IUCr forums

Discussions on IUCr projects and activities

Discussion of namespace conventions for community-developed CIF dictionaries

Snapshot of recent discussion thread, 10 August 2013

Mixed-discipline data CIFs

by **jamesrhester** » Thu Jul 18, 2013 10:13 am

This post continues on from a discussion on COMCIFS in which it was proposed that mixed-discipline data CIFS be allowed. Below I rehearse the arguments against allowing tags from different disciplines to be used in a single datafile.

(1) *By definition,* a discipline does not coordinate its namespace with another discipline. How to disambiguate datanames *within* a single discipline is *not* the object of this proposal.

(2) Dictionaries from separate disciplines that were non-conflicting when a CIF file was written may have come into conflict subsequently, as separate disciplines are by definition uncoordinated. A CIF file would therefore need to provide detailed dictionary audit information, so that CIF file reading programs can check historic dictionary versions against the versions that the CIF file reading software was written against. Compliant CIF-reading software would need to have the capability of performing such checks in the event that it was faced with such files. This implies a significant extra burden on both CIF reading and writing programs.

(3) Under the current proposal it is possible to incorporate ontologies from other disciplines: each discipline officially incorporates a dictionary from some other domain into its own domain, adopting whatever policies it wishes to ensure that clashes within its own domain do not ensue (e.g. prepending a prefix referring to the originating domain).

(4) Alternatively, DDLm-style save frames may draw datanames from a separate ontology: see [url = viewtopic.php?f=28&t=315#p507]this post [/url] for a discussion.

(5) A mixed-ontology data file is only useful compared to a single-ontology datafile if the CIF reading program understands the meanings of the datanames in both ontologies. Absent a specific agreement between sender and receiver (in which case they are free to agree whatever they want to get the job done, but can't expect the rest of the world to understand) this implies the existence of a standard against which both sender and receiver interpret the CIF. Use of a mixed ontology therefore requires standardisation effort, and so approaches (3) and (4) do not represent additional work.

So: given the small amount of work required for approaches (3) and (4) for accessing external ontologies, and the complexity and potential for error of approach (2), the recommendation is that tags from multiple disciplines are never mixed in a single data CIF.

Top

Re: Mixed-discipline data CIFs

by **yayahjb** » Thu Jul 18, 2013 12:45 pm

The heart of the argument against mixed discipline CIFs is from James Hester

on 24 December 2012:

"Implicit in my definition of a 'domain' is the possibility that non-conflicting dictionaries from different domains could be arbitrarily mixed to create ad-hoc domains. This is not desirable, because dictionaries that were non-conflicting when a CIF file was written may have come into conflict subsequently, as separate domains are by definition uncoordinated. Considerable detailed checking of dictionary versions would therefore need to be performed by software in order to disambiguate such files, defeating the purpose of the domain idea. The correct way of mixing dictionaries from multiple domains would be for each domain to incorporate a dictionary from some other domain into its own domain, adopting whatever policies it wishes to ensure that clashes within its own domain do not ensue (e.g. prepending a prefix referring to the originating domain). Note that we are therefore actively discouraging mixing of datanames from different domains, and my comment above about some mixing being possible using looped '_audit.domain' datanames was a mistake."

This argument, essentially the French Academy position, is unrealistic about how most people, including the French themselves, handle languages -- they use the words they know in whatever context works for them. Thus a French speaker is very likely to use the word "computer" even though the French academy disapproves and has provided an alternate word in the French dictionary. This sort of ad-hoc mixing is not a problem for the speaker of the moment, not for their immediately listeners, who presumably know what is being discussed from context, but becomes a serious problem when spoken language enters the literature and is, essentially, archived. Then, in the future, context may get lost or misunderstood. In literature we have adopted conventions to distinguish "wrong" language words mixed into a main flow of a primary language, e.g. by italicising them or quoting them. Even in English, which is has a very uncontrolled vocabulary, we commonly italicize embedded Latin words and phrases.

