



5 **IHE Quality, Research, and Public Health
Technical Framework Supplement**

10 **Structured Data Capture
(SDC)**

15 **Rev. 3.1 – Trial Implementation**

20 Date: May 14, 2021
Author: QRPH Technical Committee
Email: qrph@ihe.net

25 **Please verify you have the most recent version of this document. See [here](#) for Trial
Implementation and Final Text versions and [here](#) for Public Comment versions.**

Foreword

30 This is a supplement to the IHE Quality, Research and Public Health (QRPH) Technical Framework. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on May 14, 2021 for trial implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the Quality, Research and
35 Public Health Technical Framework. Comments are invited and can be submitted at http://www.ihe.net/QRPH_Public_Comments.

This supplement describes changes to the existing technical framework documents.

“Boxed” instructions like the sample below indicate to the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume.

40 *Amend Section X.X by the following:*

Where the amendment adds text, make the added text **bold underline**. Where the amendment removes text, make the removed text **~~bold strikethrough~~**. When entire new sections are added, introduce with editor’s instructions to “add new text” or similar, which for readability are not bolded or underlined.

45

General information about IHE can be found at <http://ihe.net>.

Information about the IHE QRPH domain can be found at http://ihe.net/IHE_Domains.

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at http://ihe.net/IHE_Process and <http://ihe.net/Profiles>.

50 The current version of the IHE QRPH Technical Framework can be found at http://ihe.net/Technical_Frameworks.

55 **CONTENTS**

	Introduction to this Supplement.....	6
	Open Issues and Questions	7
	Closed Issues:.....	7
60	IHE Technical Frameworks General Introduction.....	8
9	Copyright Licenses	8
	9.1 Copyright of Base Standards.....	8
	9.1.1 DICOM (Digital Imaging and Communications in Medicine)	8
	9.1.2 HL7 (Health Level Seven)	8
65	9.1.3 LOINC (Logical Observation Identifiers Names and Codes).....	8
	9.1.4 SNOMED CT (Systematized Nomenclature of Medicine -- Clinical Terms)	9
10	Trademark	9
	IHE Technical Frameworks General Introduction Appendices.....	10
	Appendix A – Actor Summary Definitions	10
70	Appendix B – Transaction Summary Definitions.....	10
	Appendix D – Glossary.....	10
	Volume 1 – Profiles	14
	Copyright Licenses.....	14
	Domain-specific additions	14
75	X Structured Data Capture (SDC) Profile	15
	X.1 SDC Actors, Transactions, and Content Modules.....	15
	X.1.1 Actor Descriptions and Actor Profile Requirements.....	17
	X.1.1.1 Form Filler	17
	X.1.1.2 Form Manager	17
80	X.1.1.3 Form Processor	17
	X.1.1.5 Form Receiver	17
	X.2 SDC Actor Options	17
	X.2.1 Form Filler: SDC Pre-Pop Option.....	18
	X.2.2 Form Filler: SDC XML Package Option	18
85	X.2.3 Form Filler: SDC HTML Package Option	18
	X.2.4 Form Filler: SDC URI Form Option	18
	X.2.5 Form Filler: SDC Auto-Pop Option	19
	X.2.6 Form Manager: SDC Pre-Pop Option	19
	X.2.7 Form Processor: SDC Pre-pop Option	19
90	X.3 SDC Required Actor Groupings	20
	X.4 SDC Overview	20
	X.4.1 Concepts	20
	X.4.1.1 Pre-Population	20
	X.4.1.2 Auto-Population.....	21
95	X.4.2 Use Cases	21
	X.4.2.1 Use Case #1: Retrieve form using a URL	21

	X.4.2.1.1 Retrieve form using a URL - Use Case Description.....	21
	X.4.2.1.2 Retrieve form using URL - Process Flow.....	21
	X.4.2.1.3 Pre-conditions	22
100	X.4.2.1.4 Main Flow.....	22
	X.4.2.1.5 Post-conditions.....	23
	X.4.2.2 Use Case #2: Capture and Submit Pre-Authorization	23
	X.4.2.2.1 Capture and Submit Pre-Authorization - Use Case Description.....	23
	X.4.2.2.2 Capture and Submit Pre-Authorization - Process Flow	24
105	X.4.2.2.3 Pre-conditions	24
	X.4.2.2.4 Main Flow.....	24
	X.4.2.2.5 Post-conditions.....	25
	X.4.2.3 Use Case #3: Public Health Reporting	25
	X.4.2.3.1 Public Health Reporting – Use Case Description.....	25
110	X.4.2.3.2 Public Health Reporting – Process Flow	26
	X.4.2.3.3 Pre-conditions:	26
	X.4.2.3.4 Main Flow.....	26
	X.4.2.3.5 Post-conditions.....	27
	X.5 SDC Security Considerations.....	27
115	X.5.1 Use of IHE ATNA for Recording Security Audit Events	27
	X.5.2 Digital Signature.....	27
	X.5.3 Consistent Time (CT).....	28
	X.5.4 Cross Enterprise User Assertion (XUA)	28
	Volume 1 Appendices.....	29
120	Volume 2 – Transactions.....	30
	Volume 2 Appendices.....	31
	Volume 2 Namespace Additions	31
	Volume 3 – Content Modules.....	32
	5 Namespaces and Vocabularies.....	32
125	6 Content Modules.....	32
	6.3.1 CDA Document Content Modules	32
	Q SDC Content Modules	32
	Q.1 SDC Pre-Pop Content Module	32
	Q.2 SDC XML Package Content Module.....	33
130	Q.2.1 SDC XML Package – Request	33
	Q.2.2 SDC XML Package – Response.....	34
	Q.3 SDC HTML Package Content Module	36
	Q.3.1 SDC HTML Package – Request.....	36
	Q.3.2 SDC HTML Package – Response	37
135	Q.4 SDC URI Form Content Module	38
	Q.4.1 SDC URI Form – Request.....	39
	Q.4.2 SDC URI Form – Response	39
	Q.5 SDC Submission Data Content Module.....	40

	Q.5.1 SDC Submission – Request.....	40
140	Q.5.2 SDC Submission - Response.....	41
	Q.6 SDC Form Definition Model	42
	Q.6.1 Scope and Approach.....	42
	Q.6.2 SDC Package	43
	Q.6.2.1 Admin	44
145	Q.6.2.2 SubmissionRule	45
	Q.6.2.3 ComplianceRule	46
	Q.6.2.4 HTMLPackage.....	46
	Q.6.2.5 XMLPackage	47
	Q.6.2.5.1 DemogFormDesign.....	48
150	Q.6.2.5.2 MapTemplate	48
	Q.6.2.5.2.1 DefaultCodeSystem	50
	Q.6.2.6 FormDesign	50
	Q.6.2.6.1 Form Structure	52
	Q.6.2.6.2 Question	55
155	Q.6.2.6.3 ListField	56
	Q.6.2.6.4 ResponseField.....	58
	Q.6.2.6.5 DisplayedItem.....	60
	Q.6.3 Supplemental Material	62
	Volume 3 Appendices.....	63
160	Appendix A – List of SDC Form Elements and Attributes Adapted from ISO/IEC 19763-13 and ISO/IEC 11179-3, © copyright ISO/IEC 2014	63
	A.1 Base Elements from ISO/IEC Standards.....	63
	A.1.1 Organization	63
	A.1.2 Person	63
165	A.1.3 Contact Information	64
	A.1.4 FileType.....	65
	A.1.5 BlobContent.....	66
	A.1.6 Datatype.....	66
	Volume 3 Namespace Additions	68
170		

Introduction to this Supplement

175 The Structured Data Capture (SDC) Content Profile provides specifications to enable an Electronic Health Record (EHR) system or other application to retrieve a data capture form and submit data from the completed form. The SDC Profile utilizes the IHE Retrieve Form for Data Capture (RFD) Profile and an ISO/IEC 19763-13 Meta-model for Framework Interoperability (MFI) form registration.

180 This supplement is based on the work of the Office of the National Coordinator (ONC) [Standards & Interoperability \(S&I\) Framework SDC Initiative](#). The SDC initiative has developed use cases, identified national standards for the structure of Common Data Elements and form model definition, developed guidance to assist in implementation, and conducted pilots for evaluation of SDC.

This supplement also references the following documents. The reader should review these documents as needed:

- 185
 - IT Infrastructure Technical Framework Volume 1
 - Retrieve Form for Data Capture (RFD) Integration Profile
 - Audit Trail and Node Authentication (ATNA) Integration Profile
 - Consistent Time (CT) Integration Profile
 - Cross-Enterprise User Assertion (XUA) Integration Profile
- 190
 - IT Infrastructure Technical Framework Volume 2b
 - HL7 CDA R2 and other standards documents referenced in Volume 1 and Volume 2
 - IETF HTTPS and TLS v1.0 standard
 - W3C SOAP
 - OASIS SAML
- 195
 - ISO/IEC 11179-3:2013 Metadata Registries - Part 3 Registry metamodel and basic attributes
 - ISO/IEC CD 19763-13 Metamodel for Forms Registration
 - Optionally, the QRPH Clinical Research Document (CRD) Trial Implementation Supplement for the definition of Audit Log message content and the [QRPH-36]
- 200
 - transaction

Open Issues and Questions

SDC Profile – Open Issues and Questions

Item #	Section	Question
1.	5.0	IHE Template Issue: Invented section numbering scheme, since none was defined in the template that allowed inclusion of other text.

Closed Issues:

- 205
1. Does SDC include a Transaction or Content Module? SDC doesn't provide you with new transactions, but instead allows you to include content sections that reference relevant sections. The constraints on the transactions (structured, unstructured, or URL) will be explained in Volume 3.
- 210
2. Why does SDC Profile use XAdES instead of IHE DSG? SDC use case requires digital signature to be inclusive (part of the transaction) rather than being included as a separate document, which is not supported by IHE DSG Profile and hence the use of XAdES.
- 215
3. Will <sdcc:form_info> element within SDC HTML Package contain a reference to SDC XML Package? The SDC HTML Package contains the mapping package, admin package, submission rules info, etc., which provides Form Filler with enough information without having to refer to the SDC Form Definition. The Form Manager/Form Processor does have an option to send this information, if they choose to do so.
- 220
4. How do we add a note indicating that the reason for having SDC Submission Data content module optional is the lack of SDC compliant forms? Based on the public comments received, this was changed to a Required content module and hence no need to add a separate note.
5. Consider future Change Proposal (CP) to constrain the <sdcc:supplemental_data> schema to conform to the same scheme used for <sdcc:submitted_data>

IHE Technical Frameworks General Introduction

- 225 The [IHE Technical Framework General Introduction](#) is shared by all of the IHE domain technical frameworks. Each technical framework volume contains links to this document where appropriate.

9 Copyright Licenses

- 230 IHE International hereby grants to each Member Organization, and to any other user of these documents, an irrevocable, worldwide, perpetual, royalty-free, nontransferable, nonexclusive, non-sublicensable license under its copyrights in any IHE profiles and Technical Framework documents, as well as any additional copyrighted materials that will be owned by IHE International and will be made available for use by Member Organizations, to reproduce and distribute (in any and all print, electronic or other means of reproduction, storage or
235 transmission) such IHE Technical Documents.

The licenses covered by this Copyright License are only to those copyrights owned or controlled by IHE International itself. If parts of the Technical Framework are included in products that also include materials owned or controlled by other parties, licenses to use those products are beyond the scope of this IHE document and would have to be obtained from that other party.

240 9.1 Copyright of Base Standards

- IHE technical documents refer to and make use of a number of standards developed and published by several standards development organizations. All rights for their respective base standards are reserved by these organizations. This agreement does not supersede any copyright provisions applicable to such base standards. Copyright license information for frequently
245 referenced base standards is provided below.

9.1.1 DICOM (Digital Imaging and Communications in Medicine)

DICOM[®] is the registered trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information.

9.1.2 HL7 (Health Level Seven)

- 250 HL7[®], Health Level Seven[®], CDA[®], FHIR[®], and CCD[®] are registered trademarks of Health Level Seven International.

Health Level Seven, Inc. has granted permission to IHE to reproduce tables from the HL7 standard. The HL7 tables in this document are copyrighted by Health Level Seven, Inc. All rights reserved. Material drawn from these documents is credited where used.

255 9.1.3 LOINC (Logical Observation Identifiers Names and Codes)

LOINC[®] is registered United States trademarks of Regenstrief Institute, Inc.

9.1.4 SNOMED CT (Systematized Nomenclature of Medicine -- Clinical Terms)

260 Some IHE Profiles incorporate SNOMED[®] CT, which is used by permission of the International Health Terminology Standards Development Organisation. SNOMED CT[®] was originally created by the College of American Pathologists. SNOMED CT is a registered trademark of the International Health Terminology Standards Development Organisation, all rights reserved.

10 Trademark

265 IHE[®] and the IHE logo are trademarks of the Healthcare Information Management Systems Society in the United States and trademarks of IHE Europe in the European Community. They may only be used with the written consent of the IHE International Board Operations Committee, which may be given to a Member Organization in broad terms for any use that is consistent with the IHE mission and operating principles.

IHE Technical Frameworks General Introduction Appendices

- 270 The [IHE Technical Framework General Introduction Appendices](#) are components shared by all of the IHE domain technical frameworks. Each technical framework volume contains links to these documents where appropriate.

- 275 *Update the following appendices to the General Introduction as indicated below. Note that these are not appendices to this domain's Technical Framework (TF-1, TF-2, TF-3 or TF-4) but rather, they are appendices the IHE Technical Frameworks General Introduction located here.*

Appendix A – Actor Summary Definitions

Add the following new or modified actors to the IHE Technical Frameworks General Introduction Appendix A:

280

New (or modified) Actor Name	Definition
No new actors	

Appendix B – Transaction Summary Definitions

Add the following new or modified transactions to the IHE Technical Frameworks General Introduction Appendix B:

285

New (or modified) Transaction Name and Number	Definition
No new transactions	

Appendix D – Glossary

Add the following new or updated glossary terms to the IHE Technical Frameworks General Introduction Appendix D.

