# Part II: User Guide

## Table of Contents

Documentation of the DDI specification is provided in three ways:

## Field Level Documentation

AUDIENCE: Developers, database developers, mappings, base level for content providers (what an object is in relation to parent and child elements)

This documentation is found within the DDI Schemas and displayed in the HTML documentation. It provides a brief description of the purpose and content of the object. Documentation found in the complex Type description will provide more detail than the element documentation. Within a complex type, the additional documentation of sub-elements will focus on its purpose within the context of the complex type.

# Part I - Technical Documentation

AUDIENCE: Developers, integrated usage and applications for content providers

Organized by related sets of objects, e.g. Question Item, Question Grid, and Question Block, this documentation provides details of the structure and its intended application. Each set contains examples of usage. It contains information on the relationship of DDI to other standards, common XML structures used by DDI, design and consistency rules, description of major structural types (modules and schemes), technical features for identification and reference, basic types for dates and strings, and all major complex elements. The complex element content is organized alphabetically by set and an index is provided for all elements. This documentation also contains lists of: 3.1 to 3.2 changes, all unique element and attribute names, and elements by extension base (Identifiable, Versionable, Maintainable, Reference, CodeValue, etc.).

## Part II - User Guide

AUDIENCE: Content providers, those focusing on specific applied uses of DDI

Provides instructions for navigating the HTML Field Level Documentation and reviews basic structural features focusing on their usage, such as exchange structures, organizing publication package content, managing data over time, common structure like strings, controlled vocabularies, dates, citation and coverage, notes and other material. This general section is followed by a set of user stories (applying DDI). The focus is on how the parts of DDI work together to describe the metadata and data for particular functions such as documenting a longitudinal study or developing a questionnaire. Wherever appropriate, Part I will reference the more detailed technical documentation in Part I.

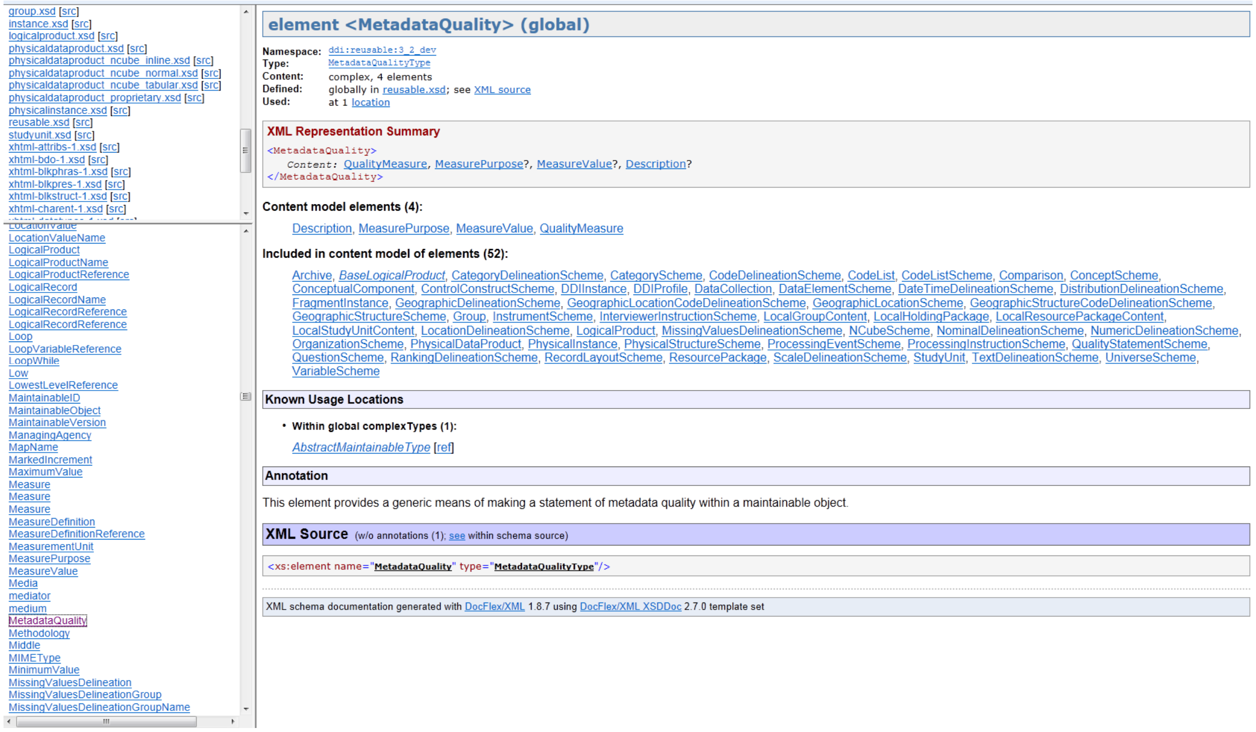
## Overview

The Part II: User Guide has been provided to help the user navigate through the DDI Lifecycle content and get a sense of its overall structure. However the primary focus is on the sets of metadata required for specific applications. References to related documentation in Part I: Technical Structures are provided throughout the User Guide as inline references indicating Part I and the section number using this format [pt1:2.1.2]. The User Guide is organized into three main sections: Navigating the HTML Field Level Documentation, DDI Structural Features, and User Stories – Applying DDI. Note that the user can start at any point in the User Guide as the User Stories will provide references to content in the DDI Structural Features when appropriate using the section number and title, for example [2.3 Organizing Publication Package Contents].

