DDI-Codebook: DDI 2.5 Detailed Change Specification

DDI Version 2.5 -- 19 October 2011

GENERAL USAGE	
Citation	**IDNo @level, Series Statement, and Qualified Dublin Core**

```
Location:
<citation MARCURI=””>
  <titlStmt>
    <IDNo> @level (study | file | project)
  <rspStmt>
  <prodStmt>
  <distStmt>
  <serStmt>   0..n
  <verStmt>
  <biblCit>
  <holdings>
  <notes>
  <dc:elementsAndRefinementsGroup>
</citation>
```

IDNo attribute “level” has been extended to include the valid value of project.
Series statement (serStmt) has been made repeatable to allow for membership in multiple series.
Provides an option to all citations for the use of qualified Dublin Core element (dcterms).

Usage:
IDNo @level - Allows for the provision of a project number for project related to the study, for example the “Indoor Residual Spraying (IRS) Impact Evaluation Survey” (ERI_2009_IRSIES_v01_M) done for the Impact Evaluation project “Indoor Residual Spraying and Incentives for Improving Larval Habitat Management” (P111233)

serStmt – There have been cases of a study belonging to more than one series, especially in cases of multiple use studies such as when the ISSP is fielded as part of a national population survey.

Qualified Dublin Core - Addition o f this group allows a replication of citation elements in Dublin Core internally within all citation elements (elements of namespace dc and dcterms)

```
Example:
<citation>
  <titlStmt>
    <titl>Indoor Residual Spraying Impact Evaluation Survey</titl>
    <IDNo level=”study”>ERI_2009_IRSIES_v01_M</IDNo>
    <IDNo level=”project”>P111233</IDNo>
  <titlStmt>
</citation>
<citation>
  <titlStmt>
    <titl>Life status survey 2007</titl>
  </titlStmt>
  <serStmt>
    <serName>ISSP</serName>
  </serStmt>
  <serStmt>
    <serName>National Life Status 2007</serName>
  </serStmt>
  <dc:title>Life Status Survey</dc:title>
  <dcterms:temporal>2007</dcterms:temporal>
<citation>
```

@codeBookAgency	**Code Book Agency Identification**

Location: attribute within **codeBook**

A place to put a registered DDI agency identifier.

Usage: This attribute of type xs:NCNAME is optional. It provides a location to capture a DDI-Lifecycle official agency identification when transforming an instance between DDI-C and DDI-L. It may also be used if an agency wishes to provide an DDI-L registered agency name regardless of whether they intend to transform metadata between the two DDI development paths.

Example:

```
<codeBook codeBookAgency="us.icpsr">
```

controlledVocabUsed	**Controlled Vocabularies**

Location: Within **docDscr** between **docSrc** and **notes**

```
<docDscr>
  <citation>
  <quide>
  <docStatus>
  <docSrc>
  <controlledVocabUsed>             0..1
    <codeListID>                    0..1
    <codeListName>                  0..1
    <codeListAgencyName>            0..1
    <codeListVersionID>             0..1
    <codeListURN>                   0..1
    <usage>                         1..n
      CHOICE                        1..1
        <selector>
        <specificElements @refs @authorizedCodeValue/>
      ENDCHOICE
      <attribute>                   0..1
    </usage>
  </controlledVocabUsed>
  <notes>
</docDscr>
```

A structure has been added to DDI-Codebook to allow for the use of the controlled vocabulary structure used in DDI-Lifecycle. To ensure backward compatibility, controlled vocabularies are listed in the document description section (docDscr) following the citation. The fields are similar to those in DDI-Lifecycle except that one must note the element or attribute that uses the controlled vocabulary using an XPATH. Restricted elements with internal code values were opened up with the addition of "other" to the allowed codes and an attribute that allows for listing the other value.

Usage: In adding the ability to use controlled vocabularies within the DDI-C structure, an approach different that that used in DDI-L was needed in order to support the requirement of backward compatibility within the DDI-C development line. Many locations where a controlled vocabulary was desirable, were attributes whereas the DDI-L controlled vocabulary structure is an element. Therefore, all controlled vocabularies used in a DDI-C instance must be declared in the docDscr. The format for identifying the controlled vocabulary themselves matches that found in DDI-L. The complex element "usage" has been added to identify the elements or attributes which used the controlled and can provide a valid code value when specific content in the instance is NOT a valid value in the controlled vocabulary but the original content is retained for legacy reasons. In most cases the valid code value would be "Other" but there may be a situation where the instance uses an alternate spelling or phrase for a valid value.

In the following examples the first applies the use of the DDI controlled vocabulary ResponseUnit to all values of response unit (respUnit) within any variable (var). The second is a case where a specific instance of the content of prodstmt/software with the ID of SW_1 contains a value of "Maddie" which is NOT a valid value of the controlled vocabulary. In this case the documentation provides the equivilent valid value from the controlled vocabulary list. This is useful when you wish to retain content for a historical reason but still wish to impose the use of controlled vocabularies to your metadata.