290

Glossary Term	Definition
Auto-Population	When an EHR system automatically fills in form fields with data that are already available within the system's database.

Glossary Term	Definition
CCD	Continuity of Care Document (CCD) is document specification standard specified by HL7/ASTM and commonly used for electronic document exchange. CCD is based on HL7's Clinical Document Architecture (CDA).
CDA	The HL7 Version 3 Clinical Document Architecture (CDA) is a document markup standard that specifies the structure and semantics of "clinical documents" for the purpose of exchange between healthcare providers and patients. It defines a clinical document as having the following six characteristics: 1) Persistence, 2) Stewardship, 3) Potential for authentication, 4) Context, 5) Wholeness and 6) Human readability.
Common Data Elements (CDEs)	Standardized data element descriptions for collection and exchange of data of common interest to a particular community, and thus the community has agreed to share their definition, management and use. CDEs share a common set of attributes which facilitates their reuse in different settings, and are intended to aide in interoperability and data reuse.
Completed Form	A form where all the fields contain data – through a combination of pre-population, auto-population, and manual edits, and is ready for submission.
Data Element (DE)	A data element is a unit of data for which the definition, identification, representation, and permissible values are specified by a set of attributes and considered in context to be indivisible.
Enhanced Form Repository	A form repository with capability to pre-populate form with the data received from the Form Filler.
External Data Repository	A database, outside of the EHR system, where completed forms data can be stored.
Form	A form with data entry fields that will be filled out by an end user or provider.
Form Repository	An authoritative source for forms.
HL7	Health Level Seven is a not-for-profit, American National Standards Institute (ANSI)-accredited health care focused International and membership-driven Standard Development Organization (SDO) based in the United States with international affiliates.
MFI	Metamodel Framework for Interoperability (MFI) -- an ISO/IEC 19763 standard.

Glossary Term	Definition
MFI-13	Metamodel Framework for Interoperability (MFI) – ISO/IEC 19763-13 Metamodel for Forms Registration
ONC	The U.S. Department of Health and Human Services (HHS) Office of the National Coordinator for Health Information Technology.
Partially Completed Form	A pre-populated and/or auto-populated form served by the EHR to the provider that contains data for most fields.
PHI	Protected Health Information, as defined in the United States Code of Federal Regulations (Part 45 CFR 160.103) and, as referenced in Section 13400 of Subtitle D ('Privacy') of the HITECH Act.
Pre-Population	When an Enhanced Form Repository fills in form fields using data sent by the Form Filler along with the retrieve request. This activity is distinguished from Auto-population in that Pre-population is performed by the Form Manager using an Enhanced Form Repository, whereas Auto-population is always performed by Form Filler.
RFD	Information Technology Infrastructure (ITI) Retrieve Form for Data Capture (RFD) Profile that provides a means for the retrieval and submission of completed forms data between clinical or research settings and electronic data capture systems or other data collection agencies.
SAML	Security Assertion Markup Language is an Extensible Markup Language standard that allows a user to log on once for affiliated but separate Web sites.
SDC Form Definition	A standardized set of attributes describing the semantics and syntax of a form design so that it may be rendered consistently in any suitable information system and can be validated using SDC Schema. Based on ISO/IEC 19763-13 with SDC extensions. This is not a fillable form.
SDC Schema	A W3C schema for an ISO/IEC 19763-13 compliant form, with SDC extensions.
SDC XML Package	A collection of XML data, meeting the SDC Schema, that includes the particular SDC Form Definition represented as a set of standardized XML elements, along with mapping information, administrative information including submission and compliance instructions.
SDC HTML Package	A collection of files that contains an HTML form instance derived from an SDC Form Definition, along with (optional) mapping information, (optional) administrative information, and (optional) supplemental data. The package is represented in SDC Content modules and contains a base64-encoded HTML form instance.

Glossary Term	Definition
SOAP	Simple Object Access Protocol: An XML-based messaging protocol.
S&I	Standards and Interoperability Framework is an open forum sponsored by ONC's Office of Standards & Interoperability (OSI) to advance harmonization and implementation of specifications that support national healthcare priorities. SDC is an S&I Framework initiative.

Volume 1 – Profiles

Copyright Licenses

Elements and Attributes Adapted from ISO/IEC 19763-13 and ISO/IEC 11179-3, © copyright ISO/IEC 2014.

295 Domain-specific additions

Section not applicable.

X Structured Data Capture (SDC) Profile

With EHR adoption expanding across the globe, the volume and detail of information captured by healthcare organizations and providers is growing exponentially. The SDC Profile provides specifications to enable an EHR system, or other application, to retrieve a data capture form and submit data from a completed form. The SDC Profile utilizes transactions from the IHE ITI Retrieve Form for Data Capture (RFD) Profile and an ISO/IEC 19763-13 based form definition, for requesting and receiving forms, and submitting data captured in forms in a standardized and structured format.

This supplement is based on the work of the ONC’s [S&I Framework SDC Initiative](#). The SDC initiative has developed use cases, identified national standards for the structure of data element (DE) and form definitions, developed guidance to assist in implementation, and conducted pilots for evaluation of SDC.

X.1 SDC Actors, Transactions, and Content Modules

This section defines the actors, transactions, and/or content modules in this profile. Figure X.1-1 shows the actors directly involved in the SDC Profile and the relevant transactions between them.

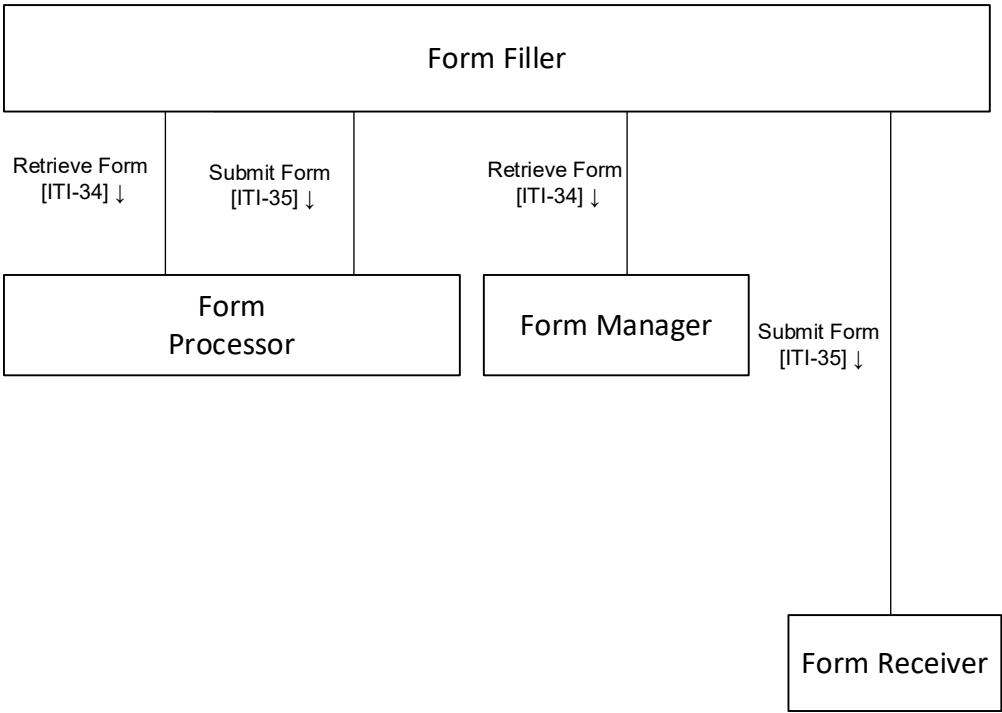


Figure X.1-1: SDC Actor Diagram

Table X.1-1 lists the transactions for each actor directly involved in the SDC Profile. To claim compliance with this profile, an actor shall support all required transactions (labeled “R”) and may support the optional transactions (labeled “O”).

320

Table X.1-1: SDC Profile - Actors and Transactions

Actors	Transactions	Optionality	Reference
Form Filler	Retrieve Form [ITI-34]	R	ITI TF-2b: 3.34
	Submit Form [ITI-35]	R	ITI TF-2b: 3.35
	Archive Form [ITI-36]	O	ITI TF-2b: 3.36
	Archive Source Documents [QRPH-36]	O	QRPH TF-2: 3.36
Form Manager	Retrieve Form [ITI-34]	R	ITI TF-2b: 3.34
Form Receiver	Submit Form [ITI-35]	R	ITI TF-2b: 3.35
Form Processor	Retrieve Form [ITI-34]	R	ITI TF-2b: 3.34
	Submit Form [ITI-35]	R	ITI TF-2b: 3.35
	]		

Table X.1-2: SDC Profile – Actors and Content Modules

Actors	Content Module	Optionality	Section in Vol. 3
Form Filler	SDC Pre-Pop	O	Q.1
	SDC XML Package ^{Note 1}	O	Q.2.1
	SDC HTML Package ^{Note 1}	O	Q.3.1
	SDC URI Form ^{Note 1}	O	Q.4.1
	SDC Submission Data	R	Q.5
Form Manager	SDC Pre-Pop	O	Q.1
	SDC XML Package	R	Q.2.2
	SDC HTML Package	R	Q.3.2
	SDC URI Form	R	Q.4.2
Form Processor	SDC Pre-Pop	O	Q.1
	SDC XML Package	R	Q.2.2
	SDC HTML Package	R	Q.3.2
	SDC URI Form	R	Q.4.2
	SDC Submission Data	R	Q.5
Form Receiver	SDC Submission Data	R	Q.5

Note 1: Form Filler SHALL support at least one of these options: SDC XML Package, SDC HTML Package, or SDC URI Form.

- 325 This SDC Profile builds upon the RFD Profile and HTML in describing both the content and presentation of (data entry) forms. It defines a form definition model and the following two approaches to form representation that are technology and platform-neutral:
1. XML-based form definitions (defines form design based on SDC Schema in Volume 3)
 2. HTML-based form instances (display and presentation of data)

330 **X.1.1 Actor Descriptions and Actor Profile Requirements**

Most requirements are documented in Transactions (Volume 2) and Content Modules (Volume 3). This section documents any additional requirements on profile's actors.

X.1.1.1 Form Filler

The Form Filler is defined in the RFD Profile in ITI TF-1:17.

- 335 See the security considerations in Section X.5.

X.1.1.2 Form Manager

The Form Manager is defined in the RFD Profile in ITI TF-1:17.

See the security considerations in Section X.5.

X.1.1.3 Form Processor

- 340 The Form Processor is defined in the RFD Profile in ITI TF-1:17.

See the security considerations in Section X.5.

X.1.1.5 Form Receiver

The Form Receiver is defined in the RFD Profile in ITI TF-1.

See the security considerations in Section X.5.

345 **X.2 SDC Actor Options**

Options that may be selected for each actor in this profile, if any, are listed in Table X.2-1. Dependencies between options when applicable are specified in notes.

Table X.2-1: SDC - Actors and Options

Actor	Option Name	Reference
Form Filler	SDC Pre-Pop	QRPH TF-1: X.2.1
	SDC XML Package ^{Note 1}	QRPH TF-1: X.2.2
	SDC HTML Package ^{Note 1}	QRPH TF-1: X.2.3
	SDC URI Form ^{Note 1}	QRPH TF-1: X.2.4

Actor	Option Name	Reference
	SDC Auto-Pop	QRPH TF-1: X.2.5
	Archive Form	ITI TF-2b: 3.36
	Archive Source Documents	QRPH TF-2: 3.36
Form Manager	SDC Pre-Pop	QRPH TF-1: X.2.6
Form Processor	SDC Pre-Pop	QRPH TF-1: X.2.7
Form Receiver	None	

Note 1: Form Filler SHALL support at least one of these options: SDC XML Package, SDC HTML Package, or SDC URI Form.

X.2.1 Form Filler: SDC Pre-Pop Option

This option defines the requirements placed on submission of pre-population data. The Form Filler's support for the SDC Pre-Pop Option determines how pre-population data is generated by Form Fillers when requesting forms using Retrieve Form [ITI-34] transaction:

- In order to claim conformance to this option, the Form Filler SHALL implement SDC Pre-Pop content module (see QRPH TF-3: Q.1).

X.2.2 Form Filler: SDC XML Package Option

This option defines the requirement placed on requesting and retrieving an SDC XML Package. The Form Filler's support for this option determines how a request for an SDC XML Package is generated when requesting it using Retrieve Form [ITI-34] transaction:

- In order to claim conformance to this option, the Form Filler SHALL implement SDC XML Package content module (see QRPH TF-3: Q.2).

X.2.3 Form Filler: SDC HTML Package Option

This option defines the requirement placed on requesting and retrieving an SDC HTML Package. The Form Filler's support for this option determines how an SDC HTML Package is requested and rendered when the Form Fillers retrieve it using Retrieve Form [ITI-34] transaction:

- In order to claim conformance to this option, the Form Filler SHALL implement SDC HTML Package content module (see QRPH TF-3: Q.3).

X.2.4 Form Filler: SDC URI Form Option

This option defines the requirement placed on requesting and retrieving form URI. The Form Filler's support for this option determines how a response providing URI to an SDC form is handled when the Form Filler retrieves the information using Retrieve Form [ITI-34] transaction:

- 375
- In order to claim conformance to this option, the Form Filler SHALL implement SDC URI Form content module (see QRPH TF-3: Q.4).

X.2.5 Form Filler: SDC Auto-Pop Option

This option defines the requirements placed on supporting auto-population before rendering the requested form.

- 380
- In order to claim conformance to this option, the Form Filler SHALL support SDC XML Package Option and/or SDC HTML Package Option.
 - In order to claim conformance to this option, the Form Filler SHALL automatically supply some additional form data from the Form Filler data store before rendering the form for human data entry.

The details of how auto-population is performed is out of scope for this profile.

