

Identification, Versioning, Maintenance and Reference

The identification structure of DDI objects is core to the functioning of the standard. The purpose of the DDI identification structure is to:

- Uniquely identify major objects in a persistent manner to ensure accurate reference and retrieval of the object content
- Provide context for objects where an understanding of related object types within a packaging structure is essential to the understanding of the object (i.e., a specific classification within a classification scheme)
- Manage metadata object change over time
- Support the range of object management used by different organizations
- Support early and late binding of references
- Support interaction with closely related standards, in particular ISO/IEC 11179 and SDMX

The identification structure is based on the ISO/IEC 11179 structure that requires a three-part means of unique identification.

OBJECT	ISO/IEC 11179	DDI
Agency Identifier	A unique identifier for the agency managing the object	A unique identifier for an agency registered with the DDI Alliance. The agency may have multiple sub-agency extensions managed within the DDI Registry or within the primary agency. An agency and sub-agency are separated by an “.”.
Unique ID	A unique identifier of the object within the context of the agency	An identification which is unique within a) the agency (sub-agency), or b) within the parent maintainable. If the context is the parent maintainable the Unique ID is the ID of the parent maintainable plus the ID of the object within that maintainable separated by a “.”.
Version Number	A version number of the object to track change over time	A version number with any number of extensions separated by a “.”.

This three part structure is the equivalent of a unique persistent identifier for an object, such as described by DOIs and other similar structures. Note that while use of a version number with a DOI is optional, based on local practice, the Version Number in DDI is required due to the need to manage metadata within a dynamic workflow over time.

Identifiable, Versionable, and Maintainable

DDI differentiates between a set of element types. Not all objects are individually identifiable, i.e. some objects only have meaning within the context of an identifiable object such as a Label or Description. The remaining objects are Identifiable, Versionable or Maintainable in order to support different levels of metadata management. Identifiable objects are those that can be referenced directly either for inclusion in another object or for the purpose of attaching Other Material or a Note to the object.

Identifiable objects have a unique ID within the context of their specified scope of uniqueness (see Scope of Uniqueness discussion within this section). Their Agency identification and Version Number match those of the object's immediate parent Versionable (or Maintainable if there is not a parent Versionable) at the **point of creation**. This means that if an Identifiable object is created within a version 1 of a parent Versionable and does NOT change its content over time it will retain its Version Number of 1 until the identifiable object itself is altered. It will then change its Version Number to that of its parent Versionable **at the time of the alteration**. In other words an Identifiable could go from a Version 1 to a Version 4 without ever having a Version 2 or 3 if the cause for versioning did not involve any change in the Identifiable object within Version 2 and 3 of the parent Versionable.

A *Versionable object* has the characteristics of an Identifiable object but may be managed over time. DDI has determined that being able to track change within the object over time is a requirement, either to understand the relationship to earlier objects of a similar type or to track provenance. Note that it is up to the individual content provider to determine whether an object is essentially new or is a modification (version) of an earlier object. Versionable objects have a unique ID within the context of their specified scope of uniqueness (see Scope of Uniqueness discussion within this section). Their Agency identification matches that of the object's immediate parent Maintainable at the **point of creation**. In other words the Agency of an object does not change simply because it is included by reference in a Maintainable managed by a different agency. The Version number of the object changes each time its content changes. See Versioning for a discussion of when and how this may be implanted within different organizations or projects.

A *Maintainable object* is a form of packaging and generally takes the form of either a module or scheme. Modules package metadata focused on specified segments of the Lifecycle for which context is important for understanding. Schemes are similar to data base tables, containing a stack of similar type objects that many have important contextual relevance to each other, i.e. a classification scheme captured in a DDI Category Scheme. There are two unique forms of a Maintainable, Instrument and CodeList, which are Maintainable in their own right to support the statistical production process, but which can only be published within the context of a parent Maintainable Scheme. All Schemes and Modules may be published within the context of a Study Unit or Group (a collection of Study Units) or as a separate Resource Package item primarily for the purpose of reuse.