We face the same problem with CIFs. As James has noted, what is a non-conflicting mix today may be a conflict tomorrow, but people being what that are we can be certain mixing will happen even in science, where despite a world-wide effort at going metric, people still mix feet and meters, minims (in three different sizes) and milliliters. Our best hope of ensuring that CIFs in archives are readable is to have available clear markers of the namespaces from which tags comes associated as closely as possible with them. One such possibility is to allow for an optional C++-style namespace prefix, such as IUCR::, so that, when archiving a mixed-used CIF or, better, when creating it, alien tags from the "wrong" disciplines can be clearly identified, as in

_IUCR::atom_site.occupancy

which, for a crystallographer living in the IUCr world would have precisely the same meaning as

_atom_site.occupancy

but which might have some different meaning in some other discipline that decides to extend or revise the meaning of occupancy (e.g. allowing negative values or values that add up to more than 1, or to make it absolute, rather than relative).

Top

Re: Mixed-discipline data CIFs

by **jcbollinger** » Thu Jul 18, 2013 4:37 pm

Surely the questions of whether or how to support mixed-domain CIFs are subordinate to the main thrust of the original proposal to establish a framework for multiple domains in the first place. I submit that if we see value in CIF being adopted outside the crystallography arena, then it behooves us to establish a framework that allows appropriate bodies in other disciplines to adopt it with as much independence as possible. That is the crux of the original proposal, and inasmuch as I *would* like to see CIF expand to other disciplines, I think it is worth doing even without providing for mixed-domain CIFs.

I am prepared to reserve judgement on whether provisions for mixed-domain CIFs should be adopted in the future, but any such provisions would be moot now and as long as there are no other domains to worry about. Given how much more disruptive such provisions would be than James's base proposal, I am inclined to defer that decision and any associated specifications until such time as they are actually warranted.

If there is sufficient will to get ahead of the game on this issue, however, then I could see collaborating on a white paper describing how support for mixed-domain CIFs *could* be implemented in the future. The features specified by James's proposal would still be relevant under such a scheme, in that they would provide a facility analogous to XML's default namespaces.

John
Top

Re: Mixed-discipline data CIFs

by **yayahjb** » Fri Jul 19, 2013 12:15 am

Dear Colleagues,

Failing to address this issue up front encourages idiosyncratic solutions to a common problem and multiplies the future software impact. Adopting one clean solution now will save us a headache in the future. I cannot support the current proposal with what is to me, an obvious, major gap.

The cost of the prefix solution for pure IUCr CIF files is simple: parse, recognize and discard IUCR:: prefixes. Life only becomes interesting when the first mixed use of tags from some other discipline are encountered in an otherwise pure IUCr CIF. If we have not adopted the :: prefix convention, we will have to come up with ad hoc solutions then. If we have adopted the prefix convention we would have a fairly simple path forward without having to take time out then for exchanges of white papers.

It something is worth doing, it is worth doing right. Nothing stops somebody from adopting CIF for another discipline right now, no matter what we do. The best we can do is to suggest practices that will reduce the chances of us tripping over one another.

Collegially,
Herbert
Top

Re: Mixed-discipline data CIFs

by **jamesrhester** » Fri Jul 19, 2013 6:43 am

If I understand Herbert's position correctly, the use of mixed-discipline CIFs is inevitable, and so we should make sure that we can handle this use-case now rather than when it is too late. I suspect that Herbert is correct in this prediction.

The single problem presented by mixed-discipline CIFs is that of disambiguating potentially identical datanames. I can see three solutions coming out of the discussion so far:

Solution (1): Opt for discipline-specific dataname prefixes as described by Herbert. This undoubtedly solves the problem, at the cost of a massive hit to backwards compatibility - that is, for every current dataname, that dataname with 'IUCr' prefixed also becomes valid, but no currently available software will recognise such tags. For this reason I originally rejected this option when searching for a solution.