## 1 - Navigating the HTML Field Level Documentation

The HTML documentation is generated from the content of the DDI schema set. The upper left frame contains primary navigation and opens on the Overview which lists the 22 Namespaces and 42 XML Schemas that comprise DDI Lifecycle. The lower left frame lists All Components (1181 Elements, 473 Complex Elements, 68 Simple Types, 71 Element Groups, 7 Attributes, and 70 Attribute Groups). It seems like a lot but over half the XML Schemas, all the Element Groups, Attributes, and Attribute Groups are there to support the use of XHTML or Dublin Core. The lower left frame can be focused on a single XML Schema content by clicking on the name of that XML Schema in the upper left Frame.

Field level documentation provides information on what the field contains and if it is a complex element how the elements and attributes within it are used. If you need to know what a specific element contains you can find it in the alphabetical element list in the lower left frame and click on it, for example, MetadataQuality.



group.xsd [src]
instance.xsd [src]
logicalproduct.xsd [src]
physicaldataproduct.xsd [src]
physicaldataproduct_ncube_inline.xsd [src]
physicaldataproduct_ncube_normal.xsd [src]
physicaldataproduct_ncube_tabular.xsd [src]
physicaldataproduct_proprietary.xsd [src]
physicalinstance.xsd [src]
reusable.xsd [src]
studyunit.xsd [src]
xhtml-attribs-1.xsd [src]
xhtml-bdo-1.xsd [src]
xhtml-blkphras-1.xsd [src]
xhtml-blkpres-1.xsd [src]
xhtml-blkstruct-1.xsd [src]
xhtml-charent-1.xsd [src]

LocationValue
LocationValueName
LogicalProduct
LogicalProductName
LogicalProductReference
LogicalRecord
LogicalRecordName
LogicalRecordReference
LogicalRecordReference
Loop
LoopVariableReference
LoopWhile
Low
LowestLevelReference
MaintainableID
MaintainableObject
MaintainableVersion
ManagingAgency
MapName
MarkedIncrement
MaximumValue
Measure
Measure
Measure
MeasureDefinition
MeasureDefinitionReference
MeasurementUnit
MeasurePurpose
MeasureValue
Media
mediator
medium
MetadataQuality
Methodology
Middle
MIMEType
MinimumValue
MissingValuesDelineation
MissingValuesDelineationGroup
MissingValuesDelineationGroupName

## element <MetadataQuality> (global)

**Namespace:** ddi:reusable:3_2_dev
**Type:** MetadataQualityType
**Content:** complex, 4 elements
**Defined:** globally in reusable.xsd; see XML source
**Used:** at 1 location

**XML Representation Summary**

```
<MetadataQuality>
   Content: QualityMeasure, MeasurePurpose?, MeasureValue?, Description?
</MetadataQuality>
```

**Content model elements (4):**

Description, MeasurePurpose, MeasureValue, QualityMeasure

**Included in content model of elements (52):**

Archive, *BaseLogicalProduct*, CategoryDelineationScheme, CategoryScheme, CodeDelineationScheme, CodeList, CodeListScheme, Comparison, ConceptScheme, ConceptualComponent, ControlConstructScheme, DDIInstance, DDIProfile, DataCollection, DataElementScheme, DateTimeDelineationScheme, DistributionDelineationScheme, FragmentInstance, GeographicDelineationScheme, GeographicLocationCodeDelineationScheme, GeographicLocationScheme, GeographicStructureCodeDelineationScheme, GeographicStructureScheme, Group, InstrumentScheme, InterviewerInstructionScheme, LocalGroupContent, LocalHoldingPackage, LocalResourcePackageContent, LocalStudyUnitContent, LocationDelineationScheme, LogicalProduct, MissingValuesDelineationScheme, NCubeScheme, NominalDelineationScheme, NumericDelineationScheme, OrganizationScheme, PhysicalDataProduct, PhysicalInstance, PhysicalStructureScheme, ProcessingEventScheme, ProcessingInstructionScheme, QualityStatementScheme, QuestionScheme, RankingDelineationScheme, RecordLayoutScheme, ResourcePackage, ScaleDelineationScheme, StudyUnit, TextDelineationScheme, UniverseScheme, VariableScheme

**Known Usage Locations**

- **Within global complexTypes (1):**

  *AbstractMaintainableType* [ref]