X.2.6 Form Manager: SDC Pre-Pop Option

This option defines the requirements placed on being able to receive pre-population data. The Form Manager's support for the SDC Pre-Pop Option determines Form Manager's capability to receive pre-population data sent by Form Fillers when requesting form using Retrieve Form [ITI-34] transaction:

- 390
- In order to claim conformance to this option, the Form Manager SHALL be able to receive pre-population data as per SDC Pre-Pop content module (see QRPH TF-3: Q.1).

Conformance to this option only means that Form Manager will have the ability to receive a request from Form Filler with pre-population data. The details of how Form Manager will use this data to pre-populate the form is out of scope for this profile.

X.2.7 Form Processor: SDC Pre-pop Option

This option defines the requirements placed on being able to receive pre-population data. The Form Processor's support for the SDC Pre-Pop Option determines Form Processor's capability to receive pre-population data sent by Form Fillers when requesting form using Retrieve Form [ITI-34] transaction:

- 400
- In order to claim conformance to this option, the Form Processor SHALL be able to receive pre-population data as per SDC Pre-Pop content module (see QRPH TF-3: Q.1).

Conformance to this option only means that Form Processor will have the ability to receive a request from Form Filler with pre-population data. The details of how Form Processor will use this data to pre-populate the form is out of scope for this profile.

X.3 SDC Required Actor Groupings

Actor(s) which are required to be grouped with another actor(s) are listed in this section. The grouped actor MAY be from this profile or a different domain/profile. These mandatory required groupings, plus further descriptions if necessary, are given in the table below.

An actor from this profile (Column 1) shall implement all of the required transactions and/or content modules in this profile *in addition to* all of the transactions required for the grouped actor (Column 2).

Section X.5 describes some optional groupings that may be of interest for security considerations.

Table X.3-1: SDC - Required Actor Groupings

SDC Actor	Actor to be grouped with	Reference	Content Bindings Reference
Form Filler	CT Time Client	ITI TF-1: 7.1	N.A.
Form Manager	CT Time Client	ITI TF-1: 7.1	N.A.
Form Processor	CT Time Client	ITI TF-1: 7.1	N.A.
Form Receiver	CT Time Client	ITI TF-1: 7.1	N.A.

X.4 SDC Overview

X.4.1 Concepts

SDC provides for two ways to insert already captured EHR data into forms: pre-population and auto-population. Each process is described in turn below, and three use cases are provided which illustrate the timing and locus of activities for these two central concepts. Use Case 1 demonstrates the use of SDC to display a form, which has no EHR data applied. Use Case 2 illustrates the use of auto-population only in a pre-authorization setting. Use Case 3 shows the feasibility for both pre-population and auto-population to be used in one form completion process.

In addition to illustrating the various permutations of pre-population and auto-population, the use cases also illustrate use of SDC in three settings: research, public health reporting, and quality reporting.

X.4.1.1 Pre-Population

The first of two ways to apply EHR data to a form is called pre-population. In this approach, the EHR exports a standard document, typically a templated CDA, to an external actor which uses the data from the document to populate fields in the form. In this profile, this capability will be

supported by Form Manager or Form Processor. The pre-population capability is also described in the Retrieve Form [ITI-34] transaction.

X.4.1.2 Auto-Population

- 435 SDC enables a capability for a Form Filler, such as an EHR system, to apply data directly to the form. In this approach, the data element definitions within the form would be interpreted by the EHR system, and corresponding instance data would be retrieved from the EHR database and applied to the form. The mechanism to accomplish this is out of scope for the profile.

X.4.2 Use Cases

440 X.4.2.1 Use Case #1: Retrieve form using a URL

In this use case, the EHR retrieves the form using a URL without providing patient data. There is no pre-population or auto-population. The setting is a research study conducted at a healthcare site.

X.4.2.1.1 Retrieve form using a URL - Use Case Description

- 445 A research study coordinator is in charge of sponsored research at the research site. The study has several patients enrolled as subjects that require them to make study visits at the research site. When subject Jones arrives for a research related visit, the study coordinator, acting within the site's EHR system, requests the URL for the appropriate form for the research. From a research electronic data capture system, the provider uses the URL to navigate to the form (based
450 on SDC Form Definition) and enter the required data.

X.4.2.1.2 Retrieve form using URL - Process Flow

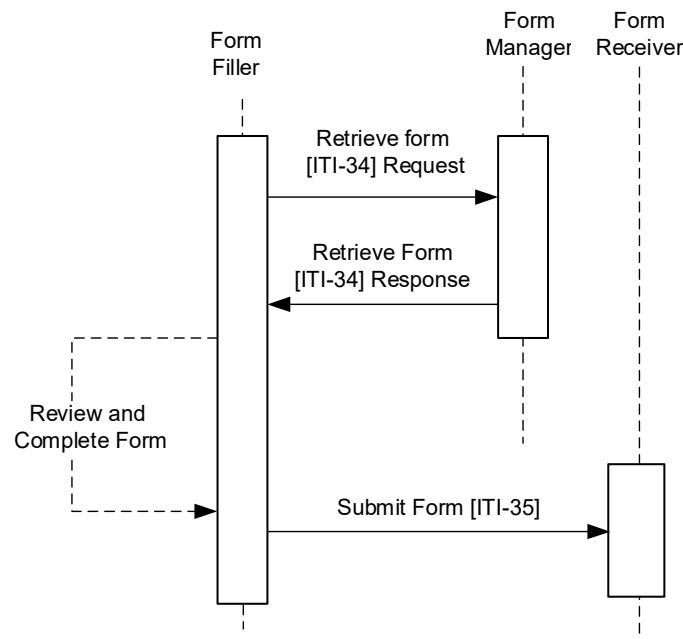


Figure X.4.2.1.2-1: Retrieve form using an URL - Process Flow diagram

455 **X.4.2.1.3 Pre-conditions**

- The EHR system performs the role of a Form Filler;
- A patient enrolled as a subject in a research study arrives for a study visit at which a form completion is required;
- The EHR system knows the source of the URL to obtain access to the appropriate form.

460 **X.4.2.1.4 Main Flow**

- A study coordinator retrieves a research form, which is hosted by a research system external to the healthcare site, through the site's EHR. The Form Manager returns a URL, and the EHR system directs to that URL to display the form (based on SDC Form Definition);
- The study coordinator completes the form through its display within the EHR user interface;
- The study coordinator submits the completed form to the external repository, and to the archive.

X.4.2.1.5 Post-conditions

- 470 • The research system stores the data captured on the form.

X.4.2.2 Use Case #2: Capture and Submit Pre-Authorization

This use case illustrates auto-population with an SDC form, with no pre-population, using a pre-authorization setting.

X.4.2.2.1 Capture and Submit Pre-Authorization - Use Case Description

- 475 Dr. R. E. Hab has a patient who needs an electric wheelchair. The patient needs to obtain pre-authorization from his insurer to confirm that the wheelchair will be covered. The insurer makes forms available electronically including the pre-authorization form. The insurer requires that the form be submitted by the patient's provider.
- 480 When Dr. Hab sees the patient in the office, he opens up the patient's EHR. After recording the visit data in the EHR, Dr. Hab requests the EHR to display the pre-authorization form. Dr. Hab uses an EHR that has the ability to process and render SDC forms. Dr. Hab requests the form from within the EHR. The EHR has been configured to electronically retrieve the pre-authorization forms from the insurer without submitting PHI data. The insurer electronically returns the form in an SDC compliant format.
- 485 The EHR displays the form with the data fields completed for data items already available within the EHR. Dr. Hab completes the form and verifies the pre-filled fields. When the form is completed Dr. Hab uses the EHR submit function to electronically submit the form. The EHR asks Dr. Hab to provide his credentials to electronically sign the form. Dr. Hab enters his credentials and the EHR electronically submits the form data to the insurer.

490 **X.4.2.2.2 Capture and Submit Pre-Authorization - Process Flow**

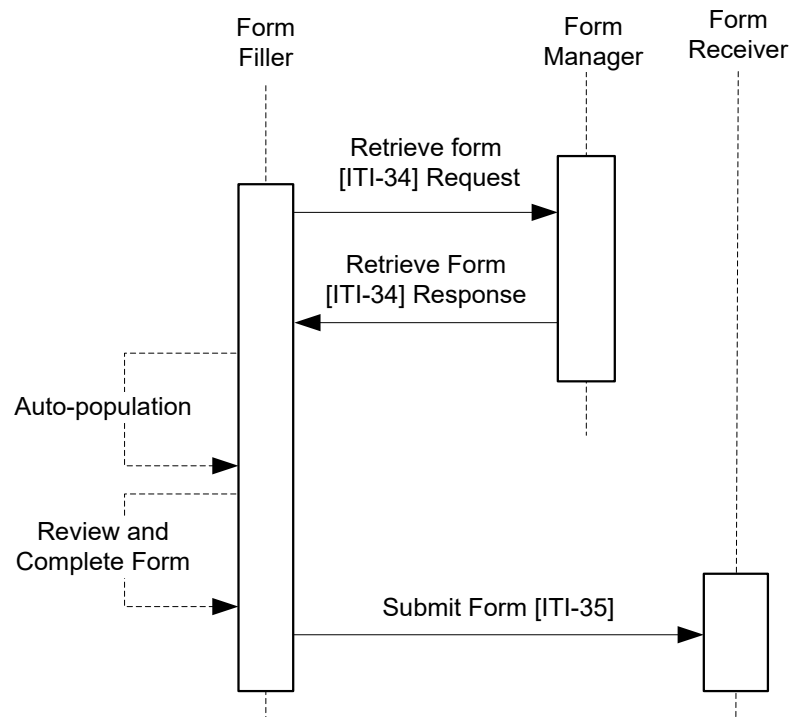


Figure X.4.2.2.2-1: Capture and Submit Pre-Authorization – Process Flow diagram

X.4.2.2.3 Pre-conditions

- The EHR system performs the role of a Form Filler;
- 495 • The EHR is pre-configured to know which form is required;
- The EHR is able to map data from its own data storage to data required in the pre-authorization form, for the purpose of auto-population;
- The EHR already knows the endpoints to submit pre-authorization form request;
- The EHR is able to digitally sign the pre-authorization form;
- 500 • The EHR is able to submit data from the completed form to the insurer;
- The insurer is able to return an electronic version of the requested form;
- The insurer is able to receive and process data submitted from the completed form.

X.4.2.2.4 Main Flow

- Dr. Hab is made aware of the requirement for pre-authorization;

- 505
- Dr. Hab’s EHR retrieves the pre-authorization form;
 - The EHR auto-populates the form, and presents it on screen for Dr. Hab;
 - Dr. Hab completes and digitally signs the form;
 - The data from the completed form is sent to the insurer.

X.4.2.2.5 Post-conditions

- 510
- Data from the completed form has been sent to the insurer;
 - The insurer has received, stored, and processed the pre-authorization data.

X.4.2.3 Use Case #3: Public Health Reporting

The SDC Initiative developed a general use case that illustrates pre-population by an enhanced forms repository. The setting for this use case is public health reporting.

515

X.4.2.3.1 Public Health Reporting – Use Case Description

A Provider has identified a patient with a reportable condition. Using an existing EHR system, the Provider submits an electronic request for an appropriate form from the jurisdictional public health organization or entity.

520 The EHR system, acting as the Form Filler, requests and retrieves the appropriate form from the enhanced form repository acting as the Form Manager. In many cases, the Form Filler may send already collected information about the patient along with the request for appropriate form. The EHR provides a standard export document e.g., a CDA document, to the Form Manager for use in the pre-population step. In this case, the Form Manager uses the information provided in the mapping section in the SDC Form Definition to map explicit location information for each data

525 element within the form to that within the CDA document. The mapping information includes XPath statements that explicitly lead to the correct instance data in the EHR export document.

The EHR system renders the pre-populated form. The provider verifies the pre-populated data, adds any missing data, and then submits the completed structured form to the Public Health Organization. The Provider receives a response that confirms that the report was successfully

530 submitted electronically and received by the public health information system.

X.4.2.3.2 Public Health Reporting – Process Flow

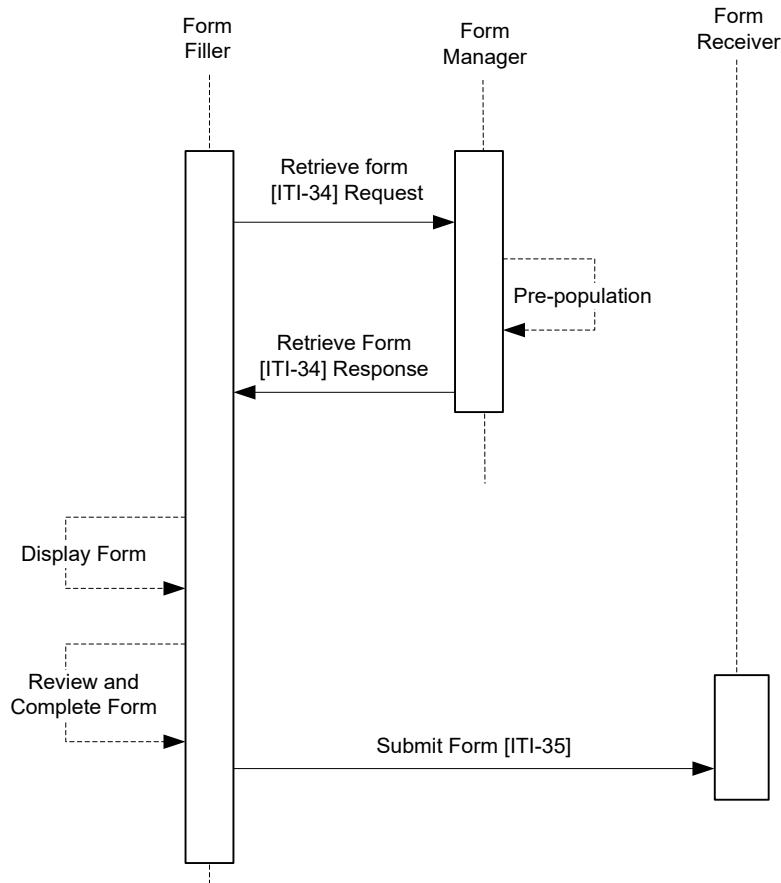


Figure X.4.2.3.2-1: SDC Profile Public Health Reporting - Process Flow diagram

X.4.2.3.3 Pre-conditions:

535 This process is invoked when a healthcare provider is required to submit a report to a public health agency. Necessary pre-conditions include:

- The EHR system performs the role of a Form Filler;
- The EHR system knows how to request the appropriate form;
- The form repository is known and contains the appropriate form;
- The EHR is able to submit data for pre-population;

540

X.4.2.3.4 Main Flow

- The provider is made aware of a required public health report;

- The provider requests the form through the EHR, providing patient data for pre-population;
- 545 • The provider confirms the pre-populated data in the form and completes any missing data;
- The data from the completed form is submitted to the public health agency;

X.4.2.3.5 Post-conditions

- The public health agency stores and uses the data from the completed form;

550 **X.5 SDC Security Considerations**

All SDC actors SHOULD be grouped with a Secure Node or Secure Application in the ATNA Profile.

