

## WG13 Issues - CIM Issues #5111

### Versioning of CIM packages

10/20/2021 01:41 PM - Chavdar Ivanov

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Target version:</b>	
<b>Author/Contact Info:</b> Chavdar Ivanov	<b>Standard(s):</b> 61970-301
<b>Base Release:</b>	<b>Version:</b> CIM18
<b>Solution to be Applied To:</b> CIM18v03	<b>Clause:</b>
<b>Solution Version:</b> CIM18v03	<b>Sub-Clause:</b>
<b>Solution Applied By:</b> Chavdar Ivanov	<b>Paragraph:</b>
<b>Completion Date:</b> 02/11/2023	<b>Table:</b>
<b>CIM Keywords:</b>	<b>Originally Closed in Version:</b>
<b>Breaking Change:</b> No	<b>Origination Date:</b>
<b>Breaking Change Description:</b>	<b>Origination ID:</b>
<b>CIM Impacted Groups:</b> WG13	<b>Originally Assigned To:</b> Yang Feng
<b>Requestor:</b>	
<b>Description</b> <p>CIM is growing and the efforts on maintenance, model management, structured development, managing dependencies, etc are increasing. There is versioning of the 3 main packages 61970, 61968 and 62325. There are dependency diagrams on high level. However there is a need to version subpackages, e.g. Base, Dynamics, Wires, Core, etc. This will allow the following</p> <ul style="list-style-type: none"><li>- provide a clear indication which package is changes in a given release. For instance it could well happen that Core does not changes for StateVariables and users will clearly know that from the version of the package</li><li>- facilitate documentation handling</li><li>- allows better understanding which profiles are affected if a package is modified. Dependency diagrams will also help here.</li><li>- enable an approach that deal with vocabularies related to sub packages, e.g. Core vocabulary, Wires vocabulary, etc</li></ul> <p>...</p> <p>Issue was discussed multiple times in WG13 and members find versioning of packages a good approach. There are also options to version on class level, but this is considered too much.</p>	
<b>Proposed Solution</b> <p>There are at least 2 options on how to add the version</p> <ul style="list-style-type: none"><li>- add a version class in each package</li><li>- add a tagged value on the package. This could be a number xx.yy.zz to indicate major.minor.revision numbers, but it could also be URI <a href="http://iec.ch/TC57/CIM/{package name}/xx.yy">http://iec.ch/TC57/CIM/{package name}/xx.yy</a> where xx is major version and yy is minor version, e.g. <a href="http://iec.ch/TC57/CIM/Core/17.40">http://iec.ch/TC57/CIM/Core/17.40</a></li></ul> <p>Such solution can already be applied in the Dynamics package and subpackages as it is in CDV stage. The version information will then be printed in 61970-302:Ed2. 301 and other standard will follow according to their release plan</p>	
<b>Decision</b> <p>UTF13 decided to apply the versioning to Base and Dynamics packages and subpackages starting from CIM18 all URIs will get v 1.0 The domain is uca, e.g. <a href="http://ucaiu.org/CIM/Dynamics/1.0">http://ucaiu.org/CIM/Dynamics/1.0</a></p> <p>Version and URI are only changed in official releases at least in the beginning of development of the package. For instance working drafts may not get newer versions.</p>	
<b>Release Notes</b> <p>Two tag values were added to the UML uri which has the URI of the package, e.g. <a href="http://ucaiu.org/CIM/Dynamics/1.0">http://ucaiu.org/CIM/Dynamics/1.0</a> version which is the version of the package, e.g. 1.0.0</p>	

---

## History

---

### #1 - 10/20/2021 03:11 PM - Todd Viegut

Chavdar, a couple of questions. For the proposed solution options you have described i.e. xx.yy.zz or <http://iec.ch/TC57/CIM/{package name}/xx.yy>. I would like to simply formalize what types of UML changes fall into each component of the semantic version. I'd like to additionally document this in the CIM modeling guidelines document.

xx -> As part of semantic versioning this would translate to breaking changes within the package such as:

- A class is removed
- A class is renamed
- An attribute is removed from an existing class
- An attribute is renamed within an existing class
- An association is removed
- An association role end is renamed

yy -> As part of semantic versioning would translate to non-breaking changes within the package such as:

- A new class is added
- A new attribute is added to an existing class
- A new association is added

zz -> As part of semantic versioning would translate to very minor non-breaking changes within the package such as:

- description of the package is updated/clarified
- description of an existing class is updated/clarified
- description of an existing attribute is updated/clarified

Additional questions/assumptions we should formalize:

- What if a package is added under an existing package (e.g. a new child package under the Base package)? Is our position that the semantic version of the Base package would or would not change to reflect this addition? Currently, I do not see a reason to change the version of the parent package (i.e. for Base in this example). Just want us all to be on the same page.
- Do we update the semantic version IF a UML diagram is updated or do we restrict to only changes to the UML proper? At first UML proper seemed reasonable but I vacillate.
- Currently for your URI-based option it seems to me we'd also want to include 'zz'. Thoughts?

