referencing specification. Values are drawn from one 2-letter ISO 639-3 language code and one 2-letter ISO 3166-1 country code and are limited to "en-CA" or "fr-CA".

Example:

```
<name language="en-CA">Royal Alexandra Hospital</name>
```

5.3 ED – Encapsulated Data <Abstract>

Description	Specification
This defines generic rules for all uses of the ED data type within Canada. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the constrained version of ED (e.g. ED.DOC) for each repetition.	The “integrityCheck” and “thumbnail” properties are <i>not permitted</i>. “compression” property indicates the compression algorithm used. If present, it is restricted to having a value of DF (Deflate) or GZ (gzip) “mediaType” property indicates the type of data being conveyed and is <i>mandatory</i>. A list of values permitted for mediaType can be found in the Terminology Worksheet, x_DocumentMediaType value set. “charset:” property is mandatory. <u>Charset is not explicitly represented in the XML representation of the ED data type but the value of charset is to be inferred from the encoding used in the XML entity in which the ED value is contained. However, p</u> Please note that “language” property indicates the principle language of the text or encapsulated document. The principles of the code domain of this attribute are specified by the Internet standard at URL "http://www.isi.edu/in-notes/rfc1766.txt". It is a set of pre-coordinated pairs of one 2-letter ISO 639 language code and one 2-letter ISO 3166 country code. If not specified, the assumption is that the text/document uses the language associated with the overall message. Example values include “en-CA” for English and “fr-CA” for French. “representation” property allowable codes are TXT (text) and B64 (base 64). Needs to be asserted for each type of media that is included in the ED data types (e.g. JPG must be B64, etc.). Note that the “representation” property does not appear in the abstract data types specification, but does exist in the XML ITS specification. Document content: If the content is compressed (the compression attribute is present), or if the content is a PDF document (mediaType=“application/pdf”), then the content must be base 64-encoded. Otherwise, the content will be sent as directly embedded text or XML. (NOTE: This means that uncompressed HTML must be XHTML compliant.) Content is restricted to a maximum of 1 megabyte after compression and base 64 encoding.

5.4 ED.DOC – Encapsulated Data (Document)

Description	Specification
This allows for the encapsulation of a document within an attribute.	Includes all constraints from “ED”. “compression” and “language” properties are <i>required</i> . Content of the document is <i>mandatory</i> . “reference” if specified, is restricted to tel.uri “mediaType” A list of values permitted for mediaType can be found in the Terminology Worksheet, x_DocumentMediaType value set. If the mediaType is multipart/x-hl7-cda-level-one, the content must be a compliant HL7 CDA document.

Example:

HTML content would be specified as:

```
<text mediaType="text/html" language="fr-CA" xsi:type="ED" specializationType="ED.DOC" ><html> .  
.. </html></text>
```

5.5 ED.DOCORREF – Encapsulated Data (Document or Reference) <Abstract>

Description	Specification
This allows for the encapsulation of a document or reference to a document within the HL7 instance.	The instance must comply with either the ED.DOC or ED.DOCREF specification.

5.6 ED.DOCREF – Encapsulated Data (Document Reference)

Description	Specification
This allows embedding a reference to a document that can be retrieved elsewhere	Includes all constraints from “ED”. In addition, the “content” property is <i>not permitted</i> . “compression” and “language” properties are <i>required</i> . “reference” is <i>mandatory</i> and data type is restricted to tel.uri. “mediaType” A list of values permitted for II.use can be found in the Terminology Worksheet, x_DocumentMediaType value set. If the mediaType is multipart/x-hl7-cda-level-one, the content must be a compliant HL7 CDA document.

Example:

A reference to a monograph might look like this:

```
<text mediaType="application/pdf" specializationType="ED.DOCREF">  
<reference use="DIR" value=ftp://monographs.somewhere.gov.ca/12345.pdf  
specializationType="TEL.URI"/></text>
```

5.7 ED.SIGNATURE – Encapsulated Data (Digital Signature)

Description	Specification
This allows incorporation of a digital signature as part of a message.	Includes all constraints from “ED”. In addition, the properties “compression”, “language” and “reference” are <i>not permitted</i>. The mediaType must be “text/xml” The content is <i>mandatory</i> and must conform to the XML digital signature specification. Please refer to the W3C XML Signature Syntax and Processing recommendation at http://www.w3.org/TR/xmldsig-core/#sec-Schema. NOTE: Additional pan-Canadian standards effort is anticipated to provide guidance on how digital signatures should be implemented within Canadian v3 instances.