When SDC actors are grouped with an ATNA Secure Node or Secure Application, that means full compliance with the ATNA requirements. For example,

- 555 • SDC actors that also support ATNA SHALL be able to secure their required transactions according to the requirements in the Authenticate Node [ITI-19] transaction (ITI TF-2a: 3.19).
- SDC actors that also support ATNA SHALL be able to audit their required transactions according to the requirements in the Record Audit Event [ITI-20] transaction (ITI TF-2a: 3.20 and Section X.5.1 below).

560

In accordance with the ATNA Profile (ITI TF-1:9), when configured for use on a physically secured network, SDC implementations may use the normal connection mechanisms, i.e., there may be environments where SDC actors do not need to do mutual TLS as defined in [ITI-19]. However, when configured for use in an environment not on a physically secured network, SDC implementations SHALL use a secure channel such that defined in [ITI-19]. It is expected that the payload used in this use case will cross domains/networks and therefore transport encryption is required.

565

X.5.1 Use of IHE ATNA for Recording Security Audit Events

570 See QRPH TF-2: 5.Z3 Audit Record Considerations (currently in the CRD Trial Implementation Supplement) for the specification for audit messages for the [ITI-34], [ITI-35], and [ITI-36] transactions.

X.5.2 Digital Signature

SDC transactions can contain numerous elements and those elements are inter-related. There will be times when it is critical that a receiver in an SDC transaction be assured that the contents of

575 the entire transaction payload were not altered in transit and the transaction is signed by the sender.

If digital signature is required for SDC transactions, ITI Document Digital Signature (DSG) Profile SHOULD be used.

X.5.3 Consistent Time (CT)

580 See Section X.3. This grouping will assure that all systems have a consistent time clock to assure a consistent timestamp for audit logging.

X.5.4 Cross Enterprise User Assertion (XUA)

585 In order to address identified security risks, all actors in SDC MAY be grouped with Cross-Enterprise User Assertion (XUA) Profile actors as appropriate. This grouping will assure that only highly trusted persons can communicate.

Volume 1 Appendices

Not applicable

Volume 2 – Transactions

590 Section not applicable.

Volume 2 Appendices

Not applicable

Volume 2 Namespace Additions

595

<i>Add the following terms to the IHE General Introduction Appendix G:</i>
--

None

Volume 3 – Content Modules

5 Namespaces and Vocabularies

Add to Section 5 Namespaces and Vocabularies

Not applicable

6 Content Modules

6.3.1 CDA Document Content Modules

Section not applicable.

Q SDC Content Modules

Various datatypes that are supported by the SDC Form Design model are listed in Appendix A, Table A.1.6. These are the W3C datatypes.

Q.1 SDC Pre-Pop Content Module

This Content Module further constrains elements defined in Retrieve Form [ITI-34] request.

Table Q.1-1: SDC Pre-Pop Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
prepopData	The context element that may be used to contain content for the purposes of pre-population. This will contain the CDA R2 document containing the relevant patient information for pre-population	1..1	R	anyXML	The <prepopData> element SHALL contain ONLY CDA R2 document and SHALL NOT be a nil construct.

```
<prepopData>
  <ClinicalDocument xmlns="urn:hl7-org:v3">
    <realmCode code="US"/>
    <!-- Valid CDA-R2 document -->
  </ClinicalDocument>
</prepopData>
```

620 This constraint may be further refined by other IHE profiles, e.g., a specific type of CDA-R2 document may be specified. This constraint also may be further refined by other profiles by specifying how the Form Manager and Form Processor use the pre-pop data.

Q.2 SDC XML Package Content Module

625 A Form Filler that supports this content module option SHALL be able to make a request as per Section Q.2.1 for an SDC XML Package and SHALL use the returned SDC XML Package as per Section Q.2.2 to render and capture data using the SDC XML Package.

All Form Managers and Form Processors SHALL be able to respond to a request for an SDC XML Package as per Section Q.2.1 and SHALL return an SDC XML Package as per Section Q.2.2.

Q.2.1 SDC XML Package – Request

630 This Content Module further constrains elements defined in the Retrieve Form [ITI-34] request.

Table Q.2.1-1: SDC XML Package-Request Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
encodedResponse	Specifies how the form is to be returned.	1..1	R	boolean	Value SHALL be “true”
@responseContentType	An XML attribute of encodedResponse which specifies the type of expected encoded response	1..1	R	string	Value SHALL be “application/xml+sdv”
formID	The identifier of a form. In SDC a form is uniquely defined by its FormDesign ID	1..1	R	string	Represents an SDC format as managed by the responding Form Manager or Form Processor. Each form and its multiple versions (where applicable) will be distinguished using a unique ID, preferably, either a URI or a GUID, within each Form Manager or Form Processor

635

640

645

650

```
<RetrieveFormRequest
  xmlns="urn:ihe:iti:rfd:2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <prepopData xsi:nil="true" />
  <!--Note: In this example, we are only showing the request for XML
Form Definition and hence are not using the pre-pop data-->
  <workflowData>
    <formID>http://myrepo.gov/form_design_id=12345.1/xml</formID>
    <encodedResponse responseType="application/xml+sdv">true
  </encodedResponse>
    <!--Please note the use of constrained value "true" above-->
    <archiveURL />
    <context xsi:nil="true"/>
    <instanceID xsi:nil="true"/>
  </workflowData>
</RetrieveFormRequest>
```

Note: In the above example, the entire URI is a form ID and shown as a suggested example only. In reality, this will be decided by the actors in the [ITI-34] transaction.

Q.2.2 SDC XML Package – Response

This Content Module further constrains elements as defined in Retrieve Form [ITI-34] response.

655

The SDC XML package contains the form design information within a single <XMLPackage> element. In addition, it contains administrative information <Admin>, and optional submission rule <SubmissionRule> and compliance rule <ComplianceRule> information. These additional information packages are separate and independent of the form design and contain helpful information for the Form Filler.

660

Table Q.2.2-1: SDC XML Package-Response Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
Structured	The XML element container for the return of encoded, structured form content. The Structured element SHALL contain one SDCPackage.	1..1	R		
+SDCPackage	The wrapper element container for the SDC-compliant form package.	1..1	R	Table Q.6.2-1	SHALL conform to SDCRetrieveForm Schema. See Section Q.6.3
++Admin	Admin contains information about the package	0..1	O	Table Q.6.2.1-1	

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
++SubmissionRule	Rules and details regarding submission of the completed form	0..*	O	Table Q.6.2.2-1	
++ComplianceRule	Rules and details regarding compliance requirements for the form	0..*	O	Table Q.6.2.3-1	
++XMLPackage	Container element for SDC-compliant form and relevant information	1..1	R	Table Q.6.2.5-1	
+++FormDesign	The SDC-compliant XML Form	1..1	R	Table Q.6.2.6-1	
contentType	The type of the returned form.	1..1	R	string	Value SHALL be “application/xml+sdc”

```

<RetrieveFormResponse
  xmlns="urn:ihe:iti:rfd:2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sdc="urn:ihe:qrph:sdc:2016">
  <form>
    <Structured>
      <SDCPackage>
        <Admin>
          <!-- Contains administrative information -->
        </Admin>
        <XMLPackage>
          <MapTemplate>
            <!--Mapping information goes here-->
          </MapTemplate>
          <FormDesign>
            <!--Contains data elements of the form e.g., header,
body, footer, sections, questions, etc.) -->
          </FormDesign>
        </XMLPackage>
      </SDCPackage>
    </Structured>
    <instanceURI>1.2.3.4.5</instanceURI>
  </form>
  <contentType>application/xml+sdc</contentType>
  <!--Please note the use of constrained value above-->
  <responseCode/>
</RetrieveFormResponse>

```

Q.3 SDC HTML Package Content Module

690 A Form Filler that supports this content module option SHALL be able to make a request as per Section Q.3.1 for an SDC HTML Package and SHALL retrieve an SDC HTML Package as per Section Q.3.2 to render and capture data.

All Form Manager and Form Processors SHALL be able to respond to a request for SDC HTML Package as per Section Q.3.1 and SHALL return an SDC HTML Package as per Section Q.3.2.

695 Q.3.1 SDC HTML Package – Request

This Content Module further constrains elements as defined in Retrieve Form [ITI-34] request.

Table Q.3.1-1: SDC HTML Package-Request Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
encodedResponse	Specifies how the form is to be returned.	1..1	R	boolean	Value SHALL be “true”
@responseContentType	An XML attribute of encodedResponse which specifies the type of expected encoded response	1..1	R	string	Value SHALL be “text/html+sdv”
formID	The identifier of a form. In SDC a form is uniquely defined by its FormDesign ID	1..1	R	string	Represents an SDC format as managed by the responding Form Manager or Form Processor. Each form and its multiple versions (where applicable) will be distinguished using a unique ID, preferably, either a URI or a GUID, within each Form Manager or Form Processor

700

```

<RetrieveFormRequest
  xmlns="urn:ihe:iti:rfd:2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <prepopData xsi:nil="true" />
  <!--Note: for this example, we are presuming no pre-pop data-->
  <workflowData>
    <formID>http://myrepo.gov/form_design_id=12345.2/html</formID>
    <encodedResponse responseContentType="text/html+sd" true
  </encodedResponse>
    <!--Please note the use of constrained values above-->
    <archiveURL />
    <context xsi:nil="true"/>
    <instanceID xsi:nil="true"/>
  </workflowData>
</RetrieveFormRequest>

```

Note: In the above example, the entire URI is a form ID and shown as a suggested example only. In reality, this will be decided by the actors in the [ITI-34] transaction.

Q.3.2 SDC HTML Package – Response

This Content Module further constrains elements defined in Retrieve Form [ITI-34] response.

The SDC HTML package contains the HTML form instance within a single <HTMLPackage> element. In addition, it contains administrative information <Admin>, and optional submission rule <SubmissionRule> and compliance rule <ComplianceRule> information. These additional information packages are separate and independent of the form instance and contain helpful information for Form Filler.

Table Q.3.2-1: SDC HTML Package-Response Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
Structured	The XML element container for the return of encoded, structured form content.	1..1	R		
+SDCPackage	The wrapper element container for the SDC-compliant form package.	1..1	R	Table Q.6.2-1	SHALL conform to SDCRetrieveForm Schema. See Section Q.6.3.
++Admin	Admin contains information about the package	0..1	O	Table Q.6.2.1-1	
++SubmissionRule	Rules and details regarding submission of the completed form	0..*	O	Table Q.6.2.2-1	

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
++ComplianceRule	Rules and details regarding compliance requirements for the form	0..*	O	Table Q.6.2.3-1	
++HTMLPackage	The XML element containing the base64-encoded HTML form instance.	1..1	R		
contentType	The type of the returned form.	1..1	R	string	Value SHALL be “text/html+sd”

```

<RetrieveFormResponse
  xmlns="urn:ihe:iti:rfd:2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sd="urn:ihe:qrph:sd:2016">
  <form>
    <Structured>
      <SDCPackage>
        <Admin>
          <!-- Contains administrative information -->
        </Admin>
        <HTMLPackage>
          <!--The base64-encoded HTML form instance goes here -->
        </HTMLPackage>
      </SDCPackage>
    </Structured>
    <instanceURI>2.3.4.5.6</instanceURI>
  </form>
  <contentType>text/html+sd</contentType>
  <!--Please note the use of constrained value above-->
  <responseCode/>
</RetrieveFormResponse>

```

The base64-encoded HTML form instance can be generated using any available technology. The SDC HTML form instance shall contain a “submit” button along with the necessary script and relevant logic to generate the submit form output as described in Section Q.5 and send it to the endpoint(s) in the submission rule(s).

Q.4 SDC URI Form Content Module

The Form Filler claiming this content module option SHALL be able to make a request as per Section Q.4.1 for SDC URI Form and SHALL retrieve the SDC URI Form per Section Q.4.2.

All Form Managers and Form Processors SHALL be able to respond to a request for SDC URI Form as per Section Q.4.1 and SHALL return SDC URI Form as per Section Q.4.2.

Q.4.1 SDC URI Form – Request

This Content Module further constrains elements as defined in Retrieve Form [ITI-34] request.

Table Q.4.1-1: SDC Form URI-Request Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
encodedResponse	Specifies how the form is to be returned.	1..1	R	boolean	Value SHALL be “false”

```
<RetrieveFormRequest
  xmlns="urn:ihe:iti:rfd:2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <prepopData xsi:nil="true" />
  <workflowData>
    <formID>http://myrepo.gov/form_design_id=12345.3/url</formID>
    <encodedResponse>>false</encodedResponse>
    <!--Please note the use of constrained value "false" above-->
    <archiveURL />
    <context xsi:nil="true"/>
    <instanceURI xsi:nil="true"/>
  </workflowData>
</RetrieveFormRequest>
```

Note: In the above example, the entire URI is a form ID and shown as a suggested example only. In reality, this will be decided by the actors in the [ITI-34] transaction.