Chavdar (et al.), please add/modify/clarify what I've proposed here.

### #2 - 10/20/2021 03:57 PM - Chavdar Ivanov

Todd, very good

in the xx we can add

- description is modified to a higher degree potentially changing the meaning
- constraint is added - that maybe in yy

I guess if a package is added to Base this is yy increase of the version as this should bring features. It will also depend if this package will have to do something with classes under Base. In case the versioning of Base might not be super useful - just on global level. What will be really good if the versioning of the packages down the hierarchy

I think versioning of diagrams might not bring that much

on the URI. If we want to have more stable approach we can keep the URI without the zz and have the zz in another tag value. The zz should normally be editorial changes that would not need change in implementation. So if we use the URI to actually get the vocabulary for that package the <http://iec.ch/TC57/CIM/Core/17.40> will always refer to the latest revision but the URI would only have xx.yy. In fact if we want to really publish this we can use the ucaui domain instead of iec.ch. Basically an RDF export of that vocabulary can be published there

### #3 - 10/20/2021 04:12 PM - Todd Viegut

Chavdar, excellent! Thanks for your feedback as that what I was hoping for. Your additional suggestions make solid sense. Also your clarifications on the URI-base approach is helpful and we will be able to discuss better when Eric and I set up the meeting with the Becky and Henry. Anything else you end up thinking of please add to this thread. :)

### #4 - 11/24/2021 10:21 AM - Eric Stephan

- Status changed from New to Open

This was approved at the October 13, 2021 WG13 meeting.

### #5 - 11/24/2021 10:58 AM - Eric Stephan

- Status changed from Open to Review

### #6 - 02/04/2022 02:59 PM - Eric Stephan

From WG13 November 24, 2022 Meeting:

- o Proposed using existing package properties (uri, version and package) for version.
- o Proposed: start at 1.0 (major.minor version)
- o Applied to the base CIM17v40 version that we used as baseline for CIM18 going forward.

Discussion on xx.yy.zz release format from the Issue 5111:

xx > ~~As part of semantic versioning this would translate to breaking changes within the package such as:~~

- A class is removed
- A class is renamed
- An attribute is removed from an existing class
- An attribute is renamed within an existing class
- An association is removed
- An association role end is renamed

yy > ~~As part of semantic versioning would translate to non-breaking changes within the package such as:~~

- A new class is added
- A new attribute is added to an existing class
- A new association is added
- description of the package is updated/clarified
- description of an existing class is updated/clarified
- description of an existing attribute is updated/clarified

zz -> As part of semantic versioning would translate to very minor non-breaking changes within the package such as:

- syntax generation
- spelling mistakes, editorial fixes etc

Additional questions/assumptions we should formalize:

- What if a package is added under an existing package (e.g. a new child package under the Base package)?
- The packages will follow the same rules as the classes. (above)

Is our position that the semantic version of the Base package would or would not change to reflect this addition? Currently, I do not see a reason to change the version of the parent package (i.e. for Base in this example). Just want us all to be on the same page.

- The packages will follow the same rules as the classes. (above)

- Do we update the semantic version IF a UML diagram is updated or do we restrict to only changes to the UML proper? At first UML proper seemed reasonable but I vacillate.

- The assumption is that the current version of the 301v7.1 is 1.0.
- In the 302 ed2 we should put in 1.0 for the package version.
- If we had a new amendment to the 301 (prior to the CIM18) we would add the versioning to that document

#### #7 - 02/04/2022 03:00 PM - Eric Stephan

I meant November 24, 2021 in the last note

#### #8 - 09/22/2022 02:23 PM - Eric Stephan

- Status changed from Review to In Progress
- Originally Assigned To set to Yang Feng

#### #9 - 02/07/2023 02:24 PM - Todd Viegut

- Version changed from CIM17 to CIM18

#### #10 - 02/11/2023 12:08 AM - Chavdar Ivanov

- Solution to be Applied To set to CIM18v03
- Solution Version set to CIM18v03
- Solution Applied By set to Chavdar Ivanov
- Completion Date set to 02/11/2023
- Decision updated
- Release Notes updated

#### #11 - 02/11/2023 12:32 AM - Chavdar Ivanov

- Status changed from In Progress to Closed