Example:

A digital signature might look like this:

```
<text xsi:type="ED" specializationType="ED.SIGNATURE" mediaType="text/xml">
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#"> . . .
    <SignedInfo>...</SignedInfo>
  </Signature></text>
```

6 Numeric

6.1 INT.POS – Positive Integer

Description	Specification
This is used to convey an integer number that is greater than or equal to “1”.	Maximum length is restricted to 10 digits Value must be >= 1

Example:

```
<sequenceNumber xsi:type="INT" specializationType="INT.POS" value="27"/>
```

Example for Find Candidates Query where the multiple birth order number is two – that is, the second child

```
<multipleBirthOrderNumber><value xsi:type="INT" specializationType="INT.POS" value="2"/></multipleBirthOrderNumber>
```

6.2 INT.NONNEG – Non-negative Integer

Description	Specification
This is used to convey an integer number that is greater than or equal to “0”.	Maximum length is restricted to 10 digits Value must be >= 0

Example:

```
<sequenceNumber xsi:type="INT" specializationType="INT.NONNEG" value="27"/>
```

6.3 REAL.COORD – Real Number (Coordinate)

Description	Specification
A real number intended for use in conveying geographic coordinate values	Maximum length is restricted to 9 characters with 4 characters before the decimal and 4 characters after the decimal

Example:

```
<value xsi:type="REAL" specializationType="REAL.COORD" value="137.2895"/>
```

6.4 REAL.CONF – Real Number (Confidence)

Description	Specification
A real number intended for use in conveying confidence level values in registries NOTE: As of R02.04.02, this datatype flavor is also used for expressing percentages. It is expected that the flavor name may change in a future release.	Maximum length is restricted to 6 characters with 1 character before the decimal and 4 characters after the decimal. Value must be <=1 and >=0

Example:

A confidence level value from a registry of .2895 –

```
<value xsi:type="REAL" specializationType="REAL.CONF" value="0.2895"/>
```

The GST as of August 2009 –

```
<value xsi:type="REAL" specializationType="REAL.CONF" value="0.05"/>
```

The following fragment shows the complete ObservationEvent class for the Find Candidates Response

```
<subjectOf typeCode="SBJ">
  <observationEvent classCode="OBS" moodCode="EVN">
    <code code="PHCM" codeSystem="2.16.840.1.113883.2.20.5.2"/>
    <value xsi:type="REAL" flavorId="REAL.CONF" value="0.2895"/>
  </observationEvent>
</subjectOf>
```

6.5 MO.CAD – Canadian Money

Description	Specification
Used to convey monetary values in Canadian currency	The “value” property contains the actual dollar amount, is <i>mandatory</i> and is constrained to 11 digits above the decimal and 2 digits beneath the decimal (<u>9999999999.99</u>) <i>Note that</i> {fractions of a cent are not supported} The “currency” property is <i>mandatory</i> and must have a value of “CAD”, referenced from ISO 4217.

Example:

“137 dollars, 15 cents Canadian” would be specified as:

```
<amt xsi:type="MO" specializationType="MO.CAD" value="137.15" currency="CAD"/>
```

6.6 PQ.BASIC – Basic Physical Quantity

Description	Specification
Used to communicate measurements using common measurement units encountered in healthcare and corresponding value	The “translation” property is <i>not permitted</i> as support for non-UCUM units will not be supported. The “value” property contains the actual amount, is <i>mandatory</i> and is constrained to 11 digits above the decimal and 2 digits beneath the decimal (9999999999.99). <u>The exponent syntax (E.g. 2.2345374E7) is not permitted.</u> The “unit” property is <i>required</i> and is constrained to the list of values found in the Terminology Worksheet, x_BasicUnitsOfMeasure value set. The unit property must be specified unless the units of measurement are “each”. For example, “packages”, “packs”, “capsules”, “tablets”, etc. are not sent using this property. These concepts should be explicit in the code identifying the type of item being measured. For example, if the drug name is “100mg Acetaminophen tablets”, the quantity would be “1” (or “2” or whatever) with no units specified. The unit would be inferred from the drug name as “tablets”.