Versioning

In general, once metadata is published, flagged as being stable for referencing purposes, any change to the content should result in a version change to the Versionable object containing the change. Note that as Versionable objects are contained by Maintainable objects and that Maintainable object may be

contained in another Maintainable object, a single change will generally trigger Versioning up the containing tree of the metadata. The purpose of versioning is to ensure that someone referencing a specific version of an object will always get back the same information.

Versioning rules and Version Number structures differ between different organizations and between different projects or products of the same organization. DDI does not dictate this structure EXCEPT to specify that it must be expressed as one or more integers separated by a "." (dot). The DDI Lifecycle specification uses a three part structure of a Major.Minor.Sub-Minor version number. The Organization should describe its versioning system or systems so that it is clear to the user when and how versioning occurs. Versioning is "required" once a Maintainable object is flagged `isPublished="true"`. During production processes, tracking version changes may be managed by other means such as Version Date or external subversion system.

Some organizations are stricter in their versioning rules than others. For example, a typographical correction within a Description text which is considered to have no impact on the intellectual content may trigger a Minor or Sub-minor version change in one system while only result in Version Date change in another. Because the impact of a change cannot be easily predicted (it is dependent upon the use of the metadata) what is important is to capture in some way that a change was made and to provide the end user with a clear set of versioning rules that supports their ability to evaluate the impact of the change for their particular use. In addition, some metadata is considered local in nature, specific to interaction with an identified system. This metadata is considered to be Administrative in nature and is viewed by DDI as a set of metadata that does not alter the intellectual content (Payload) of the metadata and does not need to drive a version change.

Administrative and Payload Metadata

The differentiation between Administrative metadata and Payload metadata lies primarily how changes to the metadata content drive version changes.

Administrative metadata is content that is related to the interaction or use of the metadata within a specific system. Changes in administrative metadata do not change the meaning of the metadata content of the object in terms of its DDI identification or intellectual meaning. For example, the addition of a local User Identification to facilitate interaction between the metadata object and local access software, or creating a URN entry from existing identification sequence content are considered to be changes to the Administrative metadata. A change in Administrative metadata does not trigger a change in the version number of a published object.

Payload metadata is all metadata NOT defined as Administrative. Payload metadata contains intellectual content that has an impact on the meaning or application of the object. A change in Payload metadata generally triggers a change in the version number of a published object.

The following objects which consist of the extension bases used for identification and referencing purposes are considered to be Administrative metadata for the purposes of versioning:

Extension bases are listed in RED and contained objects in BLACK	General description
---	----------------------------

IDENTIFIABLE OBJECT

URN
 Agency
 ID
 Version
 UserID
 @typeOfUserID
 @userIDVersion
 @typeOfUserVersion
 MaintainableObject
 TypeOfObject
 MaintainableID
 MaintainableVersion
 @typeOfIdentifier
 @inheritanceAction
 @objectSource
 @scopeOfUniqueness
 @isIdentifiable

Extension base is AbstractIdentifiable

See URN Structure below
 Base sequence of identification. If sequence is used, all are required.

 A user specified identification for a specific system such as a DOI or internal search engine. Both the ID and @typeOfUserID must be specified. Version information is optional.
 The Maintainable object containing the Identifiable object. TypeOfObject is required to create Deprecated URN. ID is required for any URN if ScopeOfUniqueness="Maintainable". Version number provides context base. URN used by Agency [Canonical | Deprecated]
 See Grouping: Inheritance
 The object source of a resolved reference
 Scope used by Agency [Agency | Maintainable
 Fixed value = "true" [specifies object base]

VERSIONABLE OBJECT

URN
 Agency
 ID
 Version
 UserID
 @typeOfUserID
 @userIDVersion
 @typeOfUserVersion
 UserAttributePair
 AttributeKey
 AttributeValue
 VersionResponsibility
 VersionResponsibilityReference
 VersionRationale
 BasedOnReference
 MaintainableObject
 TypeOfObject
 MaintainableID
 MaintainableVersion