Note that in locations where DDI-C supports an internal controlled vocuabulary an option for "Other" has been added as well as an attribute allowing the "Other" value to be stated. For example in "var" which has the attribute "aggrMeth". The value of "other" has been added to the list of valid values and the attribute "otherAggrMeth" (type=xs:"NCTOKEN)" added to the element "var". This structure occurs in:

catStat @type=[freq | percent | crosstab | other] @otherType
fileStrc @type=[rectangular | hierarchical | relational | nested | other] @otherType
measure @aggrMeth=[sum | average | count | mode | median | maximum | minimum | percent |
    other] @otherAggrMeth
nCubeGrp @type=[section | multipleResp | grid | display | repetition | subject | version | iteration |
    anlaysis | pragatic | record | file | randomized | other] @otherType
qstn @responseDomainType=[text | numeric | code | category | datetime | geographic | multiple |
    other] @otherResponseDomainType
sumStat @type=[mean | medn | mode |vald | invld | min | max | stdev | other] @otherType
var @aggrMeth=[sum | average | count | mode | median | maximum | minimum | percent | other]
    @otherAggrMeth
var @additivity=[stock | flow | non-additive | other] @otherAdditivity
var @representationType=[text | numeric | code | category | datetime | other]
    @otherRepresentationType

varFormat @schema=[SAS | SPSS | IBM | ANSI | ISO | XML-Data | other] @otherSchema
varFormat @category=[date | time | currency | other] @otherCategory
varGrp @type=[section | multipleResp | grid | display | repetition | subject | version | iteration |
anlaysis | pragatic | record | file | randomized | other] @otherType

```
Example:
<controlledVocabUsed>
  <codeListID>ResponseUnit</codeListID>
  <codeListAgencyName>DDI Alliance</codeListAgencyName>
  <codeListVersionID>1.0.0</codeListVersionID>
  <codeListURN>urn:ddi-cv:ResponseUnit:1.0.0</codeListURN>
  <usage>
  <selector>/codebook//dataDscr//var/respUnit</selector>
  </usage>
</controlledVocabUsed>
<controlledVocabUsed>
  <codeListID>SummaryStatisticType</codeListID>
  <codeListAgencyName>DDI Alliance</codeListAgencyName>
  <codeListVersionID>1.0.0</codeListVersionID>
  <codeListURN>urn:ddi-cv:SummaryStatisticType:1.0.0</codeListURN>
  <usage>
  <selector>/codebook/dataDscr/var/sumStat</selector>
  <attribute>@type</attribute>
  </usage>
</controlledVocabUsed>
<controlledVocabUsed>
  <codeListID>SoftwarePackage</codeListID>
  <codeListAgencyName>DDI Alliance</codeListAgencyName>
  <codeListVersionID>1.0.0</codeListVersionID>
  <codeListURN>urn:ddi-cv:SoftwarePackage:1.0.0</codeListURN>
  <usage>
    <specificElement refs=”SW_1” authorizedCodeValue=”Other”/>
  </usage>
</controlledVocabUsed>

Example of extended internal controlled vocabularies:
<var representationType=”other” otherRepresentationType=”scale”>
```

dateSimpleType	**date fields (new)**

```
Location:
[any element] @elementVersionDate
exPostEvaluation @completionDate
sampleFrame/validPeriod
sampleFrame/referencePeriod
standardName @date
studyAuthorization @date
```

New date fields enforce the use of the ISO structure allowing for YYYY, YYYY-MM, and YYYY-MM-DD plus time.

Usage: All newly added date fields require rather recommend the use of an ISO structured date. This format includes:

xs:dateTime	yyyy-mm-ddThh:mm:ss
xs:date	yyyy-mm-dd
xs:gYearMonth	yyyy-mm
xs:gYear	yyyy

Note that NESSTAR software enforces the use of the ISO structure on all date attributes.

```
Example:
<studyAuthorization date=”2000-03-10”>
<sampleFrame>
  <validPeriod event=”start”>1990-01-01</validPeriod>
  <validPeriod event=”end”>1990-12-31</validPeriod>
</sampleFrame>
```

ddiCodebookUrn	DDI Codebook URN

Location: an attribute available in any element

Allows all elements to have a DDI-Codebook URN.

Usage: This would be used to capture a DDI-Codebook structured URN. Currently there is no defined structure for this, however it is anticipated that one will be defined.

```
Example:
<element ddiCodebookUrn=”urnstructure”>
```

ddiLifecycleUrn	DDI Lifecycle URN

Location: an attribute available in any element

Allows all elements to capture an element URN from a DDI-Lifecycle element.