Q.4.2 SDC URI Form – Response

This Content Module further constrains the <contentType> element as defined in Retrieve Form [ITI-34] response.

Table Q.4.2-1: SDC Form URI-Response Content Module Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
URL	The XML element container for the return of a pointer to the form.	1..1	R	anyURI	
contentType	The type of the returned response.	1..1	R	string	Value SHALL be “URL”

```
<RetrieveFormResponse
  xmlns="urn:ihe:iti:rfd:2007"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sd="urn:ihe:qrph:sd:2016">
  <form>
    <URL>
      <!--URL FOR THE INSTANCE OF THIS FORM GOES HERE-->
      www.weBeForms.com/1.2.3.4.5
    </URL>
    <instanceURI>3.4.5.6.7</instanceURI>
  </form>
  <contentType>URL</contentType>
  <!--Please note the use of constrained value above-->
  <responseCode/>
</RetrieveFormResponse>
```

Q.5 SDC Submission Data Content Module

The SDC Submission Data content module specifies the structure in which the form data needs to be sent to the Form Receiver. Using this content module, the Form Filler sends the entire form along with the responses. This will allow Form Receivers and Form Processors to receive and process form data. Since the entire form is included, Form Receivers and Form Processors will be able to validate data to a more granular level e.g., validating that the response to the question has correct LOINC or SNOMED CT codes.

The Form Filler SHALL be able to generate and submit structured form data as shown in Section Q.5.1.

The Form Receiver and Form Processor SHALL be able to receive and validate (against the schema file) the submitted form data and SHALL respond to the Form Filler as shown in Section Q.5.2.

Q.5.1 SDC Submission – Request

This Content Module further constrains elements defined in Submit Form [ITI-35] Request.

Table Q.5.1-1: SDC Submit Form Request Element Constraints

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
SubmitFormRequest	The top-level container element	1..1	R		This element SHALL contain a single <SDCSubmissionPackage> element containing form data.

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
+SDCSubmissionPackage	The XML element that contains the form data	1..1	R		
++FormDesign	The top-level element containing form data items with responses	1..1	R	Table Q.6.2.6-1	
++DemogFormDesign	The optional top-level element containing demographic form data items	0..1	O	Section Q.6.2.5.1	

820

825

830

835

840

```
<?xml version="1.0" encoding="UTF-8"?>
<rfd:SubmitFormRequest xmlns:sd="urn:ihe:qrph:sd:2016"
  xmlns:rfd="urn:ihe:iti:rfd:2007">
  <!-- This is the format in which Form Filler will submit form data to Form
  Receiver -->
  <SDCSubmissionPackage>
    <FormDesign ID="2.004.001">
      <Body ID="S1" styleClass="body">
        <ChildItems>
          <Question ID="24853" mustImplement="false"
instanceGUID="a4f0cc5b-f90e-4c01-a4fa-99f63c01b4f3" parentGUID="199a4839-
a3e8-452a-bf79-7adc9e5f2acb">
            <ListItem ID=" 20907" title="Other (specify)"
selected="true">
              <OtherText type="reportText" val="Other findings:" />
              <ListItemResponseField>
                <Response>
                  <string val="hyperpigmentation" />
                </Response>
              </ListItemResponseField>
            </ListItem>
          </Question>
        </ChildItems>
      </Body>
    </FormDesign>
  </SDCSubmissionPackage>
</rfd:SubmitFormRequest>
```

- 845 When submitting form data, the Form Filler can remove the unselected ListItems and unanswered questions, along with their associated metadata, if a smaller payload is required. Similarly, for the SDC HTML forms, the “submit” button functionality can be designed to remove these unselected ListItems and unanswered questions.

Q.5.2 SDC Submission - Response

- 850 This Content Module further constrains elements defined in Submit Form [ITI-35] response.

Table Q.5.2-1: SDC Submit Form Response Element Constraint

Element Name	Description	Card.	Optionality	Data Type	Value Constraint
SubmitFormResponse	The top-level container element	1..1	R		SHALL contain a <responseCode>
+content	Content returned	0..1	R		The value shall contain the SDCSubmissionPackage sent in the preceding SubmitFormRequest with an update to the formInstanceVersionURI
+contentType	The type of data received	1..1	R	string	Value SHALL be “application/xml+sdc”
+responseCode	This element contains a response code to the form submission	1..1	R	string	The value can be an HTTP status code e.g., “200” for OK or “404” for not found, etc.

Q.6 SDC Form Definition Model

855 SDC Forms address the need for systems to interoperate by exchanging data that has been defined as part of a structured document or form. This section provides an overview for the approach and representation of the SDC form definition model.

Q.6.1 Scope and Approach

860 The role of forms in interoperability is recognized by the ISO/IEC 19763-13 Metamodel for Forms Registration (MFI-13) standard on which the SDC Form Definition is based. The standard defines a universal metamodel for forms devoid of specific domain knowledge, which allows documentation and registration of form designs, both paper and electronic, from any and all sources. MFI-13 inherits from ISO/IEC 11179 MDR-3 (MDR-3), which provides classes and types that support the identification, naming, registration, and administration of form designs and other supporting documents. The form design can be associated with appropriate entity-relationship diagrams or data models so that data and semantics may be faithfully exchanged between systems and so that those data may be compared, joined, or composed for analysis. This is accomplished through the mapping of questions on the form to data elements that are part of data or information models.

870 The basic structure of an SDC form contains one or more sections, such as Header, Footer, or Body. Each section contains one or more questions. Each response to a question, an answer, is stored as a discrete unit of data. Sometimes the answer to one question determines the next

question or section that should be presented, or is used in a calculation of data value(s). All of these different types of items are referred to as Form Elements.

Q.6.2 SDC Package

The sections below describe the SDC Package, which includes required administrative information <Admin>, optional Submission rule details <SubmissionRule>, optional compliance rule information <ComplianceRule>, and either an XML Package <XMLPackage> or an HTML Package <HTMLPackage>.

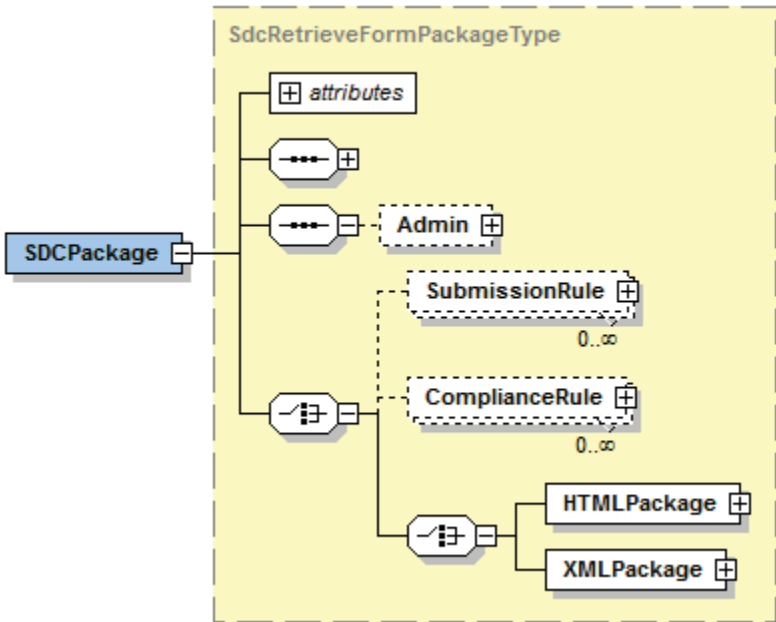


Figure Q.6.2-1: SDC Package Diagram

Table Q.6.2-1: SDC Package Elements

Element Name	Element Description	Card	Data Type
SDCPackage	Top level element		
@name	Developer assigned identifier, similar to a unique control/object name, used to provide the ability for programmatic manipulation of an element.	0..1	ID
@type	The @type attribute can contain custom metadata “tokens” for the element	0..1	NMTOKENS
@styleClass	Developer assigned class name for display styling, generally for use with an external style sheet.	0..1	NMTOKENS
@order	Allows the form template developer to define a sequential order for elements in a template for display purposes	0..1	decimal

Element Name	Element Description	Card	Data Type
@packageID	Unique ID used to identify an instance of the package.	1..1	anyURI
+Admin	The Admin element contains administrative information related to the package.	0..*	Table Q.6.2.1-1
+SubmissionRule	Rules and details regarding submission of the completed form	0..*	Table Q.6.2.2-1
+ComplianceRule	Rules and details regarding compliance requirements for the form	0..*	Table Q.6.2.3-1
+HTMLPackage	HTML form and relevant information	0..1	Section Q.6.2.4
+XMLPackage	XML form along with relevant information	0..1	Table Q.6.2.5-1

885 Q.6.2.1 Admin

Admin contains information about a package, including a description of the package contents and purpose (PackageDescription), information about the registry that contains the package file (RegistryData), and information about the package file characteristics (TemplateFile).

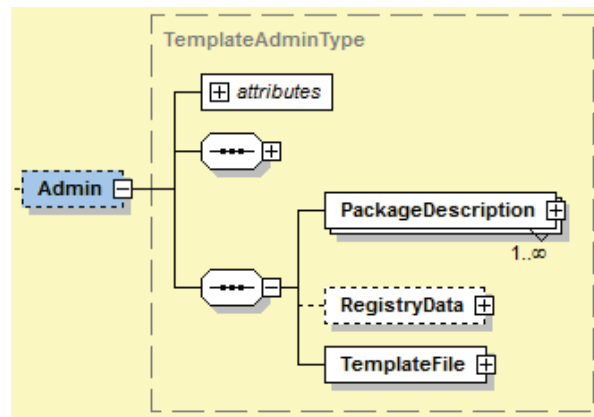


Figure Q.6.2.1-1: Admin Diagram

Table Q.6.2.1-1: Admin

Element Name	Element Description	Card	Data Type
Admin	Top level element		
+PackageDescription	Description of the XML package contents and the purpose for the contained XML templates	1..*	string
+RegistryData	Information about the registry that contains the XML template, and registration status of the package files	0..*	
+TemplateFile	Information about the package file characteristics	1..1	

Q.6.2.2 SubmissionRule

895 SubmissionRule contains information related to submission of the completed form. It contains the endpoint URI for the destination and optional description as well as information about the organization. Any specific details or instructions pertaining to the submission can be included in the optional <RuleDescription> element.

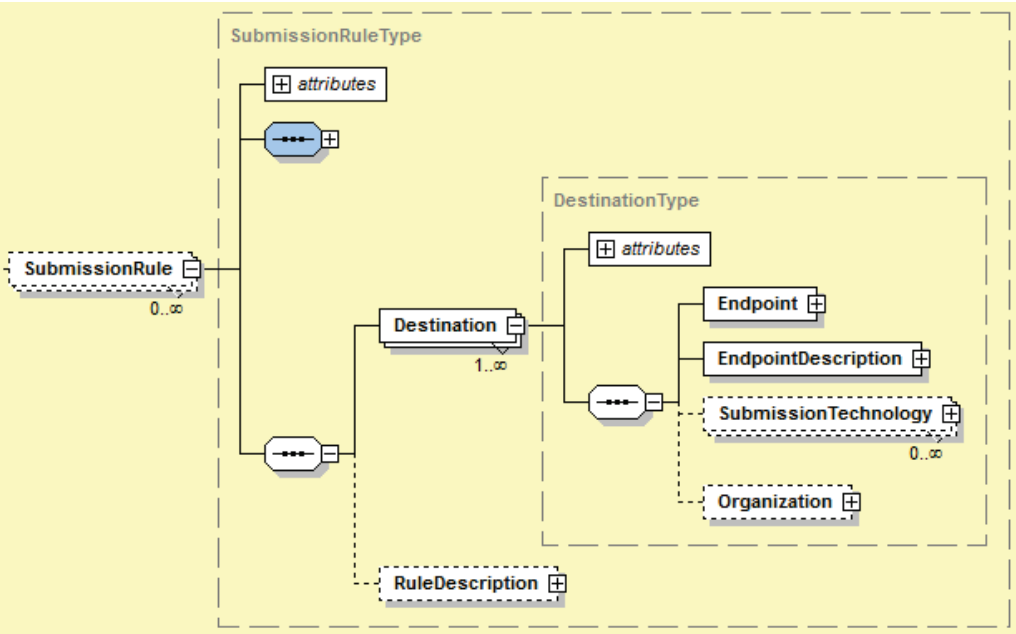


Figure Q.6.2.2-1: SubmissionRule Diagram

Table Q.6.2.2-1: SubmissionRule

Element Name	Element Description	Card	Data Type
SubmissionRule	Top level element		
+Destination	Container element for the endpoint information	1..*	
++Endpoint	The endpoint URI information where the form data need to be submitted	1..1	anyURI
++EndpointDescription	Text description of the endpoint	1..1	string
++SubmissionTechnology	Descriptive information about technology used for submitting form data	0..*	string
++Organization	Name of the receiving organization	0..1	Table A.1.1-1
+RuleDescription	Narrative description/information about submitting form data	0..1	string

Q.6.2.3 ComplianceRule

The ComplianceRule element contains instructions related to compliance with regards to form submission. The information related to the Organization responsible for defining compliance rules can be captured within <ComplianceOrganization> element.

