Data Documentation Initiative

DDI Working Paper Series -- Best Practices, No. 8

Subject

Versioning and Publication (2009-03-22)

Document identifier:

http://dx.doi.org/10.3886/DDIBestPractices08

Authors:

Michelle Edwards, Janet Eisenhauer, Jane Fry, Pascal Heus, Kirstine Kolsrud, Meinhard Moschner, Ron Nakao, Wendy Thomas

Editors:

Wendy Thomas

Target audience:

Anyone publishing a DDI instance

Anyone maintaining a collection of published DDI documents

Developers of DDI applications

Abstract:

One of the objectives in creating DDI 3.0 was full machine-actionability. This requires strict versioning of objects so that users understand the change history of the resources they are using. This Best Practice is designed to provide some guidelines to metadata creators and publishers about how to version metadata and publish it for others to use.

Status: Draft

This document is updated periodically on no particular schedule. Send comments to ddi-bp-editors@icpsr.umich.edu

INTRODUCTION 3
1.1 Problem statement 3
1.2 Terminology 3
2 BEST PRACTICE SOLUTION 3
2.1 Definitions 3
2.2 Best Practice Behaviors 6
2.3 Discussion 10
2.4 Example 11
3 REFERENCES 14
3.1 Normative 14
APPENDIX A. ACKNOWLEDGMENTS 15
APPENDIX B. REVISION HISTORY 17
APPENDIX C. LEGAL NOTICES 18

Introduction

1.1 Problem statement

This best practice addresses what constitutes a published version of a metadata set, and how versions are managed. Versioning the data is a separate issue but should reflect a consistent set of practices within an organization or study. Data and metadata are versioned independently. The link between any version of the data and its related metadata is made in Physical Instance. Whenever a data file is versioned, that change should be reflected in the Physical Instance which would result in a versioning of the Physical Instance module. Note that a change in another section of the metadata would not result in a version change for the data.

Versioning must take place once a DDI instance is "published". Version changes occur for a variety of reasons, which have different implications for internal and external references as well as for end users and their understanding of the data. Once published, however, a version must be maintained without change. A change in the metadata content must result in a new version. This has implications for how many organizations work with their metadata, before the point at which it is externally published. This best practice addresses the reasons for versions at different levels of the metadata, the possible implications for references and the end user, how to manage versioning in a consistent predictable manner, and the requirements of an information system and services for managing the publication and updating of DDI instances. This includes packaging portions of the metadata that are expected to change, such as Other Material, lists of related publications, the physical data product, and the physical instance.

1.2 Terminology

The key words *must, must not, required, shall, shall not, should, should not, recommended, may,* and *optional* in this document are to be interpreted as described in **[RFC2119].** Additional DDI standard terminology and definitions are found in http://www.ddialliance.org/bp/definitions

2 Best Practice Solution

2.1 Definitions

Published metadata: Published metadata is considered available for use outside of the community that created the original document. This broader audience may be internal to a

project or organization or external. Metadata that is published must be wrapped in a DDI instance, versioned, and available for reuse or reference from outside of the instance.

Packaging as a DDI instance does not necessarily mean packaging for publication. Metadata may be packaged for reasons other than publication during its internal development process. In these cases versioning is not required.

Versioning: The process of providing a unique identifier for an element or entity that changes over time. Versioned elements retain their original ID but their version number is incremented to reflect a difference in content. This allows a reference to persist through the ID while allowing for either the specified version or the most current version of the element to be obtained. **What is versioned, maintained, and referenced in the DDI 3.0 is the metadata itself, rather than the XML which expresses that metadata. While this might seem like a minor distinction, it has major implications for how applications are developed**.

Maintenance agencies: These organizations own the metadata objects they maintain, and only they are allowed to make changes to those objects.

Minor version: The definition and level of detail of a minor version varies according to what is being published. The minor version information is always located to the right of the first decimal and can be further subdivided at the discretion of the maintaining agency.

Major version: The definition of a major version varies according to what is being published. However, major versions are expressed by the digits to the left of the decimal point.