Usage: This allows for the capture of a DDI-Lifecycle structured URN. the URN should be structured according to the specification described in the relevent version of DDI-Lifecycle.

```
Example
<var
ddiLifecycleUrn=”urn:ddi:us.mpc:VariableScheme.VS_1.1.0.0:Variable.V1.
1.0.0”>
```

elementVersion	Element Version Number

Location: an attribute available in any element

Allows all elements to contain a version number.

Usage: Allows versioning at the element level in DDI-Codebook. There is no specification for the structure of the version number

```
Example:
```

```
<element elementVersion=”1.2”>
```

elementVersionDate	**Element Version Date**

```
Location: an attribute available in any element
```

Allows all elements to contain a version date.

Usage: Allows for the use of a version date to further define or manage versioning within DDI-Codebook. This attribute uses dateSimple (see above).

```
Example:
<element elementVersionDate=”2010-10-14T14:42:23”>
```

notes @parent, @sameNote	**Notes attributes related to DDI-L**

```
Location: Notes are located at a number of locations in DDI-Codebook.
No new locations have been added
<notes parent=”IDRefs” sameNote=”IDRef”>
```

All notes can now explicitly state their parent element (for example, a Variable Note can reference the ID of the variable it is nested under). If the same note is related to a number of elements (for example repeated for a number of DDI-Codebook variables or a single DDI-Lifecycle note referencing a number of variables), a single note can be treated as a “master note” and list the elements it is related to. This addresses the different structure of notes in DDI-Lifecycle if moved into DDI-Codebook and allows DDI-Codebook users who wish to prepare for a transformation to DDI-Lifecycle to capture this information in a single location.

Usage: DDI-C requires that notes related to more than one element must be repeated within each of those elements. DDI-L changed this relationship linking a single note to multiple elements. There are three use cases for the added attributes.

1. When evaluating DDI-C content for transformation, the evaluator may wish to provide the ID of the containing parent element explicitly. This may facilitate the use certain software used in the transformation process.
2. When evaluating DDI-C content for transformation to DDI-L, the multiple relationships can be captured in a single “master” notes field. The “master” note (chosen by the evaluator) adds the attribute “parent” of type IDRefs and lists the ID’s of all elements that the note applies to. All other instances of the same note add the attribute “sameNote” of type IDRef which lists the ID of the “master” note. This may require the addition of ID’s to the referenced elements.
3. When transforming DDI-L to DDI-C the use of the “parent” attribute can allow for a single entry of the note with multiple links. This usage may or may not be supported by software that processes a DDI-C instance.

```
Example:
Case 1:
<var ID=”V1”>
  <notes level=”var” parent=”V1”>content</notes>
</var>

Case 2:
```

```
<var ID=”V1”>
  <notes level=”var” parent=”V1 V2 V3”>content</notes>
</var>
<var ID=”V2”>
  <notes level=”var” sameNote=”V1”>content</notes>
</var>
<var ID=”V3”>
  <notes level=”var” sameNote=”V1”>content</notes>
</var>

Case 3:
<var ID=”V1”>
  <notes level=”var” parent=”V1 V2 V3”>content</notes>
</var>
<var ID=”V2”></var>
<var ID=”V3”></var>
```

xhtml:BlkNoForm.mix	**XHTML**

`Location: Anywhere PHRASE|FORM or %a.phrase|%e.form was allowed in earlier versions of DDI-C`

Permits the optional use of XHTML wherever formatted text is allowed. This enables structured text to transfer directly from DDI-Lifecycle or for the user to prepare currently structured text in DDI-Codebook to be restructured to a format compatible with DDI-Lifecycle. NOTE that structure types should not be MIXED. Use either the original formatting options OR XHTML but not both.

Usage: The limited XHTML options found in DDI-L are now available in DDI-C. This allows for the direct transfer of DDI-L content into DDI-C as well as preparation of current DDI-C material in the XHTML format for direct transfer to DDI-L. Note that a number of current DDI-C tools wrap all large text fields in CDATA tags. Consult with how individual tools will handle XHTML tags prior to use.

```
Example:
<var name=”MOMLOC”>
  <txt><xhtml:p><xhtml:b> MOMLOC</xhtml:b> is a constructed variable that indicates whether
or not the person's mother lived in the same household and, if so, gives the person number of the
mother (see <xhtml:b>PERNUM</xhtml:b>). <xhtml:b>MOMLOC</xhtml:b> makes it easy for
researchers to link the characteristics of children and their (probable) mothers.  </xhtml:p>
<xhtml:p> The method by which probable child-mother links are identified is described in
<xhtml:b>PARRULE.</xhtml:b></xhtml:p><xhtml:p> The general design of
<xhtml:b>MOMLOC</xhtml:b> and other constructed variables follows the methods developed for
IPUMS-USA "Family Interrelationships," but the details vary significantly.</xhtml:p><xhtml:p>
Note: MOMLOC identifies social relationships (such as stepmother and adopted mother) as well as
biological relationships. The variable <xhtml:b>STEPMOM</xhtml:b>is designed to identify some of
these social relationships.</xhtml:p></txt>
</var>
```

Study Description (stdyDscr)

abstract @contentType	**abstract NEW attribute contentType**

```
Location:
<stdyInfo>
  <studyBudget>
  <subject>
  <abstract> @contentType (abstract | purpose | mixed)
  <sumDscr>
  <qualityStatement>
  <notes>
  <exPostEvaluation>
</stdyInfo>
```

Provides a tag indicating that the content should be mapped to a DDI-Lifecycle Abstract or Purpose or mixed. Mixed content can be tagged internally using structure tags. This is used to facilitate the transformation between the two development branches of DDI.