Extension base is AbstractVersionable

Same as Identifiable Object

 User defined Key/Value pair used to support interaction of the metadata within the user's system.
 Who within the Agency versioned the object
 Alternate reference to who versioned the object
 Reason for version change (to inform user)
 Intellectual base of this object (local version of an external object)
 Same as Identifiable Object

@typeOfIdentifier
@inheritanceAction
@objectSource
@scopeOfUniqueness
@isVersionable
@versionDate

Fixed value = "true" [specifies object base]
Date/Time of version change

MAINTAINABLE OBJECT

URN
Agency
ID
Version
UserID
 @typeOfUserID
 @userIDVersion
 @typeOfUserVersion
UserAttributePair
 AttributeKey
 AttributeValue
VersionResponsibility
VersionResponsibilityReference
VersionRationale
BasedOnReference

Extension base is AbstractMaintainable
Same as Versionable object

Note

Software

MetadataQuality
 @typeOfIdentifier
 @inheritanceAction
 @objectSource
 @scopeOfUniqueness
 @isMaintainable
 @versionDate
 @externalReferenceDefaultURI

 @isPublished

 @xml:lang

Notes related to objects with the maintainable (Payload)
Software used to create the Maintainable object (Payload)
Quality of the metadata in the Maintainable object (Payload)
Same as Versionable object

Fixed value = "true" [specifies object base]
Same as Versionable object

Indicates that the content is available for reuse by reference
The language of the metadata in the Maintainable object (Payload)

REFERENCE TYPE

URN

The URN of the object being referenced using the specified URN structure

Agency	The agency of the object being referenced.
ID	The ID of the object being referenced.
Version	The Version of the object being referenced. This is the full Version number at the time the reference is created.
TypeOfObject	Type of object being referenced. This is a controlled list.
MaintainableObject	The Maintainable object containing of the Identifiable or Versionable object being referenced. TypeOfObject is required to create
TypeOfObject	Deprecated URN. ID is required for any URN if
MaintainableID	ScopeOfUniqueness="Maintainable". Version number provides context base.
MaintainableVersion	URN used by Agency [Canonical Deprecated]
@typeOfIdentifier	Boolean attribute. If "true" you must supply the URN
@isExternal	A local store for the referenced object expressed as a URI
@externalReferenceDefaultURI	Fixed value = "true" [specifies object base]
@isReference	Boolean attribute. Set to "true" if you wish to late-bind the reference (i.e., want to reference the most recent version)
@lateBound	If @lateBound="true" and this attribute is not provided you will get the most recent version. Use this attribute to restrict the value of the late-bind, for example to restrict to the most recent minor version of major version 4
@lateBoundRestriction	@lateBoundRestriction="4". Specify the desired language content (if available) for the referenced item using a valid xml:lang value.
@objectLanguage	The URN of the parent maintainable at the time of reference (this is not necessarily the same version number as the version of the parent maintainable at point of origin)
@sourceContext	

SCHEME REFERENCE TYPE

URN	Same as Reference.
Agency	
ID	
Version	
TypeOfObject	
MaintainableObject	
TypeOfObject	
MaintainableID	
MaintainableVersion	

Exclude	Allows the identification of objects within the Scheme which are to be excluded for the purpose of this reference.
ID	The ID of the excluded object.
Version	The Version of the excluded object.
@typeOfIdentifier	Same as Reference.
@isExternal	
@externalReferenceDefaultURI	
@isReference	
@lateBound	
@lateBoundRestriction	
@objectLanguage	
@sourceContext	

Scope of Uniqueness

DDI 3.2 supports scoping the uniqueness of identifier to the parent Maintainable or to the Agency (sub-agency). This choice affects the structure of the ID as it appears within the Canonical URN (see Type of Identifier in this section). Essentially, when the ID is scoped to the Agency the unique identification of an object requires the Agency, ID of the object, and Version Number of the object. When the ID is scoped to the Maintainable the unique identification of a non-Maintainable object requires the Agency, ID of the parent Maintainable, the ID of the object, and the Version Number of the object. The attribute `scopeOfUniqueness` is required and must contain either “Agency” or “Maintainable”. This attribute defines how the ID will be expressed in the Canonical URN and what is required for a complete reference to the object within the Maintaining Agency.