Version Rationale is an optional element in all versionable elements and provides a location for indicating the reason for the change, e.g., correction of a typographical error or correction of inaccurate content that may affect analysis performed using earlier content.

LateBound: A specification for a reference that requests the most recent version available.

EarlyBound: A specification for a reference that requests the exact version noted.

Resource package: A resource package is a means of packaging any maintainable set of metadata for referencing as part of a study unit or group. A resource package structures materials for publication that are intended to be reused by multiple studies, projects, or communities of users. A resource package uses the group module with an alternative top-level element called Resource Package that is used to describe maintainable modules or schemes that may be used by multiple study units outside of a group structure.

DDI instance: DDI Instance is the top-level wrapper for any DDI document. It may contain a set of top-level elements, which generally correspond to the modular breakdown within DDI. Every DDI Instance will use this wrapper, regardless of its content.

Metadata registry: A metadata registry is a central location in an organization where metadata definitions are stored and maintained in a controlled way.

Identifiable (in the context of DDI): “Identifiables” are those elements in DDI that carry only the basic level of identification: a URN, ID, and Name.

Versionable (in the context of DDI): “Versionables” comprise a subset of DDI “identifiable” elements. These are elements for which changes in content are important to note and thus additional attributes related to versioning are enabled.

Maintainable (in the context of DDI): “Maintainables” are complex objects that can be maintained outside of a DDI Instance (published as separate entities). Their identification strings ensure that they are globally unique.

2.2 Best Practice Behaviors

General guidelines for versioning

Published metadata is considered available for use outside of the community that created the original document. This broader audience may be internal to a project or organization or external. Metadata that is published must be wrapped in a DDI instance, versioned, and made available for reuse or reference from outside of the instance. The primary point is that the metadata is available for reuse or reference outside of the instance and therefore must be persisted by the maintenance agency. Metadata may be wrapped in a DDI instance for purposes other than publication such as a transport format between applications. Versioning can take place prior to publication for internal control during the creation process.

From the DDI Technical Specification – Part I, Section 4.1, Lines 1665-1693:

> "Because several organizations may be involved in the creation of a set of metadata throughout the lifecycle flow the rules for maintenance, versioning, and identification must be universal. Reference to other organization's metadata is necessary for re-use and is anticipated to become very common. Accurate references require accurate versioning of the metadata content. A maintenance agency is identified by its ID as declared in a maintained or internal organization scheme. DDI will set up a registry for DDI users to provide listing of unique IDs for maintenance agencies. Individual or organizations who are not in the registry may declare their identification within the organization scheme of the DDI instance itself.
>
> Maintenance agencies own the objects they maintain and only they are allowed to change or version the objects they maintain. Other organizations may reference external items in their own schemes, but may not change those items. You can make a copy which you maintain, but once you do that, you own it!
>
> If an object changes in any way, its version must change. This may be a minor change or a major change with a major change incrementing the base number and a minor change incrementing the digits to the right of the decimal. Note that version numbers can include only "[0-9]|." but multiple decimal extensions may be used to express the level of granularity needed by the maintaining agency.
>
> Any version change at a lower level will change the version of any containing maintainable object. Typically, objects grow and are versioned as they move through the lifecycle adding or correcting content as they develop. Note that

version information is only required for published metatdata, metadata that has been packaged as a DDIInstance and intended for publication. Agencies may wish to version earlier than this to track internal metadata development. When a version is not declared it is assumed to be 1.0 by default."

A description of the content and use of versionable objects is found in DDI Technical Specification – Part II, Section 2.2 Versionable Objects, Line 499-516.

The regular expression for the allowed contents of version is ([0-9])+(\.([0-9])+)* which is interpreted as a numeric with optional decimal points, e.g., 1.0 or 15.3.1 etc.

Versioning data and metadata

The versioning practices for data and metadata within an organization or study should be consistent in terms of what results in a major version and what results in a minor version. This helps maintain clear relationships between data and metadata. The version numbers of data files and their associated metadata do not need to match. A change in one may not result in a change in the other. The relationship between the data file and its specific metadata version is found in the Physical Instance that describes the specific data file. This link goes to a specific set of Record Layouts (one or more), each of which links to a Physical Segment of a Gross Record Layout as described in a Physical Structure Scheme, which in turn links to the description of the Logical Record in a Logical Product.