Usage: When preparing DDI-C content for tranformation to DDI-L this attribute allows you to assign the content to DDI-L Abstract or Purpose. If the content is mixed you could structure the content in such a way as to clarify what portion went to each DDI-L element. In addition, it allows content transformed from DDI-L to identify its source either by an explicit value of the attribute or by internal structure if “mixed”. The example below shows an instance of single content and of mixed content.

```
Example:
<abstract contentType=”abstract”>...</abstract>

<abstract contentType=”mixed”><xhtml:div><xhtml:h1>Abstract</xhtml:h1>
<xhtml:p>abstract content</xhtml:p></xhtml:div>
<xhtml:div><xhtml:h1>Purpose</xhtml:h1><xhtml:p>purpose
content</xhtml:p></xhtml:div></abstract>
```

codingInstructions	**Element/attribute name**

```
Location:
<method>
  <dataColl>
  <notes>
  <anlyInfo>
  <stdyClas>
  <dataProcessing>
  <codingInstructions @type @relatedProcesses>     0..n
    <txt>                                          0..n
    <command @formalLanguage>                      0..n
  </codingInstructions>
</method>
```

Provides for a detailed set of information on coding instructions including the type, any related processes, a description, and formal command language.

Usage: Coding instructions was added to further define the process of coding taking place at any level of data processing. The use of the type attribute (with or without an associated controlled vocabulary) allows the user to more completely identify the type process such as data processing, cleaning, assession, or tabulation. The coding instruction can be related to one or more process descriptions (i.e., cleanOps) through the use of the attribute relatedProcesses. This attribute is of type xs:IDREFS so that multiple processes can be referenced. The complex element codingInsturction contains a standard "txt" element in which to describe the instructions in a human readable format. The element "command" states the language of the command using the attribute "formalLanguage" and then contains the command itself in-line.

The following example shows a case of the command for cleaning a data file where the selected value is recoded to indicate invalid data. It refers to a cleaning operation with ID="CO_1"

```
Example:
<codingInstructions type="cleaning" relatedProcesses="CO_1">
  <txt>Due to an error in the data collection system the value of "27"
was entered for the variable NBWFBPC which should be coded as an
invalid value of "99"</txt>
  <command formalLangauge="SPSS">RECODE NBWFBPC ('27'='99'). EXECUTE.
</command>
</codingInstructions>
```

collectorTraining	**Element/attribute name**

```
Location:
<dataColl>
  <timeMeth>
  <dataCollector>
  <collectorTraining @type>        0..n
  <frequenc>
  <sampProc>
  <sampleFrame>
  <targetSampleSize>
  <deviat>
  <collMode>
  <resInstru>
  <instrumentDevelopment>
  <sources>
  <collSitu>
  <actMin>
  <ConOps>
  <weight>
  <cleanOps>
</dataColl>
```

This reflects new information that will be added to DDI-Lifecycle covering the type of training provided to the data collector. It includes a type attribute.

Usage: Allows for a general description of the training for data collectors of different types or levels. A more detailed structure will be added to DDI-L as a result of the work of the Survey Development and Implementation working group. The element was added in DDI-C with a generic content which may be structured using XHTML.

```
Example:
```

```
<collectorTraining type=”interviewer”>Each interviewer received a 5-
day course entitled “Proficient Interviewing”. Each interviewer was
tested at the end of the course and required a passing grade of 80%.
</collectorTraining>
```

dataAppr @type	**Data Appraisal NEW attribute type**

```
Location:
<anlyInfo>
  <respRate>
  <EstSmpErr>
  <dataAppr> @type
</anlyInfo>
```

A type attribute has been added to Data Appraisal (dataAppr) to support better classification of this activity.