Example:

22.35 millmoles per litre

```
<value xsi:type="PQ" specializationType="PQ.BASIC" value="22.35" unit="mmol/l"/>
```

6.7 PQ.DRUG – Drug Physical Quantity

Description	Specification
Used to convey quantities of drug products.	This is identical to the PQ.BASIC (Physical Quantity - Basic) data type except that the units are restricted to the list of values found in the Terminology Worksheet, x_DrugUnitsOfMeasure value set. As well, the value property is constrained to 4 digits beneath the decimal. <u>The “value” property contains the actual amount, is <i>mandatory</i> and is constrained to 11 digits above the decimal and 6 digits beneath the decimal (9999999999.999999). The exponent syntax (E.g. 2.2345374E7) is not permitted.</u>

Formatted: Font: Bold

Example:

“10 tablets” would be specified as:

```
<doseQuantity xsi:type="PQ" specializationType="PQ.DRUG" value="10"/>
```

6.8 PQ.TIME – Time Quantity

Description	Specification
Used to convey durations of time	This is identical to the PQ.BASIC (Physical Quantity - Basic) data type except that the units are <i>mandatory</i> and have been constrained to the list of values found in the Terminology Worksheet, x_TimeUnitsOfMeasure value set.

Example:

"14 days" would be specified as:

```
<width xsi:type="PQ" specializationType="PQ.TIME" value="14" unit="d"/>
```

6.9 PQ.HEIGHTWEIGHT – Height or Weight Quantity

Description	Specification
Used to express measurements of human height and weight The data type units are constrained to the specific use-case of human height and weight resulting in a fairly narrow number of options to be supported. Future use-cases resulting in the need to communicate other types of units of measure may require the creation of a new data type flavor. The purpose of creating multiple flavors is to make it clear what units must be supported for a given attribute and to ensure that systems are not obligated to use units of measure that are inappropriate.	This is identical to the PQ.BASIC (Physical Quantity - Basic) data type except that the units are <i>mandatory</i> and have been restricted to the list of values found in the Terminology Worksheet, x_HeightOrWeightObservationUnitsOfMeasure value set.

Example:

"30 kilograms" would be specified as:

```
<value xsi:type="PQ" specializationType="PQ.HEIGHTWEIGHT" value="30" unit="kg"/>
```

6.10 PQ.DISTANCE – Distance Quantity

Description	Specification
Used to express measurements of distance, specifically for geographic distances	This is identical to the PQ.BASIC (Physical Quantity - Basic) data type except that the units are <i>mandatory</i> and have been constrained to the list of values found in the Terminology Worksheet, x_DistanceObservationUnitsOfMeasure value set.

Example:

22 kilometres" would be specified as:

```
<value xsi:type="PQ" specializationType="PQ.DISTANCE" value="22" unit="km"/>
```

6.11 PQ.LAB – Laboratory Quantity

Description	Specification
Used to express lab results which are "measured".	The "translation" property is <i>not permitted</i> as non-UCUM units will not be supported. The "value" property is <i>required</i> , contains the actual amount, and is constrained to 11 digits above the decimal and up to 4 digits beneath the decimal (9999999999.9999). The exponent syntax

	(E.g. 2.2345374E7) is not permitted. The “unit” property is <i>required</i> and is constrained to the list of values found in the Terminology Worksheet, LaboratoryObservationUnitsOfMeasurex_LabUnitsOfMeasure value set. The “originalText” property is pre-adopted from HL7 data types Release 2. The originalText property is <i>required</i> and is constrained to ST with a maximum length of 150 characters. If the element is null with a nullFlavor code, then the “originalText” property is <i>mandatory</i> (and no other properties may be specified).
--	--

Example:

“25.6 ug per mg” would be specified as:

```
<value xsi:type="PQ" specializationType="PQ.LAB" value="25.6" unit="ug/mg"/>
```

Example using originalText, where the value is too small to be measured

```
<value xsi:type="PQ" specializationType="PQ.LAB" nullFlavor="TRC">
  <originalText>Trace amounts identified</originalText>
</value>
```

6.12 IVL<PQ.x> – Quantity Range

Description	Specification
This is used to express an allowed range of quantities.	The properties “lowClosed”, “highClosed”, “center” and “width” are <i>not permitted</i>. “low” property identifies the lower bound of the range, is <i>populated</i> and must follow one of the supported “PQ” data type flavors. “high” property identifies the upper bound of the range, is <i>populated</i> and must follow one of the supported “PQ” data type flavors. At least one of the low property or high property must be specified with a non null value.