Type of Identifier

DDI 3.2 supports two formats for expressing the Identification Sequence as a URN. The Canonical URN is recommended. The Deprecated URN is a modification of the DDI 3.1 URN. Whether the identification information is expressed as a URN or using the Identification Sequence the `TypeOfIdentifier` dictates the format of the URN supported internally by the agency and the format of the URN used by an external reference to that object. Note that regardless of the type of identifier used it is good practice to provide the full content of the Identification Sequence and details of the `MaintainableObject` for non-Maintainable objects in order to support the creation of any format of a DDI URN.

Structure of the URN

Canonical URN

Each section of the Canonical URN is separated by a “:” (colon). Within the ID section the Maintainable ID and Object ID are separated by a “.” (dot).

“urn:ddi:agency[.sub-agency]:ID:Version”

If the `scopeOfUniqueness` equals “Agency” the ID is the ID of the object.

Example:

Canonical URN with uniqueness scoped to the Agency.

Object V321 version 2 within the Minnesota Population Center (us.mpc)

urn:ddi:us.mpc:V321:2

Object V321 version 2 within the Minnesota Population Center, Project IPUMS (listed as a sub-agency within us.mpc)

urn:ddi:us.mpc.ipums:V321:2

If the scopeOfUniqueness equals "Maintainable" the ID of a non-Maintainable object is structured as follows:

"urn:ddi:agency[.sub-agency]:MaintainableID.ObjectID:Version"

Canonical URN with uniqueness scoped to the Maintainable.

Variable V321 version 2 within VariableScheme VS1 at the Minnesota Population Center (us.mpc)

urn:ddi:us.mpc:VS1.V321:2

Variable V321 version 2 within VariableScheme VS1 at the Minnesota Population Center, Project IPUMS (listed as a sub-agency within us.mpc)

urn:ddi:us.mpc.ipums:VS1.V321:2

Deprecated URN

Each section of the Deprecated URN is separated by a ":" (colon).

"urn:ddi:agency[.sub-agency]:MaintainableObjectType:MaintainableID:ObjectID:ObjectVersion"

If the object itself is Maintainable the information on the parent maintainable is not included:

"urn:ddi:agency[.sub-agency]:ObjectID:ObjectVersion"

If the scopeOfUniqueness equals "Agency" the ID is the ID of the object.

Example:

Deprecated URN with uniqueness scoped to the Agency.

Object V321 version 2 within the Minnesota Population Center (us.mpc)

urn:ddi:us.mpc:Variable:V321:2

Object V321 version 2 within the Minnesota Population Center, Project IPUMS (listed as a sub-agency within us.mpc)

```
urn:ddi:us.mpc.ipums:Variable:V321:2
```

If the scopeOfUniqueness equals "Maintainable" the ID of a non-Maintainable object is structured as follows:

```
"urn:ddi:agency[.sub-agency]:MaintainableID.ObjectID:Version"
```

Deprecated URN with uniqueness scoped to the Maintainable.

Variable V321 version 2 within VariableScheme VS1 at the Minnesota Population Center (us.mpc)

```
urn:ddi:us.mpc:VariableScheme:VS1:Variable:V321:2
```

Variable V321 version 2 within VariableScheme VS1 at the Minnesota Population Center, Project IPUMS (listed as a sub-agency within us.mpc)

```
urn:ddi:us.mpc.ipums:VariableScheme:VS1:Variable:V321:2
```

EXAMPLES:

Note that by including the information for the MaintainableObject in an Identifiable or Versionable the user has provided sufficient information to build either a Canonical or Deprecated URN scoped to either the agency or the maintainable. While the information may not be part of the URN as expressed by the maintaining agency, the information contained in the MaintainableObject provides contextual information that may be important to the user and may need to be passed to them for purposes other than strict identification. The inclusion of Versioning information in Versionable and Maintainable may be used internally to track quality control and processing activities and may also be valuable to the user in determining whether the change caused by versioning will affect their analysis work.