Usage: Data Appraisal can take place at many points in the life of the data. This attribute allows data appraisal content to be specifically typed to represent various forms of appraisal. For example, a specific software that wishes to collect data appraisal information at the time of collection and again when the data moves into an archive, can type these two pieces of information and provide separate views of the field through a specific publication software.

```
Example:
<dataAppr type=”archiveIngest”>Data was compared to the metadata
content definition, checking declared record counts, line length,
valid value ranges, and comparing generated with declared summary
values.</dataAppr>
```

dataCollector @role	**Data Collector NEW attribute role**

```
Location:
<dataColl>
  <timeMeth>
  <dataCollector @abbr @affiliation @role>
  <collectorTraining>
  <frequenc>
  <sampProc>
  <sampleFrame>
  <targetSampleSize>
  <deviat>
  <collMode>
  <resInstru>
  <instrumentDevelopment>
  <sources>
  <collSitu>
  <actMin>
  <ConOps>
  <weight>
  <cleanOps>
</dataColl>
```

A role attribute was added to distinguish between different roles provided by different data collectors. This could differentiate between steps in the process or type of position such as supervisor.

Usage: There was an expressed need to be able to differentiate between different types and levels of data collectors or between different steps in a process.

```
Example:
<dataCollector role=”planner” affiliation=”University of
Michigan”>John Smith</dataCollector>
<dataCollector role=”supervisor” affiliation=”XYZ Survey
Services”>Mary Olson</dataCollector>
<dataCollector abbr=”XYZ” role=”contractor”>XYZ Survey
Services</dataCollector>
```

exPostEvaluation	**Ex-Post Evaluation**

```
Location:
<stdyInfo>
  <studyBudget>
  <subject>
  <abstract>
  <sumDscr>
  <qualityStatement>
  <notes>
  <exPostEvaluation @completionDate @type>    0..n
    <evaluator @affiliation @abbr @role>      0..n
    <evaluationProcess>                       0..n
    <outcomes>                                0..n
  </exPostEvaluation>
</stdyInfo>
```

This element has been added to capture the step of process evaluation for a study. It provides a means of capturing information on the completion date, type of evaluation, evaluator, evaluation process, and outcome of the evaluation. This relates to the step “Process Evaluation” found in the Generaic Statistical Business Process Model (GSBPM). A similar structure will be added to DDI-Lifecycle.

Usage: Ex-Post Evaluations are frequently done within large statistical or research agencies, in particular when the survey is intended to be repeated or on-going. This process is a best practice recommended in order to learn from current experience and prepare more effectively for further data collection. An equivilent of the complex element will be added to DDI-L.

```
Example:
<exPostEvaluation completionDate=”2003-12-31” type=”comprehensive”>
  <evaluator affiliation=”United Nations” abbr=”UNSD”
role=”consultant”>United Nations Statistical Division</evaluator>
  <evaluationProcess>In depth review of...</evaluationProcess>
  <outcomes>The following steps were highly effective in increasing
response rates and should be repeated in the next collection
cycle....</outcomes>
</exPostEvaluation>
```

instrumentDevelopment	**Instrument Development**

```
Location:
<dataColl>
  <timeMeth>
  <dataCollector>
  <collectorTraining>
  <frequenc>
  <sampProc>
  <sampleFrame>
  <targetSampleSize>
  <deviat>
  <collMode>
  <resInstru>
  <instrumentDevelopment @type>      0..n
  <sources>
  <collSitu>
  <actMin>
  <ConOps>
  <weight>
  <cleanOps>
</dataColl>
```

This reflects new information that will be added to DDI-Lifecycle covering the instrument development process. There is a type attribute and an option for structured content.

Usage: Instrument development is a complex activity not currently captured in DDI-C or DDI-L. Instrument development will be added to DDI-L based on the work of the Survey Development and Instrumentation working group. This new elment in DDI-C allows for a generic structure with optionally structured content to meet basic needs of DDI-C users. As the structure of DDI-L is finalized this element may gain additional structure.

```
Example:
<instrumentDevelopment type=”pretesting”>The survey instrument was
tested....</instrumentDevelopment>
```

qualityStatement	**Quality Statement**

```
Location:
<stdyInfo>
  <studyBudget>
  <subject>
  <abstract>
  <sumDscr>
  <qualityStatement>                               0..1
    <standardsCompliance>                          0..n
      <standard>                                   1..1
        <standardName @date @version @URI>         0..n
        <producer @abbr @affiliation @role>        0..n
      <complianceDescription>                      0..n
    </standardsCompliance>
    <otherQualityStatment>                         0..n
  </qualityStatement>
  <notes>
  <exPostEvaluation>
</stdyInfo>
```

This is a structured element that allows for the identification of specific standards to which the study complies and a description of that compliance. In addition, any form of quality statement not associated with a standard can be entered here.