Clear and consistent versioning processes for metadata and data must be adhered to. An organization may have multiple processes to reflect specific variations to support specialized content. This is a reasonable approach but should always comply with an accepted overall scheme. For example, a particular multi-national survey series may determine that a major version of the data and metadata occurs only when a country is added to the series. Any other change that would affect analysis such as a significant error in the data file or metadata would be a minor version change, while corrections of typographical errors or other changes not affecting analysis would result in a sub-minor change. Whatever the specialized case may be, the method for incrementing major and minor versions would comply with the organization's version numbering structure.

The versioning process should be clear and transparent to end users so they always know which version of the metadata and data they have acquired. The versioning practices and structure of the maintaining organization should be recorded in the description of the organization within the Organization scheme. Currently, there is no specific location or element for the information. It is strongly recommended that a Note of type "Version Structure" be created within the description of the organization. All DDI instances require a reference to the organization creating the metadata, which ensures that information on the version structure will be relayed to the end user.

"Internal versioning"

When versioning internally in the DDI document prior to publication, it is recommended that the version be defaulted to 1.0 and the version date used on all versionable elements. Note that date can be expressed as date-time according to an ISO8601 format (YYYY-MM-DDThh:mm:ss) if needed, e.g., 1982-01-05T23:05:15. This allows the group developing pre-publication documentation to track changes in specific elements without affecting the referencing structure by retaining the default value of 1.0 throughout the document. When published, no changes to the version number or version date are required.

Versioning published instances

When parallel documentation in different languages exists, processes for ensuring parallel development must be put in place, e.g., when a change in one document initiates a process of review and updating in the parallel documents. Care must be taken to ensure that both documents reflect comparable material and remain synchronized.

For published instances it is recommended that the Version Rationale element be used. This is particularly important when changes are made to sections of metadata that may affect a user's analysis or reuse.

Older versions of published metadata must be persisted (for minor changes and major changes). If the intent is to remove the metadata from public use, versioned content for the item should indicate the intentional withdrawal of the earlier content.

Versioning string

Any change in a non-versionable element results in a version change in its parent versionable element. Any version change in a versionable element results in a version change in its parent maintainable element. Any version change of a maintainable element results in a version change of its parent maintainable element (if any).

Version date

From the user's perspective there are situations in which lateBound is preferable to earlyBound and vice versa. Both lateBound and earlyBound are related to referencing rather than the construction of the version. In order to ensure clear recognition of the "most recent" version for a lateBound request, it is strongly recommended that the optional element Version Date be used for all versions after the initial version 1.0. Recommendations of when each should generally be used need to be described.

Version management

An individual or organization managing publications needs to have a well-defined system and tools in place to manage the collection and ensure its persistence. Tasks include: maintain identifiers, persist URNs, ensure integrity in identifiers, operate a public registry or publish in a public registry, and deliver the metadata on demand (reliable service). Those not equipped to handle long-term maintenance should deposit their metadata in a persistent archive. Versioning activity should be listed in the Life Cycle in the Archive module.

When an instance is published for the purpose of reuse or reference, the publisher should indicate this as a means of ensuring the user of the intent to persist the metadata. This is currently not an available feature of DDI and should be listed in the Purpose statement of the Resource Package (see discussion of isPublished in discussion section below).

Versioning local metadata enhancements

Ideally, the maintenance agency should maintain its own metadata, which is referenced from a single location. When a maintenance agency distributes copies of its metadata outside of its agency and subsequently makes a correction resulting in a version change to a portion of the metadata, the agency should provide a copy of the complete new version. This is the recommended and preferred practice. Maintenance agencies may authorize their distributors to make a specific change in the metadata under the authority of the original maintenance. The responsible process for this is provision of the structured metadata including the maintenance agency assigned version number and date. The ability of the original maintenance agency to assign authority to a distributor has implications for the distributing agency for incorporating local changes such as added value metadata. The recommended approach is to include the original distributed metadata through reference or as a separate section inline.