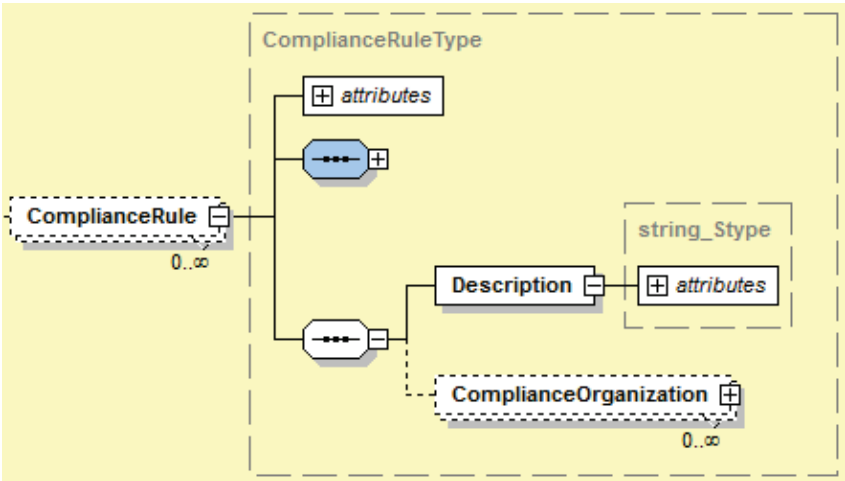


Figure Q.6.2.3-1: ComplianceRule Diagram

Table Q.6.2.3-1: ComplianceRule

Element Name	Element Description	Card	Data Type
ComplianceRule	Top level element		
+Description	General information about how information in a particular form package must be handled	1..1	string
+ComplianceOrganization	Name of the organization responsible for compliance	0..*	Table A.1.1-1

Q.6.2.4 HTMLPackage

The HTMLPackage element contains the base64-encoded SDC-compliant HTML form instance. This base64-encoded HTML form is derived from the SDC XML Form definition and retains all the required ID and other attribute information from the form definition to be able to generate the required data for submission. The base64-encoded form will be included within the <HTMLPackage> element.

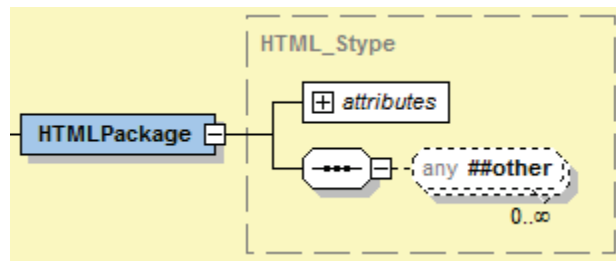


Figure Q.6.2.4-1: HTMLPackage Diagram

Q.6.2.5 XMLPackage

- 920 The XMLPackage element contains a primary XML form instance <FormDesign>, along with an optional Demographic form <DemogFormDesign>, optional mapping information <MapTemplate>.

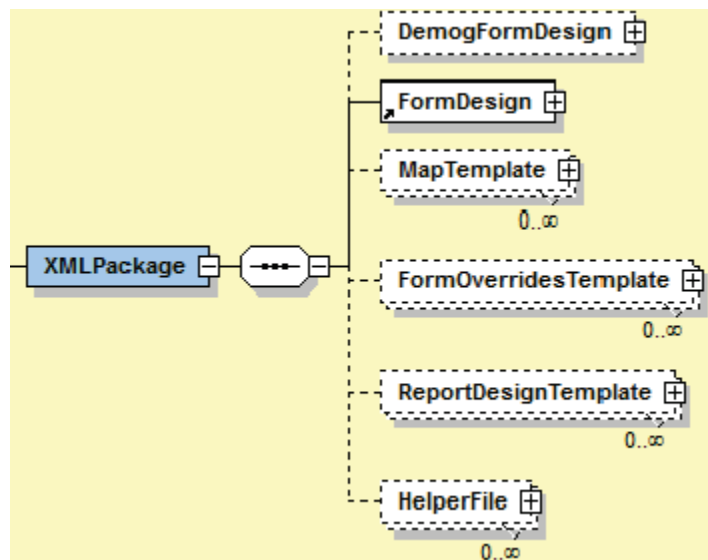


Figure Q.6.2.5-1: XMLPackage Diagram

Table Q.6.2.5-1: XML Package

Element Name	Element Description	Card	Data Type
XMLPackage	Top level element		
+FormDesign	The top level element defining SDC form definition and containing	1..1	Table Q.6.2.6-1
+MapTemplate	Describes mappings between FormDesign items and data elements, terminologies, databases, XML files, local values, etc.	0..*	Table Q.6.2.5.2-1

Element Name	Element Description	Card	Data Type
+DemogFormDesign	General information about how information in a particular form package must be handled	0..1	Section Q.6.2.5.1
+FormOverridesTemplate	Used when behavior for a specific element in a form needs to be changed for a given local instance. The overriding behavior will only be in effect for that local instance	0..*	
+ReportDesignTemplate	Describes the information content of a report (e.g., sections, questions etc.)	0..*	
+HelperFile	Any supplemental file(s) required for the display of the primary FormDesign or ReportDesignTemplate	0..*	

Q.6.2.5.1 DemogFormDesign

930 The <DemogFormDesign> is an optional container element, specifically used for exchanging Demographic form items. In many cases, the primary form being exchanged does not contain any patient demographic information and hence a need to exchange data related to patient demographic separately. The structure of this demographic form is the same as that of the primary form <FormDesign> and the word “Demog” is added for distinction only.

Q.6.2.5.2 MapTemplate

935 Describes mappings between FormDesign items and data elements, terminologies, databases, XML files, local values, etc.

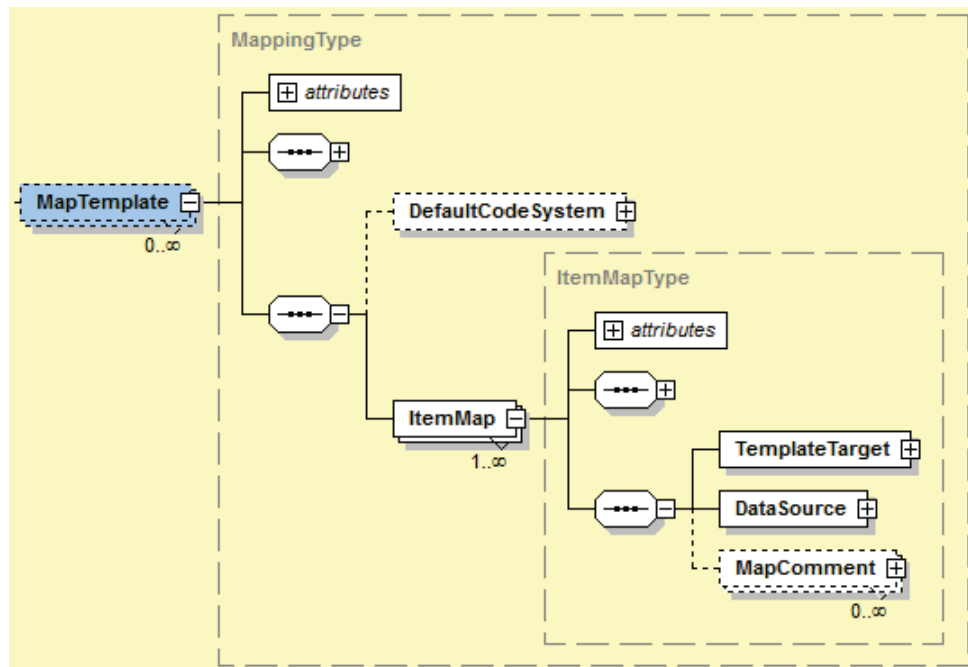


Figure Q.6.2.5.2-1: MapTemplate Diagram

940

Table Q.6.2.5.2-1: MapTemplate

Element Name	Element Description	Card	Data Type
MapTemplate	Top level element		
+DefaultCodeSystem	Specifies the default coding system to be applied for all elements listed in the ItemMap entries	0..1	Table Q.6.2.5.2.1-1
+ItemMap	Entries under this element map a target item in a FormDesign to a terminology code, local value, XML-based document entry, database record, RDF store triple, etc.	1..*	
++TemplateTarget	Container element for Target item in a FormDesign	1..1	
+++TargetItemID	Unique ID for the target item	1..1	anyURI
+++TargetDisplayText	Optional display text associated with the target item	0..1	string
++DataSource	The DataSource is an object that maps to a target item in a FormDesign. DataSources objects can include terminology codes, local values, XML-based document entries, database records, RDF store triples, etc.	1..1	
++MapComment	Optional comments related to mapping information	0..*	string

Q.6.2.5.2.1 DefaultCodeSystem

The DefaultCodeSystem represents information about the coding system used for mapping. It may refer to any type of coding, terminology, classification, keyword, or local value system.

945

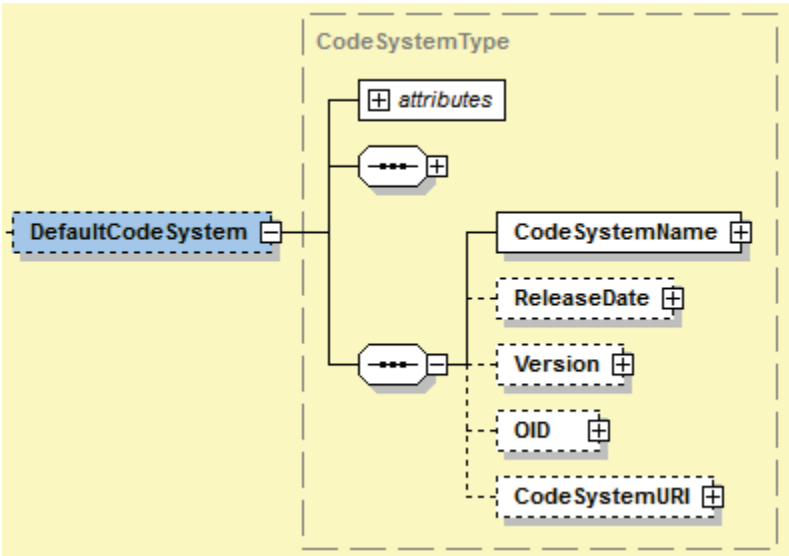


Figure Q.6.2.5.2.1-1: DefaultCodeSystem Diagram

Table Q.6.2.5.2.1-1: DefaultCodeSystem

Element Name	Element Description	Card	Data Type
DefaultCodeSystem	Top level element		
+CodeSystemName	The name of the coding system, as recommended by the coding system curators	1..1	string
+ReleaseDate	The date the selected version of the coding system was released for general use	0..1	dateTime
+Version	Version of the coding system, using the version format defined by the coding system	0..1	string
+OID	The ISO object identifier (OID) for the coding system, as found at the HL7 OID Registry	0..1	string
+CodeSystemURI	Web resource that uniquely identifies the coding system	0..1	URI

950

Q.6.2.6 FormDesign

The FormDesign is the root element defining the SDC form. The form definition consists of a required Body, with optional Header and Footer. The Body, Header, and Footer sections are identical in structure and elements.

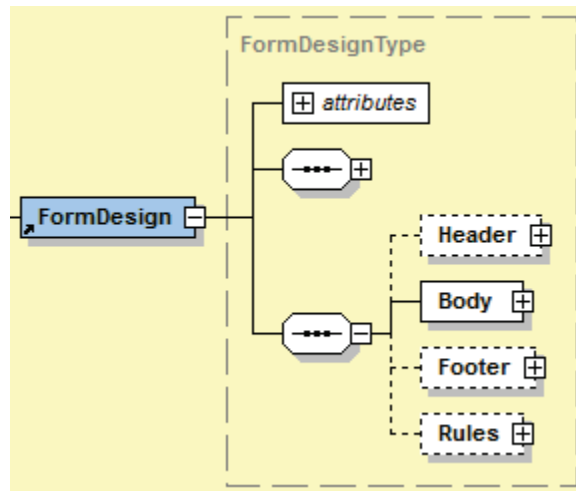


Figure Q.6.2.6-1: FormDesign Diagram

Table Q.6.2.6-1: FormDesign

Element Name	Element Description	Card	Data Type
FormDesign	Top level element		
@name	Developer assigned identifier, similar to a unique control/object name, used to provide the ability for programmatic manipulation of an element.	0..1	
@type	The @type attribute can contain custom metadata "tokens" for the element	0..1	NMTOKENS
@styleClass	Developer assigned class name for display styling, generally for use with an external style sheet.	0..1	NMTOKENS
@order	Allows the form template developer to define a sequential order for elements in a template for display purposes	0..1	Decimal
@ID	Unique ID	1..1	ID
@baseURI	URI associated with the form design	0..1	anyURI
@basedOnURI	URI used to identify the form that this FormDesign is based upon. In most cases, this should be a standard form that is modified and/or extended by the current FormDesign.	0..1	anyURI
@formTitle	human readable title for display when choosing forms.	0..1	string
@filename	Filename to use when the current FormDesign instance is saved as a file. For forms containing responses, the filename may include the formInstanceVersionURI, but	0..1	string

Element Name	Element Description	Card	Data Type
	the naming convention may be use-case-specific.		
@lineage	A text identifier that is used to group multiple versions of a single form. The lineage is constant for all versions of a single kind of form. When appended to @baseURI, it can be used to retrieve all versions of one particular form.	1..1	string
@version	@version contains the version text for the current form. It is designed to be used in conjunction with @baseURI and @lineage.	1..1	string
@fullURI	The full URI that uniquely identifies the current form. It is created by concatenating @baseURI + lineage + version. Each of the components is separated by a single forward slash.	1..1	string
@prevVersionURI	The full URI used to identify the form that is the immediate previous version of the current FormDesign	0..1	string
@formInstanceURI	Unique URI used to identify a unique instance of a form. This URI does not change for each edit session of a form instance.	0..1	anyURI
@formInstanceVersionURI	Unique URI used to identify a unique instance of a form. This URI must change for each edit session of a form instance.	0..1	anyURI
@formPreviousInstanceVersionURI	Unique URI used to identify the immediate previous instance of a form containing responses. Used for tracking form responses across time and across multiple episodes of editing by end-users. This URI must change for each edit session of a form instance.	0..1	anyURI
+Header	Optional Section that stays at the top of a form	0..1	Table Q.6.2.6.1-1
+Body	Main Section of form	1..1	Table Q.6.2.6.1-1
+Footer	Optional Section that stays at the bottom of a form.	0..1	Table Q.6.2.6.1-1
+Rules	Rules are procedures that affect form behavior. They are usually triggered by changes to the form by a form user.	0..*	

Q.6.2.6.1 Form Structure

960 The form structure contains primary SDC components that define SDC form content and behavior, including Section, Question, DisplayedItem, and ButtonAction. The ChildItems

element provides a hierarchical nesting wrapper for the 4 elements described above as defined in the SDC schema.