**Annotation**

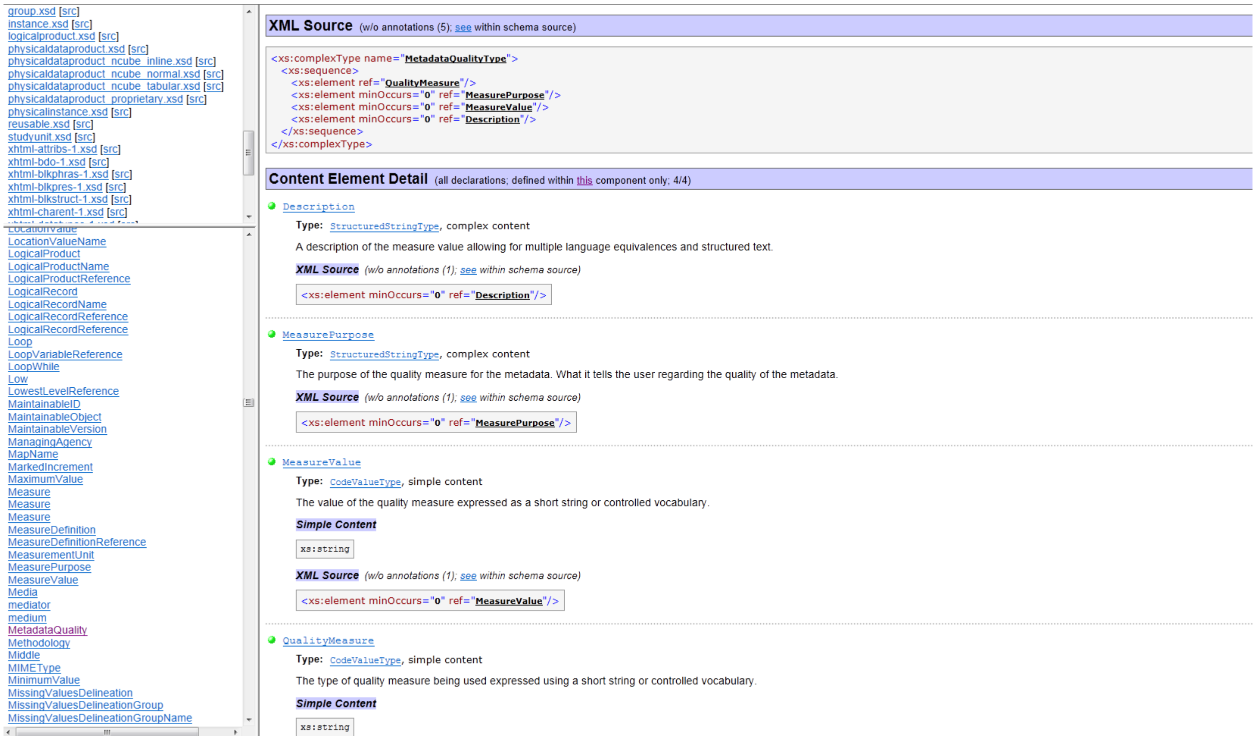
This element provides a generic means of making a statement of metadata quality within a maintainable object.

**XML Source** (w/o annotations (1); see within schema source)

```
<xs:element name="MetadataQuality" type="MetadataQualityType"/>
```

XML schema documentation generated with DocFlex/XML 1.8.7 using DocFlex/XML XSDDoc 2.7.0 template set

The right frame now provides information on the element MetadataQuality including what namespace it is in, the type, contents, where it is defined, and where it is used. This is followed by the XML Representation Summary (a listing of each element and its cardinality expressed as required/not repeatable [ ], required/repeatable [+], optional/not repeatable [?], or optional/repeatable [*]). The elements within the content model are listed in alphabetical order followed by the list of elements where MetadataQuality is available, known usage locations, documentation (annotation), and the XML Source. All of these contain clickable links for additional details. For the full description of the structure click on the “Type:” in the top section. This will provide content and documentation details for the structure. To understand the use of the object in a specific location click on the name of the object that includes it.



group.xsd [src]
instance.xsd [src]
logicalproduct.xsd [src]
physicaldataproduct.xsd [src]
physicaldataproduct_ncube_inline.xsd [src]
physicaldataproduct_ncube_normal.xsd [src]
physicaldataproduct_ncube_tabular.xsd [src]
physicaldataproduct_proprietary.xsd [src]
physicalinstance.xsd [src]
reusable.xsd [src]
studyunit.xsd [src]
xhtml-attribs-1.xsd [src]
xhtml-bdo-1.xsd [src]
xhtml-blkphras-1.xsd [src]
xhtml-blkpres-1.xsd [src]
xhtml-blkstruct-1.xsd [src]
xhtml-charent-1.xsd [src]

LocationValue
LocationValueName
LogicalProduct
LogicalProductName
LogicalProductReference
LogicalRecord
LogicalRecordName
LogicalRecordReference
LogicalRecordReference
Loop
LoopVariableReference
LoopWhile
Low
LowestLevelReference
MaintainableID
MaintainableObject
MaintainableVersion
ManagingAgency
MapName
MarkedIncrement
MaximumValue
Measure
Measure
Measure
MeasureDefinition
MeasureDefinitionReference
MeasurementUnit
MeasurePurpose
MeasureValue
Media
mediator
medium
MetadataQuality
Methodology
Middle
MIMEType
MinimumValue
MissingValuesDelineation
MissingValuesDelineationGroup
MissingValuesDelineationGroupName

**XML Source** (w/o annotations (5); see within schema source)

```
<xs:complexType name="MetadataQualityType">
  <xs:sequence>
    <xs:element ref="QualityMeasure"/>
    <xs:element minOccurs="0" ref="MeasurePurpose"/>
    <xs:element minOccurs="0" ref="MeasureValue"/>
    <xs:element minOccurs="0" ref="Description"/>
  </xs:sequence>
</xs:complexType>
```

**Content Element Detail** (all declarations; defined within this component only; 4/4)

- Description

**Type:** StructuredStringType, complex content

A description of the measure value allowing for multiple language equivalences and structured text.

***XML Source*** *(w/o annotations (1); see within schema source)*

```
<xs:element minOccurs="0" ref="Description"/>
```

- MeasurePurpose

**Type:** StructuredStringType, complex content

The purpose of the quality measure for the metadata. What it tells the user regarding the quality of the metadata.

***XML Source*** *(w/o annotations (1); see within schema source)*

```
<xs:element minOccurs="0" ref="MeasurePurpose"/>
```

- MeasureValue

**Type:** CodeValueType, simple content

The value of the quality measure expressed as a short string or controlled vocabulary.

***Simple Content***

`xs:string`

***XML Source*** *(w/o annotations (1); see within schema source)*

```
<xs:element minOccurs="0" ref="MeasureValue"/>
```

- QualityMeasure

**Type:** CodeValueType, simple content

The type of quality measure being used expressed using a short string or controlled vocabulary.

