

Backward Compatible Changes CCB Discussion

October 2013

Goal

- Ensure software tools can work across minor version changes of the IM.
- Provide time to upgrade software tools when capabilities are being phased out.

PDS4 Considerations

- XML schema are derived from the Information Model
- Software in PDS4 is dependent on XML schema
 - It is assumed that software will encompass software from across EN, nodes, international community, etc
- Changes to IM and XML schema can cause software to not be backward compatible

Changing the Information Model

- The Information Model uses the ISO/ISC 11179 standard to describe the classes and attributes in the IM
- The standard provides a mechanism for indicating the status of any item
 - For example, active vs deprecated
- Items in the IM are never removed. Their status is updated to reflect their state.
 - This is useful for tracking changes

Phasing Out Changes

- The PDS4 IM using ISO/IEC 11179 allows for deleted items to be marked as “deprecated” so as to phase the removal over time
 - This allows for software to continue to work with newer versions
 - Similar to what Java and other computer languages do
- PDS can decide “when” to drop items and warn users/software developers to upgrade

Deprecation

- Deprecation is a status applied to a computer software feature, characteristic, or practice indicating it should be avoided, typically because of it being superseded. The term is also sometimes used for a feature, design, or practice that is permitted but no longer recommended in other areas, such as hardware design or compliance to building codes.
- While a deprecated software feature remains in the software its use may raise warning messages recommending alternative practices; deprecated status may also indicate that feature will be removed in the future. Features are deprecated rather than immediately removed to provide backward compatibility and give programmers time to bring affected code into compliance with the new standard.

<http://en.wikipedia.org/wiki/Deprecation>

Major vs Minor Versions

- The PDS4 IM versioning scheme allows for indication of major vs minor releases
- *Major* version releases of the IM refer to releases that do affect backward compatibility of software tools
- *Minor* version releases of the IM refer to releases that do NOT affect backward compatibility of software tools

Current Practice for Removing Items

- The current practice has been to carry items marked to be deleted as “deprecated” in the XML Schema and documentation
 - Warnings will be raised during validation that the capability will not be supported in future versions of PDS4 (just like a compiler)
 - PDS doesn’t have to rush to push out major version changes of the IM

Optional vs Required

- Adding optional items does not affect existing software tools
 - It is therefore Backward Compatible
- Adding required items does affect existing software tools
 - It is therefore NOT Backward Compatible
- Therefore, as a rule, ...
 - Optional items can be added across minor versions
 - Required items can only be added across major versions
 - Optional items can be converted to required items when going to a major version

CCB and MC Considerations

- The CCB can approve changes that remove items and still ensure compatibility of software tools across minor versions
- PDS can decide when to go to major versions to
 - drop the items from the XML Schema
 - They will still exist in the revision history of the IM
 - convert optional items to required
- This will allow software tools to continue to work across minor versions

The Deprecation Procedure

- The Deprecation Procedure lays out the scenarios for
 - Removing an item from the IM
 - Adding an optional vs required item... across major and minor releases
- Recommendation is to capture as a CCB procedure