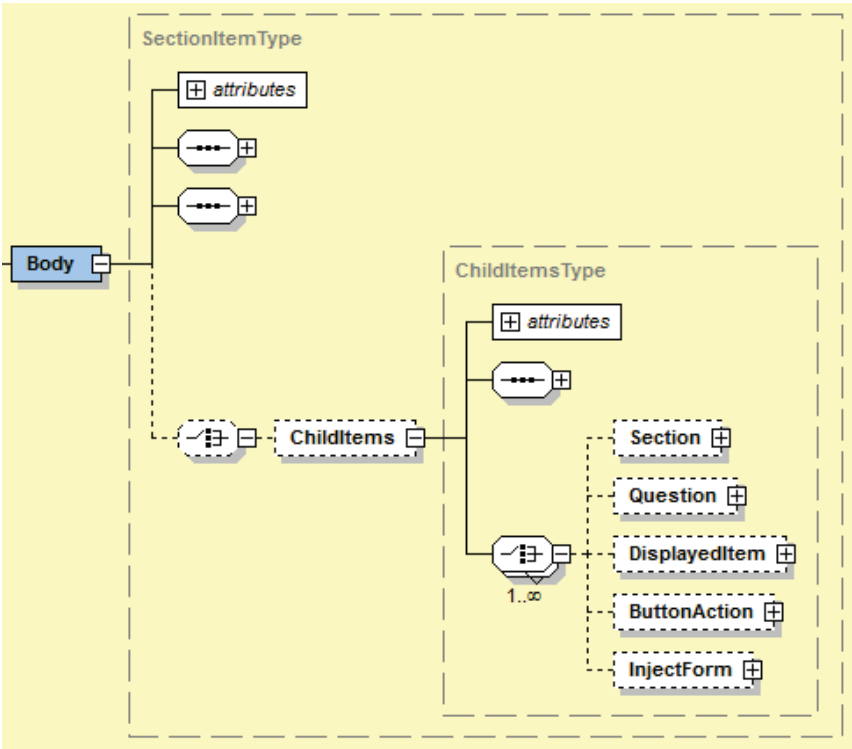


Figure Q.6.2.6.1-1: Form Structure Diagram

Table Q.6.2.6.1-1: Form Structure

Element Name	Element Description	Card	Data Type
ChildItems	Top level element containing the items forming Header, Body, or Footer sections		
+Section	Sections are distinct parts of the form that groups items for a similar purpose	0..*	
+Question	A Question makes a request for information.	0..*	Table Q.6.2.6.2-1
+DisplayedItem	Free-floating text that can appear anywhere on a form. They are not attached to any particular item Section, Question, or ListItem	0..*	Table Q.6.2.6.5-1
+ButtonAction	A Button is a visible area that can be selected ("clicked") to trigger event code to perform an action. It is ordinarily implemented with an image that looks like a rectangular button object, but a button can look like anything.	0..*	
+InjectForm	Link to a form (or portion of a form) that is imported into the current form at a specific location.	0..*	

Element Name	Element Description	Card	Data Type
Property	Property is any type of secondary text that is associated with any form item (e.g., section, question, list item) in any context. The Description context is specified by the @type attribute.	0..*	
Extension	The Extension element provides a place to insert XML that is not defined in the SDC Schema. The sub-elements of Extension must provide a namespace (and ideally an XML Schema) for any non-SDC elements and attributes that are introduced. The Extension element provides almost infinite expansion flexibility for SDC. However, it also requires that form designers and implementers agree on the supported extensions, agree on which elements to accept and the contexts in which they may be used, document this usage, and create an extension validation mechanism to enforce correct usage and detect incorrect usage.	0..*	
Comment	Comments may be provided by the Form designer, the Form implementation code, or by the Form user. The Comment content is placed in the val attribute of the Comment element.	0..*	
BeforeLoadForm	This event is fired before the page is loaded into memory, and before stored form data is loaded. It may be used, e.g., for authentication, to retrieve/prepare stored data, and/or to control form rendering according to user preferences.	0..1	
BeforeLoadData	This event is fired after the page is loaded into memory, before stored form data is loaded, and before the form is visible. For example, It may be used to determine the data to be loaded and to perform the data loading.	0..1	
BeforeShowForm	This event is fired after the page is loaded is memory, after the data is loaded into the form, but before the form is displayed. It may be used to perform form activities that are controlled by the loaded data.	0..1	
BeforeDataSubmit	This event is fired after the page is loaded is memory, after the data is submitted from the form to the Form Receiver. It may be used to alter, validate, or cancel the submission process.	0..1	
BeforeCloseForm	This event can check for changed data that has not been submitted and optionally warn the user before closing the form.	0..1	
@minSelections	minSelections has a minimum value 1, to indicate the number of answer choices that must be selected by the user. 1 is the default value if not set. If set to 2 the user must pick 2 or more choices.	0..*	
@maxSelections	Maximum number of answer choices (list items) that can be selected by the user. Must be greater than or equal to minSelections, and no larger than the total number of list items. A value of 1 (the default) indicates that the question list is single-select. (Abbreviated as QS)	0..*	

Element Name	Element Description	Card	Data Type
@minCard	@minCard is the minimum number of repetitions allowed for a section or question. The default value is 1, indicating that the user must answer any question(s) that has this attribute set. If @minCard is set to 0, then the item and all descendent questions are optional to answer.	0..*	
@maxCard	@maxCard is the maximum number of repetitions allowed for a section or question. The default value is 1, indicating that the section or question cannot be repeated on the data entry form. A value of 0 indicates that the number of repeats is unlimited. If @maxCard is not 0, then Must be greater than or equal to minCard.	0..*	

Q.6.2.6.2 Question

A Question makes a request for information. The information is captured in an answer field.

There are two types of answer fields, each with a set of applicable attributes and rules constraining the answer. These are: ListField and TextField. While a Question on a form may be designed using any of these types, only one of these types may be used for an individual Question. A Question may further contain nesting of any ChildItems – Section, Question, DisplayedItem, Button Action, or InjectForm.

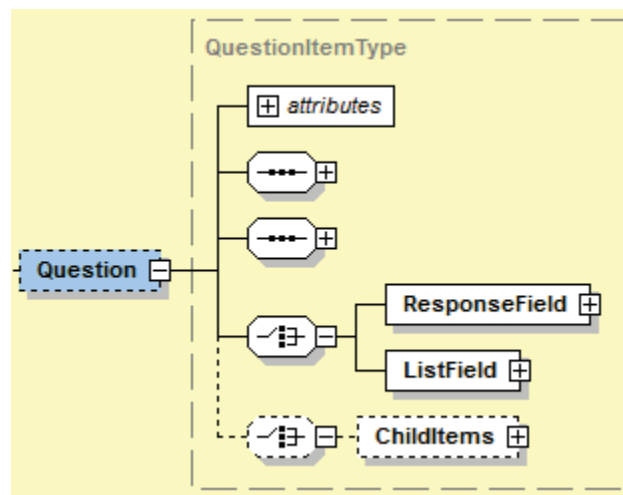


Figure Q.6.2.6.2-1: Question Diagram

The following table describes the components of a question element:

Table Q.6.2.6.2-1: Question

Element Name	Element Description	Card	Data Type
Question	Top level element		
+ListField	List field is a response field in which a list of predefined answers are allowed. . Note: while a Question on the form may be designed using any of these two types of answer fields, only one may be used for an individual Question.	0..1	Table Q.6.2.6.3-1
+ResponseField	ResponseField is a text field in which any value may be entered, subject to pattern, maximum length and unit of measure and constrains applicable to the datatype. Note: while a Question on the form may be designed using any of these two types of answer fields, only one may be used for an individual Question.	0..1	Table Q.6.2.6.4-1

980

Q.6.2.6.3 ListField

List refers to a set of possible answer choices for the user to select. Child DisplayedItems may be interspersed with Child ListItems. At least one ListItem must be present for a multi-select Question, and at least two ListItems must be present for a single-select Question.

985

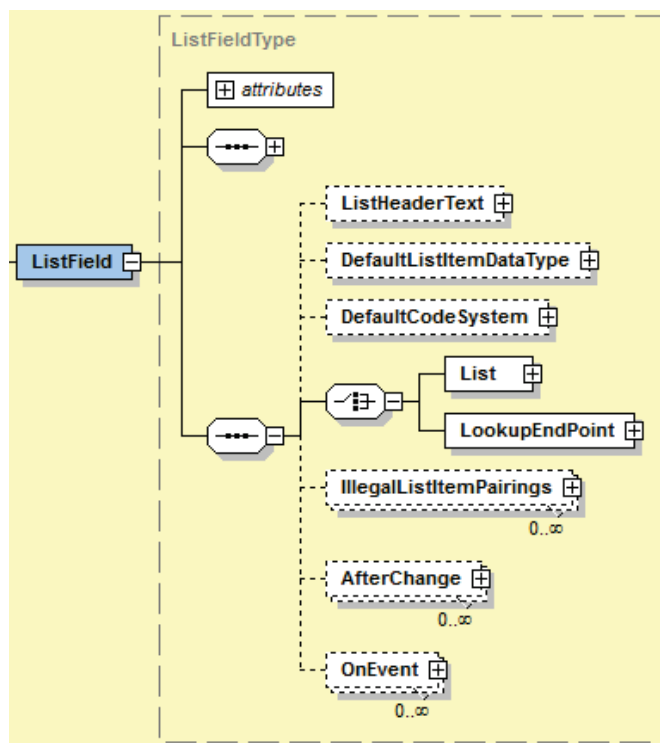


Figure Q.6.2.6.3-1: List Field Diagram

Table Q.6.2.6.3-1: ListField

Element Name	Element Description	Card	Data Type
ListField	Top level element		
+List	List refers to a set of possible answer choices for the user to select	1..1	
++ListItem	Items within the list—a container element	1..*	
+++ListItemResponseField	A place to enter values (of any data type) that are directly associated with, and attached to, a selected answer choice	0..1	Table Q.6.2.6.4-1
+ListHeaderText	The header row for a set of list items	0..1	string
+LookupEndPoint	LookupEndPoint is used when the list items are derived from a web service call of some type, instead of an explicit set of ListItem nodes specified in the FormDesign XML	1..1	
+DefaultListItemDataType	Default data type	0..1	Table Q.6.2.5.2.1-1
+DefaultCodeSystem	Default Code system to be used within a given list	0..1	
+IllegalListItemPairings	Defines list item pairing that is not considered valid This Rule specifies a set of ListItemNames that cannot be selected (@listItemNames) when a test ListItem (@testItemName) is selected. Renamed from: IllegalSelectionTest	0..*	
@listItemName	Set of @name references for a set ListItemNames, delimited by spaces.	0..1	
@testItemName	The ListItem that is being evaluated for the presence of disallowed selections (in @listItemNames). This attributes contains the @name of one ListItem.	0..1	
@selectionDisablesChildren	If set to true, then selecting this ListItem must deactivate all descendant parts of the form, and ignore any user-entered values in the deactivated part. Deselecting the ListItem should reactivate the descendant items in their state at the time the items were deactivated. If items are disabled, then any data stored in the disabled questions should be removed.	0..1	
@selectionDeselectsSiblings	If the ancestor ListField has @multiselect = 'true', then selecting this ListItem should de-select all other ListItem (sibling) nodes except the current one. For example, if a question has choices “A,” “B,” “C,” and “None of the Above”, selecting “None of the Above” will deselect any selections made in “A,” “B,” and “C,”	0..1	

Element Name	Element Description	Card	Data Type
@colTextDelimiter	Character in the DisplayText that separates the columns and rows in a single or multi-column list.	0..1	
@numCols	Number of columns in the list	0..1	
@storedCol	Determines which column of the list is stored in a database. This list is one-based.	0..1	
@defaultListItemDataType	This element is used only if the ListItems are all associated with coded values from a single coding system and there are various coding systems used throughout the form.	0..1	
@ordered	If false, then the form implementation may change the order of items in the list.	0..1	
IllegalCoSelectedListItem	This Rule restricts combinations of co-selected ListItems (answers). In other words, only zero or one item in the @listItemNames list may be selected at a time. In most cases, all ListItems should be children of a single multi-select question. Legal ("allowed") selections evaluate to false Illegal selections evaluate to true These values can be reversed if @not is set to true Renamed from: IllegalSelectionSets	0..*	

Q.6.2.6.4 ResponseField

- 990 The ResponseField is the container element for actual response text. The optional TextAfterResponse element can be used to display additional information next to the field. The ResponseUnits element can be used to capture the unit of measure. The OnEvent, AfterChange, and SetResponseValue elements can be used to trigger events based on business rules.