Example:

Take if temperature goes “above 40 degrees Celsius ” would be specified as:

```
<value xsi:type="IVL_PQ"><low xsi:type="PQ" specializationType="PQ.BASIC" value="40"
unit="Cel"/>
<high xsi:type="PQ" nullFlavor="PINF"/></value>
```

6.13 URG<PQ.x> – Uncertain Quantity

Description	Specification
This is used to express a single quantity whose specific value is not known, but whose upper and lower bounds are known. The URG data type already places implicit constraints on probability. Outside the range, probability is 0, within the range, probability is unknown. PQ.x implies any of the PQ.x data types in the specification (e.g. PQ.LAB, PQ.DISTANCE), not the PQ data type by itself.	The properties “center” and “width” are <i>not permitted</i>. “low” property identifies the lowest of the possible values, is <i>populated</i> and must follow one of the supported “PQ” data type flavors. The low value is not included in the probable range of values. “lowClosed” is similar to “low” except that it indicates that the value is included in the probable range of values. Only one of “low” and “lowClosed” is allowed. “high” property identifies the highest of the possible values, is <i>populated</i> and must follow one of the supported “PQ” data type flavors. The high value is not included in the probable range of values. “highClosed” is similar to high except that it indicates that the value is included in the probable range of values. Only one of “high” and “howClosed” is allowed. Note: In the XML ITS, the “lowClosed” and “highClosed” properties are actually implemented with an “inclusive” attribute on the low and high attributes. Thus lowClosed equates to low/@inclusive=”true” and highClosed equates to high/@inclusive=”true”. Since the inclusive property defaults to true, to specify only low or high, you might include inclusive=”false”.

Examples:

Take “1 to 3 tablets” would be specified as:

```
<value xsi:type="URG_PQ">  
  <low xsi:type="PQ" specializationType="PQ.DRUG" value="1"/>  
  <high xsi:type="PQ" specializationType="PQ.DRUG" value="3"/>  
</value>
```

Between 30 kg, inclusive and 35 kg, exclusive would be specified as:

```
<value xsi:type="URG_PQ" >  
  <low xsi:type="PQ" specializationType="PQ.HEIGHTWEIGHT" inclusive="true" value="30" unit="kg"/>  
  <high xsi:type="PQ" specializationType="PQ. HEIGHTWEIGHT" inclusive="false" value="35"  
unit="kg"/>  
</value>
```

Between 0.01 and 0.05 mmol/L :

```
<value xsi:type="URG_PQ" >  
  <low xsi:type="PQ" specializationType="PQ.LAB" inclusive="false" value="0.01" unit="mmol/L" >  
  </low>  
  <high xsi:type="PQ" specializationType="PQ.LAB" inclusive="false" value="0.05" unit="mmol/L " >  
  </high>  
  <originalText>Between 0.01 and 0.05 mmol/L</originalText>  
</value>
```

6.14 RTO<PQ.DRUG, PQ.TIME> – Drug Quantity per Time Duration

Description	Specification
Used to express a quantity of drug over a particular time-period.	Only numerator and denominator properties may be specified and both are <i>mandatory</i> . “numerator” property indicates the quantity of drug and must comply with the PQ.DRUG specification. “denominator” property indicates the time duration over which that quantity of drug should be given and must comply with the PQ.TIME specification.

Example:

10 tablets per day” would be specified as:

```
<maxDoseQuantity xsi:type="RTO_PQ_PQ"><numerator xsi:type="PQ"  
specializationType="PQ.DRUG" value="10"/><denominator xsi:type="PQ"  
specializationType="PQ.TIME" value="1" unit="d"/>  
</maxDoseQuantity>
```

6.15 RTO<MO.CAD,PQ.BASIC> – Money per Unit Quantity

Description	Specification
Used to express a monetary value over quantity.	Only numerator and denominator properties may be specified and both are mandatory. “numerator” property indicates the monetary value and must comply with the MO.CAD specification. “denominator” property indicates the unit quantity and must comply with the PQ.BASIC specification.