Usage: Many official studies must comply with specific organizational and international standards. This element allows for identifying the name, version, source, and publisher of a standard and describe the extent to which the standard has been complied with. In addition, a quality statement may be provided that is unrelated to a specific standard. This allows for a range of statements dependent upon the needs of the user.

```
Example:
<qualityStatement>
  <standardsCompliance>
    <standard>
      <standardName date=”2011-08” version=”2” URI=”
http://www.census.gov/quality/standards/Quality_Standards.pdf”>U.S.
Census Bureau Statistical Qualty Standard</standardName>
      <producer abbr=”USCB” affiliation=”Department of Commerce”
role=”maintainer”>United States Census Bureau</producer>
    </standard>
    <complianceDescription>All requirements were met. No waivers
requested</complianceDescription>
  </stanardsCompliance>
  <otherQualityStatement>In addition to the above standard,
....</otherQualityStatement>
</qualitystatement>
```

sampleFrame	**Sample Frame**

```
Location:
<dataColl>
  <timeMeth>
  <dataCollector>
  <collectorTraining>
  <frequenc>
  <sampProc>
  <sampleFrame>                                        0..n
    <sampleFrameName>                                  0..n
    <labl>                                             0..n
    <txt>                                              0..n
    <validPeriod @event>                               0..n
    <custodian>                                        0..n
    <useStmt>                                          0..n
      <confDec @required @formNo @URI>                 0..1
      <specPerm @required @formNo @URI>                0..1
      <restrctn>                                       0..1
      <contact>                                        0..n
      <citReq>                                         0..1
      <deosReq>                                        0..1
      <conditions>                                     0..1
      <disclaimer>                                     0..1
    </useStmt>
    <universe>                                         0..n
    <frameUnit @isPrimary>                             0..n
      <unitType @numberOfUnits>                        1..1
      <txt>                                            0..n
    </frameUnit>
    <referencePeriod @event>                           0..n
    <updateProcedure>                                  0..n
  </sampleFrame>
  <targetSampleSize>
  <deviat>
  <collMode>
  <resInstru>
  <instrumentDevelopment>
  <sources>
  <collSitu>
  <actMin>
  <ConOps>
  <weight>
  <cleanOps>
</dataColl>
```

This reflects new information that will be added to DDI-Lifecycle covering the details of the sample frame used for the study sample. It includes information including the standard name, label, and description elements, plus a valid date for the frame, who maintains it, its use, universe, frame unit information, a reference period, and update procedures.

Usage: Sample frames are used as the basis for identifying a population to collect a sample from. This information was missing from earlier versions of DDI-C and the content was requested by producers using DDI-C. For example, a topical poll is taken of persons in St. Paul, MN. As this poll is conducted by telephone interviews the sample frame is a current phone directory for the City of St. Paul. Note that standard elements such as labl, txt, and useStmt have been used in this complex element. They contain the standard attributes and/or elements of their type.

```
Example:
<sampleFrame>
  <sampleFrameName>City of St. Paul Directory</sampleFrameName>
  <validPeriod event=”start”>2009-07-01</validPeriod>
  <validPeriod event=”end”>2011-06-30</validPeriod>
  <custodian>DEX Publications</custodian>
  <useStmt>
    <conditions>Payment of fee, non-disclosure or reuse of contents
outside this specified survey</conditions>
  </useStmt>
  <universe>Published phone numbers in the City of St Paul valid as of
2009-06-01. Phone book is published once every two years and is
```

```
considered valid for that time period</universe>
  <frameUnit isPrimary=”true”>
    <unitType numberOfUnits=”150000”>Primary owner of published phone
numbers in the City of St Paul</unitType>
    <txt>The phone numbers may represent one or more household members
and a household may have more than one phone number.</txt>
  <referencePeriod event=”single”>2009-06-01</referencePeriod>
  <updateProcedure>Changes are collected as they occur through
registration and loss of phone number from the specified geographic
area. Data is compiled for the date June 1, of odd numbered years and
published on July 1 for the succeeding 2 year
period.</updateProcedure>
</sampleFrame>
```

sourceCitation	**Source Citation**

```
Location:
<sources>
  <dataSrc>
  <sourceCitation>                        0..n
  <srcOrig>
  <srcChar>
  <srcDocu>
  <sources>
</sources>
```

A citation has been added to sources to allow for a specific identification of an external data source.

Usage: This standard citation element and its complete contents has been added for the purpose of providing a citation and holdings information on external data sources. See the new features of citation for detailed new content information.

```
Example:
<sourceCitation>
  <titlStmt>
    <titl>U.S. Bureau of the Census. Tenth Decennial Census of the
United States, 1880. Volume I. Statistics of the Population of the
United States at the Tenth Census.</titl>
  <titlStmt>
  <rspStmt>
    <AuthEnty affiliation="Department of Commerce">United States
Census Bureau</AuthEnty>
  </rspStmt>
  <prodStmt>
    <producer>Government Printing Office</producer>
    <prodDate>1883</prodDate>
    <prodPlac>Washington, D.C.</prodPlac>
  </prodStmt>
</sourceCitation>
```

studyAuthorization	**Study Authorization**

```
Location:
<stdyDscr>
```

```
  <citation>
  <studyAuthorization @date>               0..n
    <authorizingAgency @affiliation @abbr> 0..n
    <authorizationStatement>               0..n
  </studyAuthorization>
  <stdyInfo>
  <method>
  <dataAccs>
  <othrStdyMat>
  <notes>
```

Provides structured information on the agency that authorized the study, the date of authorization, and an authorization statement, i.e., a law authorizing a census, a statement from an Internal Review Board, etc.