Solution (1a): In addition to (1), define _audit.discipline such that, when and only when it is looped with two or more disciplines, all datanames must take the prefix of the appropriate domain. If we assume the default value of _audit.discipline to be 'IUCr', then we have preserved backwards compatibility. I would further advise programmers that they are not required to accept mixed-discipline CIFs. This would minimise the compliance burden. I dislike this approach as the value of a single tag is now determining the interpretation and correctness of the whole datablock but I'd be interested to hear what others feel about this.

Solution (2): Stipulate that if _audit.discipline is looped over two or more disciplines, the _audit category must list the

dictionaries used from each discipline. A 'sort order' is provided in the _audit.discipline loop to help in resolving dataname clashes. It becomes impossible to choose different precedences for different dataname clashes, however, so this solution is only partial.

Solution (3): Save frames as per the latest STAR paper and DDLm example dictionaries are used to encapsulate datanames from different disciplines (this is legal according to the current _audit.discipline proposal). These can then be referred to using save frame references in the main datablock. I believe that this solution is the most elegant. However, COMCIFS have yet to address themselves to whether or not save frames will be acceptable in DDL3 files.

Are there any other ideas? Let's discuss the pros and cons of the above solutions.
Top

Re: Mixed-discipline data CIFs

by **jamesrhester** » Fri Jul 19, 2013 7:10 am

Just to pick up on one of Herb's points:

> *yayahjb wrote:*
> The cost of the prefix solution for pure IUCr CIF files is simple: parse, recognize and discard IUCR:: prefixes. Life only becomes interesting when the first mixed use of tags from some other discipline are encountered in an otherwise pure IUCr CIF. If we have not adopted the :: prefix convention, we will have to come up with ad hoc solutions then. If we have adopted the prefix convention we would have a fairly simple path forward without having to take time out then for exchanges of white papers.

The cost of the prefix solution is not trivial in terms of old software, which will cease to work with otherwise perfectly legitimate files full of IUCr-prefixed datanames. If we stipulate that all other disciplines must use a prefix (but not us), then we have gone in a circle and are back to the IUCr handing out dataname prefixes. The cost for new programs is also not that trivial: programmers must now be sure to check for two or more alternatives for every dataname (what if a different discipline quite reasonably defines an identical dataname for an identical concept?), and decide how to output datanames (same as input? with prefix? without prefix?) and probably therefore track the original name as well as the internally-used name.

This additional burden on everybody does not appear commensurate with the expected relative rareness of mixed-discipline CIFs. We should instead strive to minimise the impact on the majority who will not need mixed-discipline CIFs, while making them possible.
Top

Re: Mixed-discipline data CIFs

by **jcbollinger** » Fri Jul 19, 2013 3:11 pm

> *yayahjb wrote:* Failing to address this issue up front encourages idiosyncratic solutions to a common problem and multiplies the future software impact. Adopting one clean solution now will save us a headache in the future. I cannot support the current proposal with what is to me, an obvious, major gap.

I can't agree with that as written. Choosing to not address this issue up front may encourage idiosyncratic solutions to a *possible*, *future* problem, but no such problem can exist until some time after a multi-domain framework is established, and it may be that none ever materializes. As James describes, a solution based on domain prefixes would break all existing CIF software. That might nevertheless be the best solution overall, but I am not prepared to accept definite, significant headache now in order to avoid the mere possibility of uncertain future headache.

Still, if we conclude that domain prefixes would be the best solution for mixed-domain CIFs in the event that a bona fide need for those emerges, then I think my suggestion to draft specifications but defer their adoption into CIF would address

the concern about divergent approaches arising. We would effectively be describing how it should be done if it ever needed to be done, without (at this time) requiring any CIF software to recognize the convention.

On the third hand, there is potential advantage to letting idiosyncratic solutions arise in the community. As experienced and savvy as this group may (or may not) be, there are only a few of us. Though I have no doubt we could devise a good solution, I am not at all confident that someone else would not devise a better one if given the motivation and opportunity.