Example:

“\$20 per gram” would be specified as:

```
<unitPriceAmt xsi:type="RTO_MO_PQ"><numerator xsi:type="MO" specializationType="MO.CAD" value="20.00"/><denominator xsi:type="PQ" specializationType="PQ.BASIC" value="1" unit="gram"/></unitPriceAmt>
```

7 Demographics

7.1 PN.BASIC – Basic Person Name

Description	Specification
Used to express person names for general identification and communication purposes.	The “validTime” property is <i>not permitted</i>. The “use” property is <i>mandatory</i> and is constrained to the list of values found in the Terminology Worksheet, x_BasicPersonNameUse value set. No more than seven name parts must be present. The name parts are constrained to the list of values found in the Terminology Worksheet, x_BasicPersonNamePartType value set. Each name part is constrained to a <u>50-75</u> character string (ST). The “qualifier” attribute is <i>required</i> and, if present, is constrained to the list of values found in the Terminology Worksheet, x_BasicPersonNamePartQualifier value set. There is no constraint on the allowed values of either the “prefix” or “suffix” components. -Support for multiple prefix or suffix is not supported at this time. -If and when support for multiple prefix or suffix are allowed, these must be expressed as multiple <prefix></prefix> or <suffix></suffix> tags. The first occurrence of a given name is the “first name”. The requirement for middle name is met through additional repetitions of “given names”. Designation is met through use of the “suffix”. Note that the ‘HC’ – health card name use concept supported in the MR2007 release has been replaced with ‘OR’ – official registry in this release (R02.04.00).

Example:

```
<name use="L" xsi:type="PN" specializationType="PN.BASIC"><prefix>Mr</prefix>  
<given>John</given> <given qualifier="IN">W</given> <family>Smith</family></name>
```

7.2 PN.SIMPLE – Simple Person Name

Description	Specification
Used to express person names without name parts.	The “validTime” property is <i>not permitted</i>. The “use” property is <i>mandatory</i> and is constrained to the list of values found in the Terminology Worksheet, x_BasicPersonNameUse value set. The names are to be expressed as a simple string with parts not distinguished. Systems are expected to be able to support names entered or transmitted in this format, though discrete name parts should be captured whenever feasible. Note that the ‘HC’ – health card name use concept supported in the MR2007 release has been replaced with ‘OR’ – official registry in this release.

Example:

```
<name use="L" xsi:type="PN" specializationType="PN.SIMPLE">Mr. John W. Smith</name>
```

7.3 PN.FULL – Full Person Name

Description	Specification
Used to express person names within a registry where full qualification is needed.	The “validTime” property is <i>not permitted</i>. The “use” property is <i>mandatory</i> and is constrained to the list of values found in the Terminology Worksheet, x_BasicPersonNameUse value set. No more than seven name parts must be present; the minimum number of name parts is 1. The name parts are constrained to the list of values found in the Terminology Worksheet, x_BasicPersonNamePartType value set. Each name part is constrained to a 50-75 character string (ST). The “qualifier” attribute is <i>required</i> and, if present, is constrained to the list of values found in the Terminology Worksheet, x_FullPersonNamePartQualifier value set. There is no constraint on the allowed values of either the “prefix” or “suffix” components. Support for multiple prefix or suffix is not supported at this time. The first occurrence of a given name is the “first name”. The requirement for middle name is met through additional repetitions of “given names”. Designation is met through use of the “suffix”. Note that the ‘HC’ – health card name use concept supported in the MR2007 release has been replaced with ‘OR’ – official registry in this release (R02.04.00).

Example:

```
<name use="L" xsi:type="PN" specializationType="PN.FULL"><prefix>Mr</prefix>
<given>John</given> <given qualifier="IN">W</given> <family>Smith</family></name>
```

7.4 PN.SEARCH – Searchable Name

Description	Specification
This type is used to convey name information for searching purposes. All matches are considered to be “starts with” for each specified property.	All name parts available to PN.FULL are permitted and are noted as optional. However, at least one part type must be specified (e.g. family). The use property is optional Remaining constraints are the same as for PN.FULL.