Identifiable

```
<l:Code isIdentifiable="true" typeOfIdentifier="Canonical"
scopeOfUniqueness="Maintainable">
  <r:URN>urn:ddi:us.mpc:CL_1.Code_1:1</r:URN>
  <r:Agency>us.mpc</r:Agency>
  <r:ID>Code_1</r:ID>
  <r:Version>1</r:Version>
  <r:UserID typeOfUserID="NHGIS">Gender_1</r:UserID>
  <r:MaintainableObject>
    <r:TypeOfObject>CodeList</r:TypeOfObject>
    <r:MaintainableID>CL_1</r:MaintainableID>
    <r:MaintainableVersion>1</r:MaintainableVersion>
  </r:MaintainableObject>
  ...
</l:Code>
```

Minimum required content of the above:

```
<l:Code isIdentifiable="true" typeOfIdentifier="Canonical"
scopeOfUniqueness="Maintainable">
  <r:URN>urn:ddi:us.mpc:CL_1.Code_1:1</r:URN>
  ...
</l:Code>
```

Versionable

```
<l:Variable isVersionable="true" typeOfIdentifier="Canonical"
scopeOfUniqueness="Agency" versionDate="2012-10-31">
  <r:URN>urn:ddi:us.mpc:Var_1234:2</r:URN>
  <r:Agency>us.mpc</r:Agency>
  <r:ID>Var_1234</r:ID>
  <r:Version>2</r:Version>
  <r:UserID typeOfUserID="IPUMS">MOMLOC</r:UserID>
  <r:VersionResponsibility>John Doe</r:VersionResponsibility>
  <r:VersionRationale>
    <r:RationaleDescription><r:String xml:lang="en">Expanded
description to more clearly define the process of determining the
MOCLC value in households with multiple
mothers.</r:String></r:RationaleDescription>
    <r:RationaleCode>Update</r:RationaleCode>
  </r:VersionRationale>
  <r:MaintainableObject>
    <r:TypeOfObject>VariableScheme</r:TypeOfObject>
    <r:MaintainableID>VS_IPUMS</r:MaintainableID>
    <r:MaintainableVersion>6</r:MaintainableVersion>
  </r:MaintainableObject>
  ...
</l:Variable>
```

Minimum required content of the above:

```
<l:Variable isVersionable="true" typeOfIdentifier="Canonical"
scopeOfUniqueness="Agency" versionDate="2012-10-31">
  <r:URN>urn:ddi:us.mpc:Var_1234:2</r:URN>
  ...
</l:Variable>
```

Maintainable

```
<l:VariableScheme isVersionable="true" typeOfIdentifier="Canonical"
scopeOfUniqueness="Agency" versionDate="2012-10-31" isPublished="true"
xml:lang="en">
  <r:URN>urn:ddi:us.mpc:VS_IPUMS:6</r:URN>
  <r:Agency>us.mpc</r:Agency>
  <r:ID>VS_IPUMS</r:ID>
  <r:Version>6</r:Version>
  <r:UserID typeOfUserID="IPUMS">IPUMS_VARS</r:UserID>
  <r:VersionResponsibility>John Doe</r:VersionResponsibility>
  <r:VersionRationale>
```

```

    <r:RationaleDescription><r:String xml:lang="en">Versioned
MOMLOC</r:String></r:RationaleDescription>
    <r:RationaleCode>Update</r:RationaleCode>
  </r:VersionRationale>
  <r:Software></r:Software>
  <r:MetadataQuality>
    <r:QualityMeasure>InternalReview</r:QualityMeasure>
    <r:MeasurePurpose><r:Content xml:lang="en">Content has be
reviewed and confirmed for use on Public
site.</r:Content></r:MeasurePurpose>
    <r:MeasureValue>Public</r:MeasureValue>
  </r:MetadataQuality>
  ...
</l:VariableScheme>

```

Minimum required content of the above:

```

<l:VariableScheme isVersionable="true" typeOfIdentifier="Canonical"
scopeOfUniqueness="Agency" versionDate="2012-10-31" isPublished="true"
xml:lang="en">
  <r:URN>urn:ddi:us.mpc:VS_IPUMS:6</r:URN>
  ...
</l:VariableScheme>