Furthermore, we should be clear about what the controversy is. The proposal as currently formulated *does* provide for mixed-domain CIFs, so that is not the issue. What it does not provide for is mixing data names from different domains in the same save frame or directly in the same data block. Even if we postulate that mixed-domain CIFs are a foreseeable future need, it is not clear to me that the proposal's existing support for it would be inadequate. Certainly it is unclear that the existing support would be *so* inadequate as to induce people to jump directly to devising idiosyncratic alternatives.

> *yayahjb wrote:* The cost of the prefix solution for pure IUCr CIF files is simple: parse, recognize and discard IUCR:: prefixes. Life only becomes interesting when the first mixed use of tags from some other discipline are encountered in an otherwise pure IUCr CIF. If we have not adopted the :: prefix convention, we will have to come up with ad hoc solutions then. If we have adopted the prefix convention we would have a fairly simple path forward without having to take time out then for exchanges of white papers.

My musing about allowing a solution to arise in the wild notwithstanding, I am hardly suggesting waiting until such time to decide how to address the problem (of names from multiple domains appearing in the same block or frame). I am suggesting waiting until at least the time that such a need is actually realized to require compliant CIF software to support a solution that we choose now.

> *yayahjb wrote:* It something is worth doing, it is worth doing right. Nothing stops somebody from adopting CIF for another discipline right now, no matter what we do. The best we can do is to suggest practices that will reduce the chances of us tripping over one another.

I submit that the proposal as currently written provides an excellent mechanism for minimizing the chances of different domains tripping over one another. A much more reliable one, in fact, than anything that allows names from different domains to appear in the same block or frame. Nevertheless, considering that a future need for such mixing may be deemed inevitable by some, I will shortly describe an alternative approach that will work within the current CIF framework as amended by the current form of the proposal.

John
Top

Re: Mixed-discipline data CIFs

by **jcbollinger** » Mon Jul 29, 2013 7:47 pm

"Shortly" has become quite a lot longer than I intended, but here goes...

> *jamesrhester wrote:*The single problem presented by mixed-discipline CIFs is that of disambiguating potentially identical datanames.
>
> [...]
>
> Are there any other ideas? Let's discuss the pros and cons of the above solutions.

Solution 4: either prospectively or at need, items from foreign domains are mapped into the domain of a given block or frame by means of a physical or logical adapter dictionary. Such an adapter dictionary would reference a specific version of each

of one or more specific dictionaries in the foreign domain, and conceivably could narrow its scope to a subset of the data names therein. Any mapping from the foreign domain into the target domain could be used, but one likely implementation for the IUCr host domain would be to prepend a prefix to the foreign name, according to the current prefixing system.

Inasmuch as prefixing is a likely mechanism for mapping names, this is similar to Herbert's proposal. It has a few notable distinctions, however:

- Every block or frame uses only names from a single domain, though some of their definitions may reference foreign domains' dictionaries.
- Mapping specific definitions of foreign data names into the target domain serves to insulate one domain from the dictionary maintenance policies of others.
- **Names drawn directly from a block's or frame's target domain never carry a domain prefix**
- If desired, this could be implemented as an IUCr-domain policy, and thus merely as a model policy for other domains to consider. That would lighten the policy and implementation requirements on other domains.
- A target-domain dictionary for the foreign names provides a platform for full definitions where needed, such as when the foreign domain dictionary is written in a different DDL than those used in the IUCr domain.

Top

Re: Mixed-discipline data CIFs

by **jamesrhester** » Thu Aug 08, 2013 4:21 am

So in practice Solution 4 should work as follows: Programmer A wishing to emit CIF data files that include moon surface temperatures from the 'astronomy' discipline together with (IUCr dataname) chemical structures found on that moon would create an adapter dictionary that e.g. prefixes all astronomy datanames with '_astro', and then write the output program assuming that dictionary. Programmer B writing a program that wishes to read such files would understand the contents of the adapter dictionary, and that the '_astro' prefix is prepended to astronomy discipline datanames, and thus be able to find the temperatures.