Example:

```
<name xsi:type="PN" specializationType="PN.SEARCH"><family>Smith</family></name>
```

Example for Find Candidates Query. Note the use attribute is not specified because it is optional.

```
<personName>
  <value xsi:type="PN" specializationType="PN.SEARCH">
    <prefix>Dr</prefix>
    <given>John</given>
    <given qualifier="IN">W</given>
    <family>Smith</family>
  </value>
</personName>
```

7.5 TN – Trivial Name

Description	Specification
As per the HL7 data type	A string used for simple name for things (e.g. drug names) and places (e.g. "Canada Health Infoway")

Example:

<name>Canada Health Infoway</name>

7.6 AD.BASIC – Basic Address

Description	Specification
Used to communicate addresses for simple display, mailing and contact purposes. The data type is not generally suitable for registries	The “useablePeriod” and “isNotOrdered” properties are <i>not permitted</i>. The “use” property is <i>required</i> and is constrained to the list of values found in the Terminology Worksheet, x_BasicPostalAddressUse value set. No more than 3 “use” properties for 1 address may be specified. The supported address parts (part types) can be located in the Terminology Worksheet; x_BasicAddressPartType values set.. The address parts are Required. Other part types are <i>not permitted</i>. All other address information is sent as plain text, separated by delimiter tags. There may be up to 4 additional lines of delimiter-separated information in addition to the specified address parts of city, state, ~~postalCode~~<u>postal Code</u> and country. Both address parts and delimiter-separated text are constrained to a length of 80 characters. It is likely that <addressLine> will be supported in a future release of these specifications to avoid the use of mixed text in an address. Implementers are thus cautioned to avoid the use mixed text. Mixed text is defined as a combination of address parts and delimiter-separated text. For country and state address parts, the code attribute is allowed and is <i>optional</i> (taking advantages of the SC data type), restricted to ISO 3166-1 2-character alpha codes and ISO 3166-2 code suffixes respectively. Use of ISO-3166-1 (alpha-3) is <i>not permitted</i> with this specification (alpha-2 is the only coding system suggested). Using multiple methods (alpha-2, alpha-3 or numeric) would require Implementers to support all methods. The alpha-2 approach was used to be consistent with Canada Post. For example, Canada would be “CA” and British Columbia would be “CA-BC”. These are the same codes used by Canada Post. The code attribute for all other address parts is not permitted.

Example:

```
<addr use="H PST" xsi:type="AD" specializationType="AD.BASIC">
  Apt A5 123 Some Street N.W.<delimiter/><city>Edmonton</city>
  <state code="CA-AB" codeSystem="2.16.840.1.113883.2.20.5.2">Alberta</state>
  <postalCode>A1B2C3</postalCode>
  <delimiter/><country code="CA" codeSystem="2.16.840.1.114884.5.28">Canada</country>
</addr>
```

7.7 AD.FULL – Full Address

Description	Specification
Used to communicate fully validatable addresses. Usually appropriate only for registry-type situations.	The “isNotOrdered” property is <i>not permitted</i>. The “useablePeriod” is <i>optional</i>. If present, it must follow the IVL<TS.DATE> specifications. “use” property is constrained in the same manner as AD.BASIC. The supported address parts (part types) can be located in the Terminology Worksheet; x_FullAddressPartType values set. The “city”, “state”, “postalCode” and “country” properties are <i>mandatoryrequired</i>. In addition, applications may <i>optionally</i> choose to support the remaining properties listed in the value set. Other part types (not listed) are <i>not permitted</i>. Note that PO Box, Rural Route addresses and other non-street-based addresses must NOT use the SAL address part. Remaining constraints are the same as for AD.BASIC.

Example:

STREET ADDRESS EXAMPLE

```
<addr use="H PST" xsi:type="AD" specializationType="AD.FULL">
  <unitID>A5</unitID><unitType>APT</unitType>
  <houseNumber>123</houseNumber><streetNameBase>Some</streetNameBase>
  <streetNameType>Street</streetNameType> <direction>N.W.</direction>
  <delimiter/><city>Edmonton</city><state code="CA-AB"
codeSystem="2.16.840.1.113883.2.20.5.2">Alberta</state>
  <postalCode>A1B2C3</postalCode><delimiter/><country code="CA"
codeSystem="2.16.840.1.114884.5.28">Canada</country>
</addr>
```

RURAL ROUTE EXAMPLE

```
<addr use="H PST" xsi:type="AD" specializationType="AD.FULL">
  RR 3 Box 12000
  <delimiter/><city>Edmonton</city><state code="CA-AB"
codeSystem="2.16.840.1.113883.2.20.5.2">Alberta</state>
  <postalCode>A1B2C3</postalCode><delimiter/><country code="CA"
codeSystem="2.16.840.1.114884.5.28">Canada</country>
</addr>
```