***Simple Content***

`xs:string`

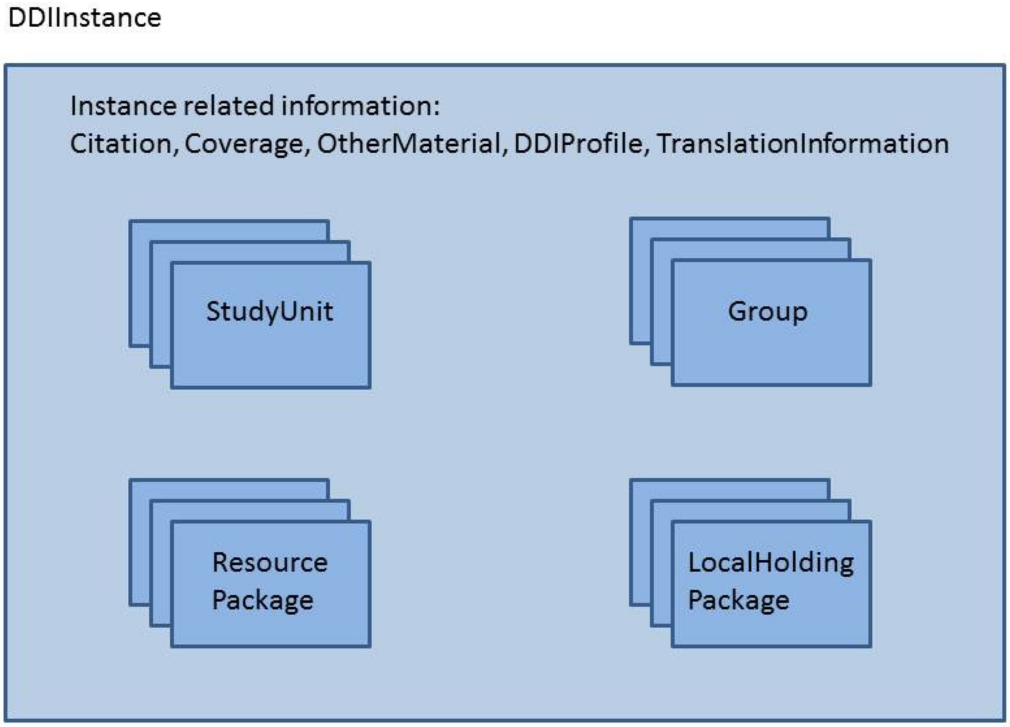
Now in addition to the information on first screen you have more extensive documentation for the complex element, a more detailed XML Source showing order and cardinality as XML schema, and content element detail including type information and documentation of what each element is intended to contain.

If you are starting from scratch and just want to explore start at the element DDIInstance which is the top level publishing structure in DDI then work your way down through the structures of the primary document types. The most commonly used DDI lifecycle document is StudyUnit and it corresponds most closely to the content coverage of DDI Codebook. See DDI Structural Features for basic content organization [2.1 Exchange structures, and 2.2 Maintainable structures].

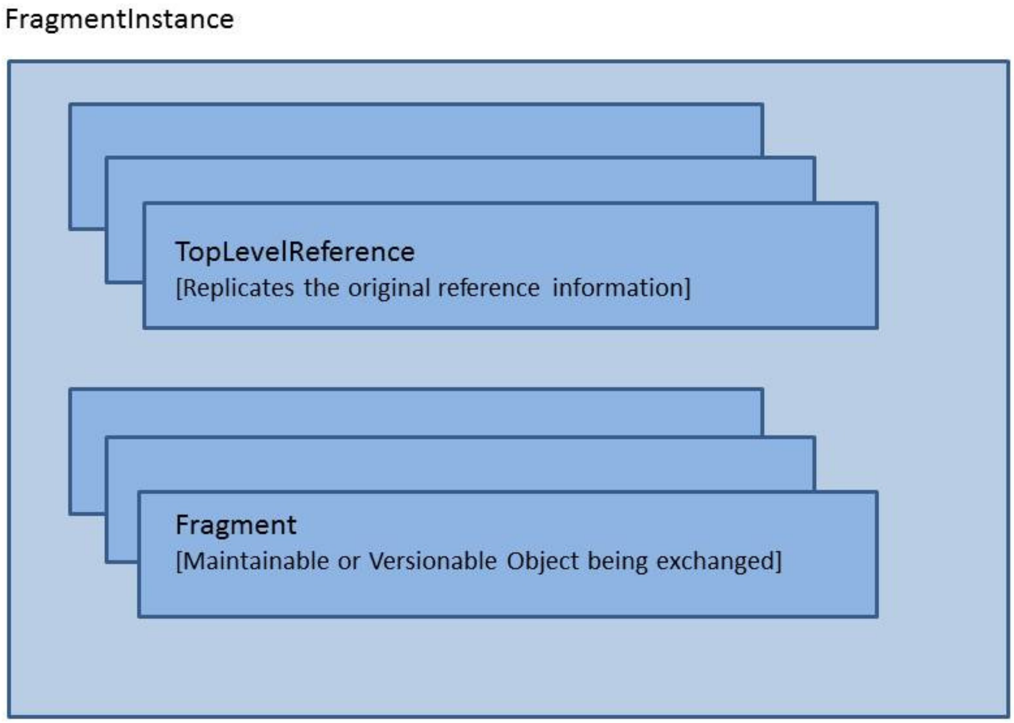
## 2 - DDI Structural Features

### 2.1 - Exchange structures

DDI Lifecycle instances are published and exchanged with one of two external wrappers, DDIInstance or FragmentInstance. The DDIInstance provides a consistent top-level publication wrapper and in addition to some basic information about itself serves as the publication wrapper for four primary document types: StudyUnit, Group, ResourcePackage, and LocalHoldingPackage.  The FragmentInstance is a uniform package used to transfer maintainable or versionable objects plus any associated Notes or OtherMaterial. These would be packets sent in response to system calls (external references, query calls, etc.).