Usage: Official statistical agencies frequently receive authorization to collect data through a specific law. Academic researcher often require the authorization of an Internal Review Board prior to fielding a survey. The complex element studyAuthorization provides a location to specifically note the source of authorization. Copies of laws or other documents may be cited in othStdyMat as appropriate.

```
Example:
<studyAuthorization date=”2010-11-04”>
  <authorizingAgency affiliation=”Purdue University” abbr=”OUHS”>
Office for Use of Human Subjects</authorizingAgency>
  <authorizationStatement>Required documentation covering the study
purpose, disclosure information, questionnaire content, and consent
statements was delivered to the OUHS on 2010-10-01 and reviewed by the
compliance officer. Statement of authorization for the described study
was received on 2010-11-04.</authorizationStatement>
</studyAuthorization>
```

studyBudget	**Study Budget**

```
Location:
<stdyInfo>
  <studyBudget>                             0..n
  <subject>
  <abstract>
  <sumDscr>
  <qualityStatement>
  <notes>
  <exPostEvaluation>
</stdyInfo>
```

Provides a text field where the overall budget of a study can be entered. The field allows for structured content.

Usage: A generic element for describing the overall study budget was provided at DDI-C user request. Internal structure is allowed using XHTML elements. Different organizations express their bugets in different formats and this open format allows flexibility.

```
Example:
<studyBudget>
 <xhtml:div>
   <xhtml:h1>Overall Budget</xhtml:h1>
   <xhtml:p>general statement</xhtml:p>
 </xhtml:div>
 <xhtml:div>
   <xhtml:h1>Detail</xhtml:h1>
   <xhtml:table>...</xhtml:table>
 </xhtml:div>
</studyBudget>
```

targetSampleSize | Target Sample Size

```
Location:
<dataColl>
  <timeMeth>
  <dataCollector>
  <collectorTraining>
  <frequenc>
  <sampProc>
  <sampleFrame>
  <targetSampleSize>                0..n
    <sampleSize>                    0..1
    <sampleSizeFormula>             0..1
  </targetSampleSize>
  <deviat>
  <collMode>
  <resInstru>
  <instrumentDevelopment>
  <sources>
  <collSitu>
  <actMin>
  <ConOps>
  <weight>
  <cleanOps>
</dataColl>
```

Allows for specific information regarding the target sample size, actual sample size, and the formula used to determine this.

Usage: Target sample size allows for stating the desired “target” sample size of the study. The sample size is expressed as an integer and the sample formula may be provided.

```
Example:
<targetSampleSize>
  <sampleSize>400</sampleSize>
  <sampleSizeFormula>Ewen’s sampling formula</sampleSizeFormula>
</targetSampleSize>
```

File Description (fileDscr)

fileCitation	**File Citation**

```
Location:
<fileTxt>
  <fileName>
  <fileCitation>              0..1
  <fileCont>
  <fileStrc>
  <dimensns>
  <fileType>
  <format>
  <filePlac>
  <dataChck>
  <ProcStat>
  <dataMsng>
  <software>
  <verStmt>
</fileTxt>
```

Allows for the provision of a citation to a specific data file that is being described by the DDI-Codebook.

Usage: The inclusion of a file citation allows for the capture of file identification information found in DDI-L Physical Instance. This citation, in combination with the @fileStrcRef in fileStrc can create a one-to-one relationship between a fileTxt element and an external data file. A DDI-C user can create a fileTxt element for each external file and provide the citation, location, and DOI or other unique identifier in the IDNo element available in citation.

```
Example:
<fileTxt>
  <fileCitation>
    <titlStmt>
      <titl>Census of Population and Housing: Summary File 1, 2000:
Summary Level 040 (State)</titl>
      <IDNo agency=”G_DOI” level=”file”>GDO-0001</IDNo>
    </titlStmt>
  </fileCitation>
<fileTxt>
```

fileStrc @fileStrcRef	**File Structure NEW attribute fileStrcRef**

```
Location:
<fileTxt>
  <fileName>
  <fileCitation>
  <fileCont>
  <fileStrc @fileStrcRef>
  <dimensns>
  <fileType>
  <format>
  <filePlac>
  <dataChck>
  <ProcStat>
  <dataMsng>
  <software>
  <verStmt>
</fileTxt>
```

This allows for a single description of a file structure when multiple data files share the same structure. After the first file's structure is described, the others can simply reference this description.

