

Status:	Closed
Project:	PDS4 Standards Change Control Board
Component/s:	Information Model
Affects Version/s:	1.1.0.0
Fix Version/s:	None

Type:	Enhancement / Improvement	Priority:	Normal / Non-urgent
Reporter:	Ronald Joyner	Assignee:	Emily Law
Resolution:	Approved as Is		
Labels:	Implemented_and_Released		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

target date:	30/Aug/13
Proposed Solution:	Loosen the existing constraint on the value of the attribute Idd_version_id (in the Ingest_LDD class) so that the value can include up to four component parts (ex. 1.0.1.0, 1.3.2.13).
Dependencies / Contingencies:	This SCR is related to CCB-26 , which requests that Idd_version_id be added as an optional attribute to the XML_Schema class. While it would make sense to consider the two CCBs simultaneously, neither is dependent on the other.
Additional Information:	None
Working Group:	None
Requested Changes:	Modify the PDS Information Model such that the current constraint (specified using an xml schema pattern) on the value of the Idd_version_id attribute is loosened. Any method may be used to implement this change (whether a change to the existing pattern or a Schematron rule or any other mechanism).
Impact Statement:	<p>Impact:</p> <ul style="list-style-type: none"> -- Information Model -- modify element definition <p>No Impact:</p> <ul style="list-style-type: none"> -- APG -- Concepts Document -- DPH

	<ul style="list-style-type: none"> -- External Agencies -- ISO Standards -- PDS Tools -- PDS Website -- PAG -- Standards Reference
Technical Assessment:	<p>This change request is reasonable. The following specific changes are planned for this SCR.</p> <p>(1) Remove the current validating pattern for attribute <Ingest_LDD.Idd_version_id>. This allows the requested flexibility. The LDD developer will now be responsible for their version ids.</p> <p>(2) Add an optional attribute <Idd_version_id> to the XML_Schema class. This will allow the LDDTool to indicate the version identifier of the LDD used to produce the schema.</p> <p>See CCB-26 - Add Idd_version_id to XML_Schema Class</p>
System Impact:	backwards compatible

Description

The value of the attribute "Idd_version_id" (contained in the Ingest_LDD class, which is used for submitting local data dictionaries) is currently limited to two parts, (ex., 1.0, 1.1, 2.0). This constraint is too limiting when attempting to keep local dictionary versions in sync with the version of the main PDS model / common dictionary, which has four parts (ex. 0.3.1.0, 1.0.0.0).

Since there are often dependencies between local dictionaries and the common dictionary, it can be important to be able to tie together the version of a local dictionary with the common dictionary. By allowing a four component local version identifier, the first two components of the local dictionary can be used to match the first two components of the common dictionary. This then provides an additional two components for tracking changes internal to the local dictionary that occur between major releases of the common dictionary.

Thus, a local dictionary developed with a dependency on the common dictionary version 1.0.0.0, could start off with a version of 1.0.0.0. The LDD might then have a minor change resulting in an update to version 1.0.0.1. A more significant change to the LDD might be released as version 1.0.1.0. Finally, the dictionary would need to be re-released after a release of the common dictionary as version 1.1.0.0; the LDD would also be updated to have this version. (This assumes that any changes to the common dictionary that are only reflected in the third and fourth component would not be significant enough to have an impact on any dependencies in a local dictionary.)

Comments

Comment by [Ronald Joyner](#) [01/Jul/13]

JS_xls_78

Comment by [Elizabeth Rye \(Inactive\)](#) [08/Jul/13]

Hi Folks. I perhaps didn't make it clear when I requested this change that I would like the schema to support **up to** four parts, not to mandate four parts. I have no problem if other dictionary producers wish to use only two or three part version ids.

Comment by [Richard Simpson](#) [17/Aug/13]

SR 6D.3 explicitly addresses versioning only for products, collections, and bundles, which are limited to the m.n format. DDWG discussed the format for other types of versioning but postponed a decision. In the meantime ad hoc 'standards' have been implemented in various corners of PDS4 for other version_id applications — such as use of the m.n.o format for the Standards Document (which is 4.0.8 in some places and 1.0.0 in others) and the m.n.o.p format for PDS4 itself (which is currently 1.0.0.0). [CCB-8](#) proposes formalizing (up to) m.n.o.p for Idd_version_id so that local dictionary versions can be "synchronized" with common dictionary versions. But only the 'm' and 'n' components will be synchronized.

Is there any reason to force synchronization (this seems to add work for local DD managers, who might never want more than a version 1.0 of their own DDs)? If there is a reason to synchronize, shouldn't we document the relationship (by changing the definition of Idd_version_id)? And wouldn't it be better to address these specialized versioning needs as a group rather than piecemeal by solving the problem only for Idd_version_id at this time?

Comment by [Elizabeth Rye \(Inactive\)](#) [19/Aug/13]

As I have pointed out numerous times, I am not trying to force anyone to do anything. I am attempting to *permit* those of us who want to provide synchronization among dictionaries to do it. This is why I keep stating that I want the constraint on the form of the version id loosened, not changed to my personal constraints. This will enable synchronization on my part, without forcing a change on anyone else.

Comment by [Ronald Joyner](#) [20/Aug/13]

Summary posted on : <http://http://pds-engineering.jpl.nasa.gov/pds2010/ccb/2013-08-20/Minutes-08202013.pdf>

Comments:

(1) Discussion revolved around product ids remaining 2 digit (M.m) and schemas, local data dictionaries, etc. allowing the 4 digit (M.M.m.m) versioning currently employed by the Information Model.

Generated at Wed Mar 15 16:21:03 PDT 2023 by Vivian Tang using Jira 8.22.6#822006-sha1:a60819604027c401cc97bed69f4574413f3aa3b8.