DDIInstance
Instance related information:
Citation, Coverage, OtherMaterial, DDIProfile, TranslationInformation
StudyUnit
Group
Resource Package
LocalHolding Package

The Citation, Coverage, OtherMaterial, DDIProfile, and TranslationInformation pertain to the DDI Instance as a whole. Note that the DDIInstance may be viewed as a temporary wrapper for publishing or transporting any of the major publication structures within DDI.


FragmentInstance
TopLevelReference
[Replicates the original reference information]
Fragment
[Maintainable or Versionable Object being exchanged]

The FragmentInstance is a wrapper for transporting a response to a request for a specific set of information. Although Maintainable, Versionable, and Identifiable objects may be referenced, the FragmentInstance can only transport a Maintainable or Versionable object. If an Identifiable is the referenced object, its parent Versionable (or Maintainable) will be supplied. A FragmentInstance may contain any number of Versionable or Maintainable ojbects. The objects that are the specific responses to the request are listed in the TopLevelReference. For example, a single CodeList (Versionable object) may be requested and the FragmentInstance sent in response may contain the CodeList, the containing CodeListScheme, the referenced Categories, and the containing CategoryScheme. The requested CodeList would be noted in the TopLevelReference and the content of the returned objects would make up the Fragments.

## 2.2 - Maintainable structures

DDI has two types of maintainable structures; Modules and Schemes. Modules are conceptual y related groups of metadata related to stages within a lifecycle. Schemes are maintainable lists of reusable objects of specified generic types (i.e., questions) and usually include a means of expressing groups of these objects for administrative purposes. The following is a list of Modules and Schemes available in DDI Lifecycle along with the namespace of the object, the object name, and a description.

| Modules | | |
|---|---|---|
| a | Archive | Contains information concerning the organization providing archival functions, the position of the related metadata and data within the organization, and preservation/provenance information about the metadata and data including LifeCycleEvents. |
| l | CodeList | A special form of maintainable that allows a single codelist to be maintained outside of a CodeListScheme. |
| cm | Comparison | Contains information on comparison of similar metadata objects using mapping between a source and target object. |
| c | ConceptualComponent | Contains descriptions of Concepts, Universes, DataElements, GeographicStructures, and GeographicLocations. |
| d | DataCollection | Contains information on data collection, capture, methodology, and processing of data. |
| pr | DDIProfile | A specialized meta-model structure that specifies the elements in DDI used by an application, organization, or project and how they are used. |
| g | Group | A publication module that pulls together multiple StudyUnits with either an intended relationship (i.e., longitudinal study) or an ad-hoc relationship (i.e., studies on aging used within an instructional package). |
| g | LocalHoldingPackage | A publication structure that allows an archive or library to bind locally produced metadata to deposited metadata without altering the original metadata set. |
| g | LocalGroupContent | Locally produced Group content within a LocalHoldingPackage. |
| g | LocalResourcePackageContent | Locally produced ResourcePackage content within a LocalHoldingPackage. |
| g | LocalStudyUnitContent | Locally produced StudyUnit content within a LocalHoldingPackage. |
| l | LogicalProduct | Contains information on the intellectual structure of the data (i.e., Variables, NCubes), including CategorySchemes, CodeListSchemes, and information on how the data are organized into LogicalRecords and the Relationship of those records to each other. |

| | | |
|---|---|---|
| p | PhysicalDataProduct | Contains information on the physical structure of the data including file structures and RecordLayout structures. Links to the LogicalRecord. |
| pi | PhysicalInstance | A metadata record for a data file providing identification information for the data file, a link to the RecordLayouts found in the data file, and summary statistics for the data file. |
| g | ResourcePackage | A publication structure that allows any maintainable object that is not a publication package to be published as a reusable resource outside of the context of a specific study. |
| s | StudyUnit | A publication structure for a specific study. Structures identification information, full bibliographic and discovery information, administrative information, all of the reusable delineations used for response domains and variable representations, and modules covering different points in the lifecycle of the study (DataCollection, LogicalProduct, PhysicalDataProduct, PhysicalInstance, Archive, and DDIProfile). |
| **Schemes** | | |
| l | CategoryScheme | |
| l | CodeListScheme | |
| c | ConceptScheme | |
| c | ConceptualVariableScheme | |
| d | ControlConstructScheme | |
| c | GeographicLocationScheme | |
| c | GeographicStructureScheme | |
| d | InstrumentScheme | |
| d | InterviewerInstructionScheme | |
| l | NCubeScheme | |
| r | ManagedRepresentationScheme | |
| a | OrganizationScheme | |
| p | PhysicalStructureScheme | |
| d | ProcessingEventScheme | |
| d | ProcessingInstructionScheme | |
| r | QualityStatementScheme | |
| d | QuestionScheme | |
| p | RecordLayoutScheme | |
| l | RepresentedVariableScheme | |
| c | UniverseScheme | |
| l | VariableScheme | |

## 2.3 - Organizing Publication Package Contents

The major publication packages (StudyUnit, Group, ResourcePackage, and LocalHoldingPackage) organize their contents in a set order or sequence. Although all objects may not be available in each publication structure, the order of all the included non-maintainable objects remains the same. The maintainable objects in Group and StudyUnit follow the same order. In ResourcePackage all maintainable modules fall before the DDI Schemes within the content sequence. Note that within the list of maintainable Modules and maintainable DDI Schemes, the ordering is consistent with Group and StudyUnit. The table in Appendix A lists the content and order of the three primary publication packages. LocalHoldingPackage is a specialized structure that bundles together a publication package from an external agency (StudyUnit, Group, or ResourcePackage) with similarly structure locally added materials. Further information on LocalHoldingPackage structure and usage is found in 3.10 DDI and OAIS – Archives and provenance.