Local additions to the metadata should be held in separate modules (inside or outside of the instance containing the distributed metadata). This clearly differentiates material provided by each agency and allows clean updates and versioning. Ideally, the DDI instance provided by the original agency should be held as a local copy and included in the local agency's DDI instance by reference (currently this is not possible to do). The example provided shows the use of a second Study Unit in the DDI Instance to contain locally provided information. A cleaner means of relating this information is being reviewed.

Updates in the distributed data should result in a review of any locally added metadata to evaluate its accuracy and applicability in light of the change in the distributed metadata. For example, an error in the distributed metadata may be noted at a local level and entered as local information and reported to the distributing agency. In time the distributing agency verifies the error and issues a correction to the distributed metadata. This update would result in a change in the local information content, noting that this correction has now been updated in the distributed version.

2.3 Discussion

The versioning system was designed to allow users to identify which sections of a distributed document are original and where local enhancements have been made. This presupposes that DDI publishers are responsible citizens in terms of applying versioning in a consistent and compatible way.

How to determine whether a set of metadata is a new publication or a new version of an existing publication is not always clear. The issues are similar to those related to traditional book publishing, in terms of when a publication is considered a new version and when is it a new edition or a completely new publication. We were unable to answer these questions specifically in terms of a best practice. However, maintenance agencies should consider these issues internally and make decisions that reflect a consistent approach within their agencies, taking into consideration the specific nature of individual studies and study series.

This best practice includes recommendations for handling less than ideal distribution practices. Ideally the originating agency would publish the metadata and the secondary agency or depository would include it in its own version by reference. However, this best practice recognizes that many originating agencies will continue to physically distribute copies of their metadata and the recommendation includes a best practice for handling this by both the originating agencies and the agencies responsible for retaining a depository copy.

Possible future feature of DDI: A bug has been filed on the current DDI structure to add a “final” flag to differentiate between a set of metadata packaged for publication (actually published) as opposed to packaged for internal development. Discussion focused on both the desirability of such a flag and the fact that what it was really indicating was whether the packaged metadata had been officially published. A comment will be filed on this bug suggesting that a more accurate flag would be isPublished with a Boolean value of true indicating that the metadata has been officially published and that any future changes would be reflected in a version change.

Related to the discussion above, when one is creating a one-off subset of metadata for someone to accompany a data subset, is this a published version? It was determined that it doesn’t really matter because even if that person put it up and someone referenced it, it would still be the same metadata. If the person obtaining the object changes it, the maintenance agency changes and therefore it is no longer the same. If an isPublished flag is available in the future, this type of metadata file may have the recommendation to change the flag value to “false” indicating that this particular metadata package is not considered to be a published object as the intent is not to persist it or make it available for reuse or reference.

2.4 Example

Example major –minor version (ESS):

Major and minor versions vary by what is being published. However, major versions are expressed by the digits to the left of the decimal point and minor version changes are located to the right of the first decimal.

One example of a distinction between major and minor versions could be in the case of a cross-national study (e.g., European Social Survey) when a major version (position X) would refer to a change in the inclusion of countries in the data file (and consequently the metadata), and a minor version at the first minor level (y) would refer to a change in at least one of the variables in the data file (but no change in the countries contained in the file). The second minor position (z) could be used for changes in the metadata that did not involve any changes in the data file.