```

Reference

Note that in this case Version 1 of Var_1234 originally appeared in Version 1 of VS_IPUMS. However, the sourceContext indicates that VS_IPUMS:4 is the context at the point of reference.

```

<l:VariableReference isReference="true" isExternal="false"
lateBound="false" objectLanguage="en"
sourceContext="urn:ddi:us.mpc:VS_IPUMS:4.0"
typeOfIdentifier="Canonical">
  <r:URN>urn:ddi:us.mpc:Var_1234:1.0</r:URN>
  <r:Agency>us.mpc</r:Agency>
  <r:ID>Var_1234</r:ID>
  <r:Version>1.0</r:Version>
  <r:TypeOfObject>Variable</r:TypeOfObject>
  <r:MaintainableObject>
    <r:TypeOfObject>VariableScheme</r:TypeOfObject>
    <r:MaintainableID>VS_IPUMS</r:MaintainableID>
    <r:MaintainableVersion>1.0</r:MaintainableVersion>
  </r:MaintainableObject>
</l:VariableReference>

```

Minimum required content of the above:

```

<l:VariableReference isReference="true" isExternal="false"
lateBound="false" typeOfIdentifier="Canonical">
  <r:URN>urn:ddi:us.mpc:Var_1234:1.0</r:URN>
  <r:TypeOfObject>Variable</r:TypeOfObject>

```

```
</l:VariableReference>
```

Above reference as a lateBound reference where the most recent minor version of major version 1 of the variable is being requested.

```
<l:VariableReference isReference="true" isExternal="false"
lateBound="true" objectLanguage="en"
sourceContext="urn:ddi:us.mpc:VS_IPUMS:4.0"
typeOfIdentifier="Canonical" lateBoundRestriction="1">
  <r:URN>urn:ddi:us.mpc:Var_1234:1.0</r:URN>
  <r:Agency>us.mpc</r:Agency>
  <r:ID>Var_1234</r:ID>
  <r:Version>1.0</r:Version>
  <r:TypeOfObject>Variable</r:TypeOfObject>
  <r:MaintainableObject>
    <r:TypeOfObject>VariableScheme</r:TypeOfObject>
    <r:MaintainableID>VS_IPUMS</r:MaintainableID>
    <r:MaintainableVersion>1.0</r:MaintainableVersion>
  </r:MaintainableObject>
</l:VariableReference>
```

SchemeReference

```
<l:VariableSchemeReference isReference="true" isExternal="false"
lateBound="false" objectLanguage="en" typeOfIdentifier="Canonical">
  <r:URN>urn:ddi:us.mpc:VS_IPUMS:1.0</r:URN>
  <r:Agency>us.mpc</r:Agency>
  <r:ID>VS_IPUMS</r:ID>
  <r:Version>1.0</r:Version>
  <r:TypeOfObject>VariableScheme</r:TypeOfObject>
  <r:Exclude isReference="true" isExternal="false" lateBound="false"
typeOfIdentifier="Canonical">
    <r:URN>urn:ddi:us.mpc:Var_1234:1.0</r:URN>
    <r:TypeOfObject>Variable</r:TypeOfObject>
  </l:Exclude>
</l:VariableSchemeReference>
```

Minimum required content of the above:

```
<l:VariableSchemeReference isReference="true" isExternal="false"
lateBound="false" objectLanguage="en" typeOfIdentifier="Canonical">
  <r:URN>urn:ddi:us.mpc:VS_IPUMS:1.0</r:URN>
  <r:TypeOfObject>VariableScheme</r:TypeOfObject>
  <r:Exclude isReference="true" isExternal="false" lateBound="false"
typeOfIdentifier="Canonical">
    <r:URN>urn:ddi:us.mpc:Var_1234:1.0</r:URN>
    <r:TypeOfObject>Variable</r:TypeOfObject>
  </l:Exclude>
</l:VariableSchemeReference>
```