All maintainable objects published within StudyUnit and Group and all non-DDI scheme maintainable objects in ResourcePackage may be included in-line or by reference. DDI Schemes that are included in a ResourcePackage as separate items (i.e., not included within another Module) must be in-line. ResourcePackage is intended as a means of publishing metadata intended for reuse outside of a single study therefore it is the primary publishing structure for DDI Schemes with content that is used by multiple studies.

How an organization decides to structure its publication packages depends on how they intend to organize, manage, and reuse their metadata. Some organizations publish all their potentially reusable metadata as ResourcePackages with in-line content. StudyUnits and Groups are composed as a set of object specific metadata (Citation through Embargo content) followed by a stack of references. Others prefer to manage all metadata that is not specifically reused in-line within the context of the StudyUnit or Group. There are advantages and disadvantages to both approaches depending upon User Story in question. Both approaches will be discussed within the context of each User Story [3 User Stories – Applying DDI]. If an organization uses one extreme or the other for all or a class of metadata, this should be noted within the organization’s DDIProfile. For example, noting that the in-line option is not used for specific objects where there is a choice.

## 2.4 - Managing Metadata Over Time

Organizing for management purposes

Versioning

DDI Scheme Groups

## 2.5 - Common Structures

There are a number of common structures used by many of the objects in DDI. These deal either with content like strings, dates, and controlled vocabularies, or with common complex structures like Citation, Coverage, Notes, and OtherMaterial. A basic understanding of these common structures allows you to focus on the content coverage and arrangement rather than the fine details. Note that Identification and Reference structures are covered in Part I [pt1:x.x.x].

### 2.5.1 - String, Controlled Vocabularies

All DDI string content is based on an extension of xs:string and is designed to support the use of multiple language content for a given element where appropriate, structured text content, and for questionnaire related materials, dynamic text. In addition, DDI supports the use of external controlled vocabularies through the structure CodeValue which identifies the source and location of the external controlled vocabulary as well as the term content.

The basic structure is an xs:string which allows for any character in any sequence. Note that XML ignores leading and trailing white spaces as well as control characters like tabs and hard returns. In short it will ignore internal structuring of content. DDI has created the following xs:string extensions to provide support for content structure and language specification where needed.

In some cases, such as the value of a code, leading and trailing spaces are important to both understanding and matching the content. Elements of type=”ValueType” provide the attribute xml:space with which the user can declare that leading and trailing white spaces have implications for the meaning of the content. The default value of xml:space is “default”. This states that the leading and trailing spaces may be stripped off. By changing the value of xml:space to “preserve” the user specifies that leading and trailing spaces should be retained as they are critical to the understanding of the content.

All elements of type=”InternationalStringType” support the use of one or more strings with equivalent language content [pt1:x.x.x]. A common example of this occurs in all primary element names, i.e., VariableName. An InternationalStringType bundles together one or more language equivalents of the same content. This requires the use of a sub-element “String” which is repeated for each language provided. String contains attributes to designate the language of the content and basic translation information.

```
<l:VariableName>
      <r:String xml:lang=”en” isTranslated=”false”
isTranslatable=”true”>Household Relationship</r:String>
      <r:String xml:lang=”fr” isTranslated=”false”
isTranslatable=”true”>Relation des ménages</r:String>
      <r:String xml:lang=”es” isTranslated=”true” isTranslatable=”true”
translationSourceLanguage=”en” translationDate=”2012-12-03”>Relación
de Hogares</r:String>
</l:VariableName>
```

What this example states is that the contents of the three strings are language equivalents for the VariableName content. The English and French are both original language content. The Spanish content is a translation of the English done on 2012-12-03. All the content may be translated. Bundling language equivalents together within a single object clarifies which language strings contain the same meaning when an object is repeatable.

All elements of a StructuredStringType use the sub-element “Content”. Content supports the same language structures using the same attributes as an InternationalStringType. In addition Content may contain a limted set of XHTML structure tags to provide structure to the content. There is one addition attribute “isPlainText” has been added to clarify if the content is to be treated as plain text (no formatting structure). The default value for this attribute is “true”. If the content contains structure tags this attribute should be changed to “false”. Label and Description are two commonly used elements of this type. A full list of allowed XMTL tags and their usage is found in the appendixes [Appendix B – XHTML Tags Supported by DDI]. The following example is a Description using an unordered (i.e., bulleted) list. Note that, like InternationalStringType the sub-element Content can be repeated for language equivalents.

```
<r:Description>
      <r:Content xml:lang=”en” isTranslated=”false”
isTranslatable=”true” isPlainText=”false”>A single person may include
any of the following:
            <xhtml:list>
                  <xhtml:item>Never married</xhtml:item>
                  <xhtml:item>Widowed</xhtml:item>
                  <xhtml:item>Divorced</xhtml:item>
            </xhtml:list>
      </r:Content>
</r:Description>
```

It would be interpreted as:

A single person may include any of the following:

- Never married
- Widowed
- Divorced

Note that if isPlainText=”true” the same line would be interpreted as:

```
A single person may include any of the
following:<xhtml:list><xhtml:item>Never
married</xhtml:item><xhtml:item>Widowed</xhtml:item><xhtml:item>Divorc
ed</xhtml:item></xhtml:list>
```

### 2.5.2 - Dates

### 2.5.3 - Citation and Coverage

### 2.5.4 – Notes

The element Note is available within all Maintainable objects. A Note allows the user to provide information not covered by DDI. It is not intended to replace formal local extensions of the schema, but to support capturing run-time extensions, content that is held in anticipation of a bug correction, or a temporary work-around. The primary use of Note is to capture mid-process Notes or instructions which may be removed later during the processing of the metadata.