POST OFFICE BOX EXAMPLE

```
<addr use="H PST" xsi:type="AD" specializationType="AD.FULL">
  PO Box 12000 STN MAIN A
  <delimiter/><city>Edmonton</city><state code="CA-AB"
codeSystem="2.16.840.1.113883.2.20.5.2">Alberta</state>
  <postalCode>A1B2C3</postalCode><delimiter/><country code="CA"
codeSystem="2.16.840.1.114884.5.28">Canada</country>
</addr>
```

7.8 AD.SEARCH – Searchable Address

Description	Specification
This type is used to convey high level address information for searching purposes. All matches are considered to be "starts with" for each specified property.	The "isNotOrdered", "use" and "useablePeriod" properties are <i>not permitted</i> . At least one part type must be specified (e.g. city). Mixed content is not permitted Remaining constraints are the same as for AD.BASIC.

Example:

```
<addr xsi:type="AD" specializationType="AD.SEARCH">  
  <city>Edmonton</city><state code="CA-AB"  
  codeSystem="2.16.840.1.113883.2.20.5.2">Alberta</state>  
  <country code="CA" codeSystem="2.16.840.1.114884.5.28">Canada</country></addr>
```

TO USE AS A QUERY REQUEST PARAMETER:

```
<personAddress>  
  <value xsi:type="AD" specializationType="AD.SEARCH">  
    <city>Edmonton</city>  
    <state code="CA-AB" codeSystem="2.16.840.1.113883.2.20.5.2">Alberta</state>  
    <country code="CA" codeSystem="2.16.840.1.114884.5.28">Canada</country>  
  </value>  
</personAddress>
```

7.9 TEL.ALL – Phone, E-mail or URI <Abstract>

Description	Specification
<u>Allows phone and fax numbers, e-mail addresses as well as web addresses and other internet communication schemes to be communicated.</u> <u>This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as TEL.PHONE, TEL.EMAIL or TEL.URI.</u>	<u>The specific data type must be declared using specializationType of TEL.PHONE, TEL.EMAIL or TEL.URI.</u> <u>Please refer to TEL.PHONE, TEL.EMAIL and TEL.URI for further details.</u>

7.97.10 **TEL.PHONEMAIL – Phone or E-mail <Abstract>**

Description	Specification
Allows phone and fax numbers as well as e-mail addresses to be communicated. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as TEL.PHONE or TEL.EMAIL for each repetition.	The specific data type must be declared using specializationType of TEL.PHONE or TEL.EMAIL Please refer to TEL.PHONE and TEL.EMAIL for further details ~~The telecommunication schemes supported are constrained to the list of values found in the Terminology Worksheet, x_PhoneOrEmailURLScheme value set.~~

7.11

TEL.PHONE – Phone Number

Description	Specification
Allows phone and fax numbers to be communicated.	The “useablePeriod” property is <i>not permitted</i>. “use” property is <i>required</i>, may repeat up to three times and is restricted to the list of values found in the Terminology Worksheet, x_BasicTelecommunicationAddressUse value set. Additional guidance for DIR: to be used for immediate direct communications between providers (e.g. pharmacist to physician for potential medication interaction issues). The “value” property is <i>mandatory</i> and must follow the specifications as described in http://www.ietf.org/rfc/rfc3966.txt. The maximum length is 40 characters. Note: 1. No constraints have been applied to the content to restrict it to Canadian country codes AND 2. No constraints have been applied to the format. Best practice is to always include country code and area code when known. The telecommunication schemes supported are restricted to the list of values found in the Terminology Worksheet, x_PhoneURLScheme value set.