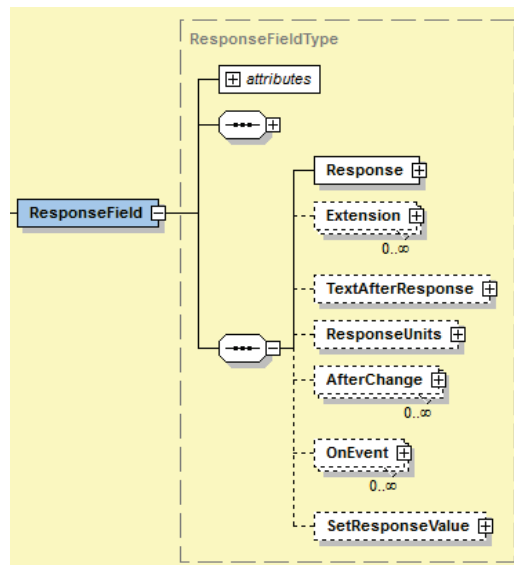


Figure Q.6.2.6.4-1: ResponseField Diagram

Table Q.6.2.6.4-1: ResponseField

Element Name	Element Description	Card	Data Type
ResponseField	Top level element		
+Response	Contains the actual response text	1..1	Table A.1.6-1
+TextAfterResponse	A fixed text that appears after (to the right of) the user's response on the data entry form. This may be text for units such as, mm, cm, etc.	0..1	string
+ ResponseUnits	Represents the measurement standard and its notation for quantifiable objects e.g., miles, kg., etc. The default system for standard notations is UCUM.	0..1	string
+AfterChange	Event that occurs after the Response value is changed, usually fired after a user leaves the Response field. Reserved for later use	0..*	
+SetResponseValue	Expression that is used to automatically set the Response value. Reserved for later use	0..1	
@instanceGUID	A globally unique string assigned to a repeating Question or Section, Injected Form, or any selected answer ListItem. This attribute's value is assigned at the time that answers are entered into a form, to unambiguously identify a single instance among sections or questions, including those that are allowed to repeat and nest deeply.	0..*	
@parentGUID	A globally unique string assigned from a Section, Question, ListItem or InjectedForm to the ID of its parent node (S, Q, LI or IF). This attribute's value is assigned at the time that answers are entered into a	0..*	

Element Name	Element Description	Card	Data Type
	form, to unambiguously identify a single instance among the above objects, including those that are allowed to repeat and nest deeply.		
SetValue	Script that may return a datatype based on information elsewhere in the form to run a calculation, or assign a score.	0..*	
CallSetValue	Function or web service that returns a string value.	0..*	

Q.6.2.6.5 DisplayedItem

1000 DisplayedItems are free-floating items that can appear anywhere on a form. However, DisplayedItems are not “attached” to any specific item Section, Question, or ListItem. A DisplayItem contains Note, Blob (Binary Large Object), and/or Link objects, and may also include Contacts and CodedValues that relate to the Note/Blob/Link objects.

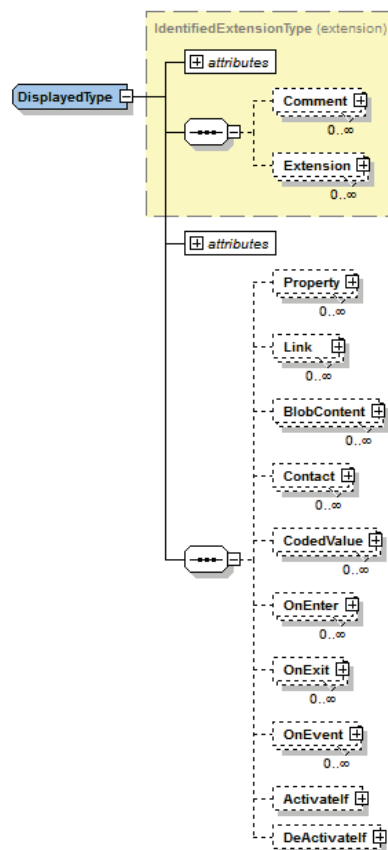


Figure Q.6.2.6.5-1: DisplayedItem Diagram

Table Q.6.2.6.5-1: DisplayedItem

Element Name	Element Description	Card	Data Type
Link	Link to external information.	0..*	string
BlobContent	Blob information associated with the current DisplayedType. The blob may contain any type of special media, but must be base 64 encoded.	0..*	Table A.1.5-1
Contact	Any desired contact information regarding, e.g., the form item, authoring process, certifications, code or coding process.	0..*	Table A.1.3-1
CodedValue	Coding, terminology, classification, keyword, or local values.	0..*	string
OnEnter	Generic event handler occurring when value is entered in specific field. eventName must be specified.	0..*	
OnExit	Generic event handler occurring on exiting the form field. eventName must be specified.	0..*	
OnEvent	Generic event handler occurring at the time of the event. eventName must be specified.	0..*	
ActivateIf	Activate the DisplayedType item if the contained conditions are met.	0..*	
DeActivateIf	DeActivate the DisplayedType item if the contained conditions are met.	0..*	
@title	The primary text to show on the form. Also known as "prompt" or "label" or "visibleText" or "caption"	0..*	
@enabled	Determines whether the user can interact with the displayed item when the form is first displayed.	0..*	
@visible	Determines whether the item should be visible on a computer screen when the form is first displayed.	0..*	
@mustImplement	If this attribute is set to "true" (the default), then the form implementation must make this item available for use on the form. If this attribute is set to "false" on a parent or ancestor item, then all child/descendant items inherit this "false" setting, even if they have this attribute set to "true." In other words, the child/descendent items do not need to be implemented if any parent/ancestor item has this attribute set to false.	0..*	
@showInReport	If @showInReport on a Question is false, then the Question and its ListItems (if any) should not be present in a typical report derived from this template. If this attribute appears on a ListItem, then the entire question and all ListItems should be omitted from the report. If this attribute appears on a Section, then the entire Section and all Section contents (descendants) should be omitted from the report. If @showInReport appears on a DisplayedItem, then the DisplayedItem (and all sub-items) should be omitted from the report. This attribute is usually set to false when a Question is used to control form behavior (e.g., skip logic), or when the Question provides unhelpful "negative" information about actions that did not occur or were not performed, or things that were not observed.	0..*	

Q.6.3 Supplemental Material

The following supplemental material will help with both the understanding of the SDC form definition model and implementation. These supplemental materials are considered “non-normative” and are available on the IHE FTP site:

ftp://ftp.ihe.net/TF_Implementation_Material/QRPH/SDC/.

- SDC Schema files and Documentation Package: This package contains schema files in the “SDC Schema v2.0” folder. It also contain helpful documentation for the three main schema files – SDCRetrieveForm.xsd, SDCFormDesign.xsd, and SDCSubmitForm.xsd under “SDC Schema v2.0 Documentation” folder. To ensure that the HTML files have correct links, simply unzip them maintaining the folder structure.
- SDC Form Design Sample Files Package: This package contains four different form design sample files. These are XML form design files that are fully compliant to the SDC Form Design schema.
- SDC XML-to-HTML Transform Package: In order to support HTML and URI options, the XML form definition needs to be transformed to an SDC-compliant HTML file. This package contains script files and relevant stylesheet (.css files) to convert any given SDC XML form definition to an SDC compliant HTML form instance. It also contains javascript that will generate the submission package compliant to SDC Submit Form schema. This is just an example instance showing an approach to transform XML form definition to an HTML form instance and is not meant to be a prescriptive or a mandatory method.

Volume 3 Appendices

1030 **Appendix A – List of SDC Form Elements and Attributes Adapted from ISO/IEC 19763-13 and ISO/IEC 11179-3, © copyright ISO/IEC 2014**

A.1 Base Elements from ISO/IEC Standards

A.1.1 Organization

1035 Organization is a class, each instance of which models an organization, which is a unique framework of authority within which individuals act, or are designated to act, towards some purpose. For additional details, reference: ISO/IEC 11179-3.

Table A.1.1-1: Organization Class

Element Name	Element Definition	Card	Data Type
Organization			
+OrgName	Name of the organization	0..1	string
+Department	Name of the Department	0..1	string
+StreetAddress	Postal address for the organization	0..*	Table A.1.3-1
+Email	Email address for the organization	0..*	Table A.1.3-1
+Phone	Phone number for the organization	0..*	Table A.1.3-1
+WebURL	Uri for the organization	0..*	Table A.1.3-1

A.1.2 Person

1040 A Person is defined as a single human being.

Table A.1.2-1: Person

Element Name	Element Description	Card	Data Type
Person			
+NamePrefix	Sign that designates the individual	0..1	string
+FirstName	First or Given name of the individual	1..1	string
+MiddleName	Middle name or initial of the individual	0..1	string
+LastName	Last name or surname of the individual	1..1	string
+NameSuffix	Suffix	0..1	string
+DegreeCert	Acronyms for degrees and certifications, e.g., MD, PhD, FACP (repeat element for each acronym)	0..*	string

Element Name	Element Description	Card	Data Type
+Title	Name of the position held by the individual	0..1	string
+StreetAddress	Postal address for the individual	0..*	Table A.1.3-1
+Email	Email address for the individual	0..*	Table A.1.3-1
+Phone	Phone number for the individual	0..*	Table A.1.3-1
+WebURL	Uri for the individual	0..*	Table A.1.3-1

A.1.3 Contact Information

1045 A set of elements defining contact information for the individual or an organization. It includes street/postal address, email address, phone number, email information, and other relevant elements as described below.

Table A.1.3-1: Contact Information

Element Name	Element Description	Card	Data Type
StreetAddress			
+AddressType	Type of address, e.g., home, office, etc.	0..1	string
+InternalAddress	Address instructions for directing mail within an organizations buildings or corporate infrastructure	0..10	string
+AddressLine	Address instructions for directing mail to a street address	0..10	string
+City	Name of the City	0..1	string
+Jurisdiction	Other optional locale identifier	0..1	string
+State	State, or Province information	0..1	string
+PostalCode	Local postal code, e.g., zip code	0..1	string
+Country	Name of the country	0..1	string
Email	A structure for recording email addresses	0..*	
+EmailAddress	Email Address	1..1	string
+EmailClass	Type of email, e.g., home, office, etc.	0..1	string
Phone	A structure for recording telephone numbers	0..*	
+PhoneType	Usage type of phone number, e.g., home, office, etc.	0..1	string
+CountryCode	Country code	0..1	string
+AreaCode	Area Code	0..1	string
+PhoneNumber	Phone Number	1..1	string
+PhoneExtension	Phone number extension	0..1	string
WebURL	The web URL associated with the individual or an organization	0..*	anyURI

Element Name	Element Description	Card	Data Type
+Usage	When the relevant item (e.g., email, phone number, address, etc.) should be used	0..1	string

A.1.4 FileType

1050 Information about a file usually thought of as a binary byte stream stored on disk. A FileType can also represent a "virtual" file, such as an XML module in a larger XML document. Such a virtual file could be stored as a byte stream, as an independent file on disk or as a database record or set of records, even if this byte stream is never actually persisted as an independent disk file.

Table A.1.4-1: FileType

Element Name	Element Description	Card	Data Type
FileID	Internal/local File ID, not necessarily in the format of the FileURI used for all SDC FormDesign items	0..1	string
FileURI	Unique ID for the file	1..1	string
DisplayName	Text to be displayed representing the file contents. This may be the same as the internal title of the file	0..1	string
Title	Official title of the file	0..1	string
Version	Version number for the file	0..1	string
FileName	The name of the file as saved on disk or other persistent storage	0..1	string
FileExtension	The file type extension that describes the files internal format. This is usually the 3-4 character text that appears after the last period in the file name, e.g., txt, docx, etc.	0..1	string
FileClass	A short description of the class of file, such as "FormDesign XML"	0..1	string
FileSizeKB	Byte size of the file	0..1	positiveInteger
Description	Any additional information about the template or file. The type of information should be specified in the @type attribute.	0..*	string
Copyright	Copyright information	0..1	string
TermsOfUse	Terms of Use description	0..1	string
Usage	Guidance for when this file should be used, and when it should not be used	0..1	string
Dates	Various dates associated with the file release, versioning and usage	0..1	DateTime
Contacts	People and Organizations associated with the file. Specify the type of Contact in the @type attribute. Examples of @type include Curator, Author, and Authority	0..1	Table A.1.3-1

Element Name	Element Description	Card	Data Type
Approval	Documentation of review and acceptance of the file for production usage	0..*	string
FileHash	A hash of file data keyed by the file's name	0..1	string
Language	A generic structure for recording language used for text in the form. The codes are ISO 639-3 codes	0..1	string
AssociatedFiles	Link to any associated files, such as schemas, reference documents, manuals, etc.	0..*	FileType
Provenance	Provenance information	0..1	string
DefaultSubmissionRule	Information about where to submit a completed form	0..*	string
DefaultComplianceRule	General information about how information in a particular form package must be handled. This may include, e.g., security, completeness, transmission, and validation criteria	0..*	string

1055

A.1.5 BlobContent

Standard structure for including Binary Large Objects (Blobs) in XML templates. Blobs can handle any type of binary media, such as images, audio, video, and data streams of any specified format (such as a MIME type).

1060

Table A.1.5-1: BlobContent

Element Name	Element Description	Card	Data Type
BlobContent			
+Description	Description of the Blob	0..1	string
+Hash	Binary hash of the blob data	0..1	string
+BlobURI	This URI should contain a URL or web service link to the Blob. It is used instead of incorporating the Blob directly into XML templates	1..1	string
+BinaryMediaBase64	Binary representation of the Blob data, encoded using a Base 64 transformation from the original source binary data. Base 64 encoding uses 64 character alphanumeric text (characters A–Z, a–z, 0–9, =, /) to represent the binary data, so that it may be transmitted as "text" inside an XML instance document without the need for escape characters	1..1	string

A.1.6 Datatype

A datatype is a set of distinct values, characterized by properties of those values and by operations on those values. The datatypes are based on W3C types. The following table shows the list of datatypes:

1065

Table A.1.6-1: Datatypes

anyType
anyURI
base64Binary
boolean
duration
ID
int
integer
byte
date
dateTime
dateTimeStamp
decimal
double
float
gDay
gMonth
gMonthDay
gYear
gYearMonth
hexBinary
HTML
long
negativeInteger
NMTOKENS
nonNegativeInteger
nonPositiveInteger
positiveInteger
short
string
time
unsignedByte
unsignedInt
unsignedLong
unsignedShort
XML
yearMonthDuration

Volume 3 Namespace Additions

<i>Add the following terms to the IHE Namespace:</i>
--

None

1070