Usage: Frequently a study or archive will contain a number of files all using the same content structure but held in a number of files based on geographic coverage, year, case subset, etc. The addition of a fileStrcRef to fileStrc allows the file structure to be declared a single time and then referenced by other files using the same structure. This element in combination with the use of a fileCitation allows for a complete listing of files using the same metadata and file structure.

```
Example:
<fileTxt>
  <fileName ID="F2000SF1-040" source="archive">2000SF1-040</fileName>
  <fileCitation>
    <titlStmt>
      <titl>Census of Population and Housing: Summary File 1, 2000:
Summary Level 040 (State)</titl>
      <IDNo agency=”G_DOI” level=”file”>GDO-0001</IDNo>
    </titlStmt>
  </fileCitation>
  <fileStrc ID=”FS_1” type=”rectangular”>
    <recGrp ID="REC_1">
      <labl level="recGrp">Physical Record 1 of 1</labl>
      <recDimnsn>
        <logRecL>71553</logRecL>
      </recDimnsn>
    </recGrp>
  </fileStrc>
  <dimensns>
    <caseQnty>52</caseQnty>
    <recPrCas>1</recPrCas>
  </dimensns>
</fileTxt>
<fileTxt>
  <fileName ID="F2000SF1-050" source="archive">2000SF1-050</fileName>
  <fileCitation>
    <titlStmt>
      <titl>Census of Population and Housing: Summary File 1, 2000:
Summary Level 050 (County)</titl>
      <IDNo agency=”G_DOI” level=”file”>GDO-0002</IDNo>
    </titlStmt>
  </fileCitation>
  <fileStrc fileStrcRef=”FS_1”>
  <dimensns>
    <caseQnty>3152</caseQnty>
    <recPrCas>1</recPrCas>
  </dimensns>
</fileTxt>
```

Data Description (dataDscr)

qstn @responseDomainType @otherResponseDomainType	Question NEW attributes responseDomainType and otherResponseDomainType

```
Location:
<qstn @qstn @var @seqNo @sdatrefs @responseDomainType
@otherResponseDomainType>
  <preQTxt>
  <qstnLit>
  <postQTxt>
  <forward>
  <backward>
  <ivuInstr>
</qstn>
```

Questions have a new attribute which allows for the direct identification of a specific DDI-Lifecycle response domain type (i.e., code, datetime, numeric, etc). This clarifies any ambiguity for transforms from DDI-Codebook to DDI-Lifecycle and allows for specific identification of this information when transforming from DDI-Lifecycle to DDI-Codebook.

Usage: DDI-L differentiates between a number of different response domains for questions which may differ from the representation of the variable created from the question. Frequently this is only represented in DDI-C as text within the literal question. The attribute responseDomainType allows for capturing the result of a content review as a specified response domain type valid in DDI-L. The response set contains the standard types available in DDI 3.1 (text, numeric, code, category, datetime, geographic, multiple) plus other. If new forms of response domains (such as scale) are added to DDI-L these attributes would contain the following: responseDomainType=”other” otherResponseDomainType=”scale”.

```
Example:
<qstn responseDomainType=”category”>
  <qstnLit>
    What is the gender of your dog?
    [] Male
    [] Female
  </qstnLit>
</qstn>
```

var @representationType @otherRepresentationType	Variable NEW attribute representationType and otherRepresentationType

```
Location:
<var @name @wgt @wgt-var @weight @qstn @files @vendor @dcml @intrvl
@rectype @sdatrefs @methrefs @pubrefs @access @aggrMeth @otherAggrMeth
@measUnit @scale @origin @nature @additivity @otherAdditivity @geog
@geoVocab @catQnty @representationType @otherRepresentationType>
...
</var>
```

Variables have a new attribute which allows for the direct identification of a specific DDI-Lifecycle representation type (i.e., category, code, text, etc). This clarifies any ambiguity for transforms from DDI-Codebook to DDI-Lifecycle and allows for specific identification of this information when transforming from DDI-Lifecycle to DDI-Codebook.

Usage: DDI-L differentiates between a number of different representation types for variables which are more explicit than those of DDI-C. Frequently variables contain both category responses AND a description of a numeric response range in DDI-C. The attribute representationType allows for capturing the result of a content review as a specified representation type valid in DDI-L. The response set contains the standard types available in DDI 3.1 (text, numeric, code, and datetime) plus other. If new forms of representation (such as scale) are added to DDI-L these attributes would contain the following: representationType=”other” otherRepresentationType=”scale”.

```
Example:
<var name=”DogGender” representationType=”code”>
  <labl level=”var”>Dog Gender</labl>
  <qstn responseDomainType=”category”>
    <qstnLit>
      What is the gender of your dog?
      [] Male
      [] Female
    </qstnLit>
  </qstn>
  <valrng>
    <range min=”1” max=”2”/>
  </valrng>
  <catgry>
    <catVal>1</catVal>
    <labl>Male</labl>
  </catgry>
  <catgry>
    <catVal>2</catVal>
    <labl>Female</labl>
  <catgry>
```