X.yz

1.00	Country added	2.00
2.00	Variable changed	2.10
2.10	Spelling error	2.11

Example versioning for local information

The following example shows the proper use of inclusion by reference in a case where the originating agency publishes a version and a depository wishes to create a local version adding local information and then updating it.

```
<?xml version="1.0"?>
<DDIInstance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="ddi:instance:3_0 instance.xsd"
xmlns="ddi:instance:3_0" xmlns:r="ddi:reusable:3_0"
xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:dce="ddi:dcelements:3_0"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:a="ddi:archive:3_0"
xmlns:g="ddi:group:3_0" xmlns:cm="ddi:comparative:3_0"
xmlns:c="ddi:conceptualcomponent:3_0" xmlns:d="ddi:datacollection:3_0"
xmlns:l="ddi:logicalproduct:3_0" xmlns:pd="ddi:physicaldataproduct:3_0"
xmlns:ds="ddi:dataset:3_0" xmlns:pi="ddi:physicalinstance:3_0"
xmlns:m1="ddi:physicaldataproduct/ncube/normal:3_0"
xmlns:m2="ddi:physicaldataproduct/ncube/tabular:3_0"
xmlns:m3="ddi:physicaldataproduct/ncube/inline:3_0"
xmlns:s="ddi:studyunit:3_0" xmlns:pr="ddi:profile:3_0"
isMaintainable="true" id="icpsr9999" version="1.0" versionDate="2009-01-
01" agency="icpsr.us.ddi"
urn="urn:ddi:3.0:Instance=icpsr9999:icpsr.us.ddi[1.0]">
<r:Citation>
<r:Title>Question Test Example</r:Title>
<r:Creator>Wendy L. Thomas</r:Creator>
<r:Publisher>Minnesota Population Center</r:Publisher>
```

```
<r:PublicationDate>
<r:SimpleDate>2007-02-03</r:SimpleDate>
</r:PublicationDate>
</r:Citation>
<s:StudyUnit isMaintainable="true" id="WLT_QUEST"
urn="urn:ddi:3.0:StudyUnit=WLT_QUEST:mpc.us.ddi[2.0]">
<!-- NOTE THAT IN 3.0 StudyUnit can only be listed in-line within
Instance. This has been filed as a bug. -->
<r:Citation>
<r:Title>Question Test Example</r:Title>
<!--REST OF CITATION -->
</r:Citation>
<s:Abstract isIdentifiable="true" id="ABS_1"><r:Content
xml:lang="en">Simple questionnaire using coded and numeric response with
one IfThenElse sequence</r:Content> </s:Abstract>
<r:UniverseReference
isReference="true"><r:ID>U1</r:ID></r:UniverseReference>
<s:Purpose isIdentifiable="true" id="PUR_1"><r:Content xml:lang="en">To
show it can be done</r:Content></s:Purpose>
<c:ConceptualComponent isMaintainable="true" id="CC_1">
<!-- CONCEPT SCHEME -->
<c:UniverseScheme isMaintainable="true" id="US_1">
<c:Universe isVersionable="true" id="U1">
<c:HumanReadable>Population</c:HumanReadable>
<c:SubUniverse isVersionable="true" id="U2">
<c:HumanReadable>Population 18 years of age and older</c:HumanReadable>
</c:SubUniverse>
</c:Universe>
</c:UniverseScheme>
</c:ConceptualComponent>
<!-- REMAINDER OF DOCUMENT -->
</s:StudyUnit>
<s:StudyUnit isMaintainable="true" id="ICPSR9999"
urn="urn:ddi:3.0:StudyUnit=WLT_QUEST:mpc.us.ddi[2.0]">
<!-- NOTE THAT IN 3.0 StudyUnit can only be listed in-line within
Instance. This has been filed as a bug. -->
<r:Citation>
<r:Title>Question Test Example [ICPSR 9999]</r:Title>
<!--REST OF CITATION -->
</r:Citation>
<s:Abstract isIdentifiable="true" id="ABS_1"><r:Content
xml:lang="en">ICPSR depository holding.</r:Content> </s:Abstract>
<r:UniverseReference
isReference="true"><r:URN>urn:ddi:3.0:UniverseScheme.Universe=US_1:mpc.us.
ddi[1.0].U1[1.0]</r:URN></r:UniverseReference>
<s:Purpose isIdentifiable="true" id="PUR_1"><r:Content
xml:lang="en">Provides local archival information on the depository copy
of the study unit held in this instance.</r:Content></s:Purpose>
<a:Archive isMaintainable="true" id="ARCH_ICPSR09999" version="1.0"
versionDate="2009-01-01">
<a:ArchiveSpecific>
<a:ArchiveOrganizationReference isExternal="true"
URI="urn:ddi:3.0:OrganizationScheme:Organization=icpsr.us.ddi:ORGS_MAIN[1.
0].ICPSR[1.0]" isReference="true" lateBound="true">
<r:URN
type="URN">urn:ddi:3.0:OrganizationScheme:Organization=icpsr.us.ddi:ORGS_M
AIN[1.0].ICPSR[1.0]</r:URN>
</a:ArchiveOrganizationReference>
```

```
</a:ArchiveSpecific>
<a:OrganizationScheme isMaintainable="true" id="O_1" version="1.0"
versionDate="2009-01-01">
<a:OrganizationSchemeReference isExternal="true"
URI="urn:ddi:3.0:OrganizationScheme=icpsr.us.ddi:ORGS_MAIN[1.0]"
isReference="true" lateBound="true">
<r:URN
type="URN">urn:ddi:3.0:OrganizationScheme=icpsr.us.ddi:ORGS_MAIN[1.0]</r:U
RN>
</a:OrganizationSchemeReference>
</a:OrganizationScheme>
<r:LifecycleInformation>
<r:LifecycleEvent isIdentifiable="true" id="LFC_1">
<r:EventType>Received for Deposit</r:EventType>
<r:Date><r:SimpleDate>2009-01-01</r:SimpleDate></r:Date>
<r:AgencyOrganizationReference isExternal="true"
URI="urn:ddi:3.0:OrganizationScheme:Organization=icpsr.us.ddi:ORGS_MAIN[1.
0].MPC[1.0]" isReference="true" lateBound="true">
<r:URN
type="URN">urn:ddi:3.0:OrganizationScheme:Organization=icpsr.us.ddi:ORGS_M
AIN[1.0].MPC[1.0]</r:URN>
</r:AgencyOrganizationReference>
<r:Description>Question structure obtained from Wendy Thomas at MPC for
deposit</r:Description>
</r:LifecycleEvent>
</r:LifecycleInformation>
</a:Archive>
</s:StudyUnit>
</DDIInstance>
```