Example:

To specify (780) 123-4567 as an emergency contact (EC) and a mobile contact (MC) number:

```
<telecom use="EC MC" xsi:type="TEL" specializationType="TEL.PHONE"
value="tel:+17801234567"/>
```

Example:

To specify (416) 923-7867, extension (posd) 200 as a workplace (WP) number:

```
<telecom use="WP" xsi:type="TEL" specializationType="TEL.PHONE"
value="tel:+14169237867;ext=200"/>
```

7.107.11 TEL.EMAIL – Email Address

Description	Specification
Allows e-mail addresses to be communicated.	The “useablePeriod” property is <i>not permitted</i> . “use” property is <i>required</i> , may repeat up to three times and is restricted to the list of values found in the Terminology Worksheet, x_BasicEmailAddressUse value set The “value” property is <i>mandatory</i> and has a maximum length of 50128 characters. The telecommunication scheme supported is “mailto” (Mail to)

Example:

```
<telecom use="WP" xsi:type="TEL" specializationType="TEL.EMAIL"
value="mailto:someone@somewhere.com"/>
```

To mask only an email address (datatype flavour indicates that the mask is for an email address):

```
<telecom use="H" nullFlavor="MSK" specializationType="TEL.EMAIL"/>
```

7.117.12 TEL.URI – Uniform Resource Identifier

Description	Specification
Used to communicate web addresses and other internet communication schemes.	The “useablePeriod” and “use” properties are <i>not permitted</i> . The “value” property is <i>mandatory</i> , has a maximum limit of 255 characters and must conform to the format associated with the identified telecommunication scheme. The telecommunication schemes supported are restricted to the list of values found in the Terminology Worksheet, x_URIURLScheme value set.

Example:

```
<telecom xsi:type="TEL" specializationType="TEL.URI"
value="https://www.someserver.ca/somelocation"/>
```

8 Miscellaneous

8.1 ANY – Unrestricted <Abstract>

Description	Specification
Used in a specification where the specific types to be communicated may vary. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the constrained version of ANY (e.g. ST) for each repetition.	Each value sent over the wire must correspond to one of the non-abstract data type flavor specifications defined in this document. The specific data type must be declared using xsi:type. The use of this type is discouraged in specifications unless additional constraints are included indicating what specific data type flavors may be substituted.

8.2 ANY.LAB – <Abstract>

Description	Specification
Used in a specification where the specific types to be communicated may vary. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the allowed constrained version.	Each value sent over the wire must correspond to one of the following non-abstract data type flavor specifications defined in this document: CD.LAB, ST, PQ.LAB, IVL<PQ.x>, INT.NONNEG, INT.POS, TS.FULLDATETIME, URG<PQ,x> The specific data type must be declared using xsi:type for ST and specializationType for all others . The use of this type is restricted to messages in the LAB domain

8.3 ANY.CA.IZ – <Abstract>

Description	Specification
Used in a specification where the specific types to be communicated may vary. This data type may be referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the allowed constrained version.	Each value sent over the wire must correspond to one of the following non-abstract data type flavor specifications defined in this document: ST, PN.BASIC, INT.POS, BL The specific data type must be declared using xsi:type for ST and BL and specializationType for all others .

8.4 ANY.PATH – <Abstract>

Description	Specification
Used in a specification where the specific types to be communicated may vary to allow large text blocks to be messaged. This data type may be	Each value sent over the wire must correspond to one of the following non-abstract data type flavor specifications defined in this document: ST, PQ, ED.DOCORREF or CD.LAB The specific data type must be declared using xsi:type for ST and PQ and specializationType for all others .

referenced in Canadian specifications. If this data type is used in a message instance, it must be declared as one of the allowed constrained version.	
--	--

8.5 BL - Boolean

Description	Specification
Allows conveying true-false information.	The "value" property is <i>mandatory</i> and case-sensitive. It must be either "true" or "false". Note that unless the attribute declaring this type is marked as <i>mandatory</i> , the value is allowed to be null. This must be taken into account when performing boolean logic. Consult the HL7 international abstract data types specification for further details.

Example:

```
<negationInd value="true"/>
```

Example for Find Candidates Query where the multiple birth indicator is false

```
<multipleBirthIndicator><value value='false' /></multipleBirthIndicator>
```

8.6 LIST<x> - List

Description	Specification
Used when multiple repetitions are allowed and order matters.	Each repetition must follow the specification defined for "x". Applications are expected to retain information about the sequence of the repetitions. The semantics of the sequence can be found in the description of the attribute or property. Duplicate repetitions are technically allowed, but should occur only if they make sense in the context of use.

8.7 SET<x> - Set

Description	Specification
Used when multiple repetitions are allowed, order is irrelevant and duplicates are prohibited.	Each repetition must follow the specification defined for "x". Duplicate repetitions are not allowed.