A Note is captured once within a Maintainable object and then references the objects that it is related to. A Note can be attached by reference to any object with an ID. The intent of a Note is to be easily removable (removal of the Note also removes all reference links between the Note and the related objects). If a Note is related to objects outside of the Maintainable within which it exists, the Note should be duplicated in the Maintainable object which contains the other related objects. By placing the Note in the parent Maintainable, the user is assured of having all notes related to an object by checking in the parent Maintainable.

When a Note contains information that will be transferred to future elements or attributes (new content of a sub-minor version correction or the development of formal extensions) the use of the ProprietaryInfo (key/value pair) or well-structured content within the NoteContent field is recommended. Examples of different types of Notes are provided in Part I [pt1:note] .

### 2.5.5 - OtherMaterial

# 3 - User Stories – Applying DDI

## 3.1 - Documenting a simple study – Researcher perspective

## 3.2 - Documenting a simple study – Archive perspective (legacy material)

## 3.3 - Documenting a longitudinal study

## 3.4 - Questionnaire development

## 3.5 - Quality control of reusable content

## 3.6 – Variable management

## 3.7 - Concept management

## 3.8 - Geography and universe management

## 3.9 - Physical data management

## 3.10 - DDI and OAIS – Archives and provenance

## 3.11 - DDI and GSIM – implementation of the conceptual model

## 3.12 - DDI and Data Discovery

## 3.13 - Metadata driven processing

## Appendix A: Sequence of ResourcePackage, Group, and StudyUnit

| **ResourcePackage** | **Group** | **StudyUnit** |
|---|---|---|
| Citation | Citation | Citation |
| Abstract | Abstract | Abstract |
| AuthorizationSource | AuthorizationSource | AuthorizationSource |
| UniverseReference | UniverseReference | UniverseReference |
| SeriesStatement | SeriesStatement | SeriesStatement |
| QualityStatementReference | QualityStatementReference | QualityStatementScheme (inline or reference) |
|  | ExPostEvaluation | ExPostEvaluation |
| FundingInformation | FundingInformation | FundingInformation |
| ProjectBudget | ProjectBudget | StudyBudget |
| Purpose | Purpose | Purpose |
| Coverage | Coverage | Coverage |
|  | AnalysisUnit | AnalysisUnit |
|  | KindOfData | KindOfData |
| OtherMaterial | OtherMaterial | OtherMaterial |
|  | RequiredResourcePackages | RequiredResourcePackages |
| Embargo | Embargo | Embargo |
| *MAINTIANABLE MODULES, CHOICE OF INLINE OR BY REFERENCE* | *MAINTIANABLE OBJECTS, CHOICE OF INLINE OR BY REFERENCE* | *MAINTIANABLE OBJECTS, CHOICE OF INLINE OR BY REFERENCE* |
| ConceptualComponent | GeographicLocationCodeDelineationScheme | GeographicLocationCodeDelineationScheme |
| DataCollection | GeographicStructureCodeDelineationScheme | GeographicStructureCodeDelineationScheme |
| LogicalProduct | TextDelineationScheme | TextDelineationScheme |
| PhysicalDataProduct | DateTimeDelineationScheme | DateTimeDelineationScheme |
| PhysicalInstance | NumericDelineationScheme | NumericDelineationScheme |

| | | |
|---|---|---|
| Archive | CodeDelineationScheme | CodeDelineationScheme |
| DDIProfile | CategoryDelineationScheme | CategoryDelineationScheme |
| Comparison | GeographicDelineationScheme | GeographicDelineationScheme |
| *MAINTIANABLE DDI SCHEMES, INLINE ONLY* | NominalDelineationScheme | NominalDelineationScheme |
| OrganizationScheme | ScaleDelineationScheme | ScaleDelineationScheme |
| ConceptScheme | LocationDelineationScheme | LocationDelineationScheme |
| UniverseScheme | RankingDelineationScheme | RankingDelineationScheme |
| DataElementScheme | DistributionDelineationScheme | DistributionDelineationScheme |
| GeographicStructureScheme | MissingValuesDelineationScheme | MissingValuesDelineationScheme |
| GeographicLocationScheme | ConceptualComponent | ConceptualComponent |
| InterviewerInstructionScheme | DataCollection | DataCollection |
| ControlConstructScheme | LogicalProduct | LogicalProduct |
| QuestionScheme | PhysicalDataProduct | PhysicalDataProduct |
| CategoryScheme | PhysicalInstance | PhysicalInstance |
| CodeListScheme | Archive | Archive |
| NCubeScheme | DDIProfile | DDIProfile |
| VariableScheme | Comparison | |
| PhysicalStructureScheme | StudyUnit | |
| RecordLayoutScheme | SubGroup | |
| QualityStatementScheme | | |
| InstrumentScheme | | |
| ProcessingEventScheme | | |
| ProcessingInstructionScheme | | |
| GeographicLocationCodeDelineationScheme | | |
| GeographicStructureCodeDelineationScheme | | |
| TextDelineationScheme | | |
| DateTimeDelineationScheme | | |
| NumericDelineationScheme | | |

| | | |
|---|---|---|
| CodeDelineationScheme | | |
| CategoryDelineationScheme | | |
| GeographicDelineationScheme | | |
| NominalDelineationScheme | | |
| ScaleDelineationScheme | | |
| LocationDelineationScheme | | |
| RankingDelineationScheme | | |
| DistributionDelineationScheme | | |
| MissingValuesDelineationScheme | | |