3 References

3.1 Normative

[RFC2119] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

OASIS, Best Practice, http://www.oasis-open.org/committees/uddi-spec/doc/bp/uddi-spec-tc-bp-template.doc, 2003

Appendix A. Acknowledgments

The following individuals were members of the DDI Expert Workshop held 10-14 November 2008 at Schloss Dagstuhl, Leibniz Center for Informatics, in Wadern, Germany.

Nikos Askitas, Institute for the Study of Labor (IZA)

Karl Dinkelmann, University of Michigan

Michelle Edwards, University of Guelph

Janet Eisenhauer, University of Wisconsin

Jane Fry, Carleton University

Peter Granda, Inter-university Consortium for Political and Social Research (ICPSR)

Arofan Gregory, Open Data Foundation

Rob Grim, Tilburg University

Pascal Heus, Open Data Foundation

Maarten Hoogerwerf, Data Archiving and Networked Services (DANS)

Chuck Humphrey, University of Alberta

Jeremy Iverson, Algenta Technology

Jannik Vestergaard Jensen, Danish Data Archive (DDA)

Kirstine Kolsrud, Norwegian Social Science Data Services (NSD)

Stefan Kramer, Yale University

Jenny Linnerud, Statistics Norway

Hans Jørgen Marker, Danish Data Archive (DDA)

Ken Miller, United Kingdom Data Archive (UKDA)

Meinhard Moschner, GESIS - Leibniz Institute for the Social Sciences

Ron Nakao, Stanford University

463 Sigbjørn Revheim, Norwegian Social Science Data Services (NSD)

464 Wendy Thomas, University of Minnesota

465 Mary Vardigan, Inter-university Consortium for Political and Social Research (ICPSR)

466 Joachim Wackerow, GESIS - Leibniz Institute for the Social Sciences

467 Wolfgang Zenk-Möltgen, GESIS - Leibniz Institute for the Social Sciences

468

Appendix B. Revision History

Rev	Date	By Whom	What
0.9	2009-03-22	Stefan Kramer	Removed date from filename to accommodate linking. Began revision history tracking.

Appendix C. Legal Notices