## Appendix B: XHTML Tags Support by DDI

The following table provides the tag names, descriptions, and legal content for all XHTML tags supported by DDI. A list of unsupported tags follows this list. Most tags support several types of core attributes covering classification, style, internationalization, and events. Refer to W3C for detailed information on attribute usage.

- Note that these elements exist in the xhtml namespace and must be prefixed with that namespace, e.g. <xhtml:p>
- Additional information about XHTML tags can be found at http://www.w3schools.com/tags/default.asp

| **BLOCK ELEMENTS** | | |
|---|---|---|
| address | contact information for the document owner or author | May contain Inline Elements or text |
| blockquote | block quotation, a long quotation set off in a block of text | Contains Block Elements |
| div | division - generic way to divide group contents | May contain Block Elements, Inline Elements or text |
| dl | definition list | Must contain at least one dt or dd element |
| h1 | heading level 1 | May contain Inline Elements or text |
| h2 | heading level 2 | May contain Inline Elements or text |
| h3 | heading level 3 | May contain Inline Elements or text |
| h4 | heading level 4 | May contain Inline Elements or text |
| h5 | heading level 5 | May contain Inline Elements or text |
| h6 | heading level 6 | May contain Inline Elements or text |
| hr | horizontal line - content separator | No content  generally expressed as <xhtml:hr/> |
| ol | ordered list | Must contain at least one li |
| p | paragraph | May contain Inline Elements or text |
| pre | preformatted text | May contain Inline Elements (except img, object, big, small, sub, and sub, at any depth) or text |

| | | |
|---|---|---|
| table | table | Contains: caption may appear as the first item and only once; optional col or colgroup; one or more of the following tags in order: thead (0..1 and only if tbody is used), tfoot (0..1 and only if tbody is used), tbody (1..n) OR tr (1..n) |
| ul | unordered list | Must contain one or more li |
| **INLINE ELEMENTS** | | |
| a | anchor which defines the hypertext link using an id attribute | May contain Inline Elements (except a at any depth) or text |
| abbr | abbreviation | May contain Inline Elements or text |
| acronym | acronym | May contain Inline Elements or text |
| b | bold | May contain Inline Elements or text |
| big | big text | May contain Inline Elements or text |
| br | line break | No content generally expressed as <xhtml:br/> |
| cite | citation | May contain Inline Elements or text |
| code | computer code text | May contain Inline Elements or text |
| dfn | definition term | May contain Inline Elements or text |
| em | emphasized text | May contain Inline Elements or text |
| i | italics | May contain Inline Elements or text |
| kbd | keyboard text | May contain Inline Elements or text |
| q | quotation, short in line | May contain Inline Elements or text |
| samp | sample computer code | May contain Inline Elements or text |
| small | small text | May contain Inline Elements or text |
| span | section in a document | May contain Inline Elements or text |
| strong | strong text | May contain Inline Elements or text |
| sub | subscripted text | May contain Inline Elements or text |
| sup | superscripted text | May contain Inline Elements or text |
| tt | teletype text or monospaced text style | May contain Inline Elements or text |

| | | |
|---|---|---|
| var | variable part of text - indicates instance of a computer code variable or program argument | May contain Inline Elements or text |
| **LIST ELEMENTS** | | |
| dd | list definition description | May contain Block Elements, Inline Elements or text |
| dt | definition (list) term | May contain Inline Elements or text |
| li | list item - ordered or unordered | May contain Block Elements, Inline Elements or text |
| **TABLE ELEMENTS** | | |
| caption | table caption | May contain Inline Elements or text |
| col | attribute values for one or more columns in a table | No content - provides attributes describing alignment, width, formating of cells |
| colgroup | group of columns in a table for formatting | May contain col |
| tbody | body content in a table | May contain tr |
| td | cell in a table | May contain Block Elements, Inline Elements or text |
| tfoot | footer content in a table | May contain tr |
| th | header cell in a table | May contain Block Elements, Inline Elements or text |
| thead | header content in a table | May contain tr |
| tr | row in a table | Must contain one or more of td OR th |

***Unsupported Tags:***

| Element Group | Tag | Description |
|---|---|---|
| Block | del | delete content |
| Block | fieldset | Form control group |
| Block | form | data entry form |

| | | |
|---|---|---|
| Block | nocript | alternate content for client-side script |
| Block | ins | inserted content |
| Block | script | client-side script |
| Inline | bdo | bidirectional text override |
| Inline | button | form button control |
| Inline | del | deleted content |
| Inline | ins | inserted text |
| Inline | img | image |
| Inline | input | form control |
| Inline | label | form control label |
| Inline | map | client-side image map |
| Inline | object | generic embedded object |
| Inline | ruby | pronunciation annotations for East Asian languages |
| Inline | script | client-side script |
| Inline | select | option selector from control |
| Inline | textarea | multi-line text field form control |