Questions:
1. Are all CIF-writing programs and/or dictionary writers expected to understand that the _astro prefix is reserved? If so, in what way is this different to the current prefix system? Or is the method of translation of datanames specified in each adapter dictionary?
2. How are the adapter dictionaries constructed? Who does this work and makes sure that it agrees with all other adapter dictionaries that might be relevant to the same datablock?
3. Are all CIF reading programs expected to read the adapter dictionaries in order to understand how to obtain foreign discipline datanames?

As you can probably tell I don't think I've completely understood the subtleties of the proposal, so if you have a chance John could you take us through an example usage scenario?
Top

Re: Mixed-discipline data CIFs

by **yayahjb** » Thu Aug 08, 2013 1:24 pm

I do not understand how John's proposal differs from the existing prefix mechanism, that is why I proposed a different, non-conflicting, prefix mechanism, to handle the more global namespace issues. If we combine the two concepts, we would have the following tag structure in DDL2 use (which is, I think, the most complex case):

{<disciplinePrefix>::}{<disciplinespecificPrefix>}<category>.{<disciplinespecificPrefix>_}<column>

where the {} delimit an optional component, and <> delimits a syntactical token.

<disciplinePrefix> is an identifier from some master registry (which could be maintained by IUCr as a courtesy) which can optionally be used to disambiguate tags from multiple disciplines
<disciplinespecificPrefix> is an identifier from the relevant discipline's master registry of prefixes maintained and used by whatever rules that discipline is following. Brian has published the rules that the IUCr follows. Note that in the IUCr, dictionaries have to explicitly carry categories and

columns with their prefixes explicitly given, and the only way to make them truly optional is to put in the necessary aliases in the dictionaries. Other disciplines might take a more liberal approach to their prefixes, similar to the one I am proposing for <disciplinePrefix>.
Top

Re: Mixed-discipline data CIFs

by **jcbollinger** » Thu Aug 08, 2013 8:53 pm

> *jamesrhester wrote:*So in practice Solution 4 should work as follows: Programmer A wishing to emit CIF data files that include moon surface temperatures from the 'astronomy' discipline together with (IUCr dataname) chemical structures found on that moon would create an adapter dictionary that e.g. prefixes all astronomy datanames with '_astro', and then write the output program assuming that dictionary. Programmer B writing a program that wishes to read such files would understand the contents of the adapter dictionary, and that the '_astro' prefix is prepended to astronomy discipline datanames, and thus be able to find the temperatures.

The CIF producer might or might not need to create an adapter dictionary himself. At IUCr or interested-party option, supposing we agree to allow it, some adapter dictionaries could be centrally curated just as other ancillary dictionaries already are now. Still, if the CIF producer did need to create an adapter dictionary himself, one of the alternatives that could be supported is to do so parametrically, rather than as separate, literal dictionary. More on that below.

On the consumer side there are two main cases:

- The CIF consumer sees the data name without particular prior knowledge, and wants to know what it means, how to validate it, or whatever. In this case it simply looks up the name in the appropriate local-domain dictionary. Under some circumstances it might be referred from there to a definition (likely of a different name) in some other dictionary. The correct local-domain dictionary is chosen by the mechanisms already defined, or based on a dictionary specification included in CIF form in the file itself.
- The CIF consumer is looking for a particular datum. That item perforce belongs to the local domain, because there is no provision for direct usage of foreign items. For the consumer to have any reason to expect the item might be present, it must either be relying on a centrally curated adapter dictionary, or it must have a special arrangement with the CIF producer. Either way, it knows what name to look for -- it does not have to try to deduce what local name is mapped to a foreign name of interest.

Nevertheless, this proposal can and should provide for sufficient data to be encoded so that a general CIF consumer can recognize which data names are associated with foreign-domain items, including which foreign domain, which dictionary, and which foreign item name. That is sufficient for a program to locate in a given CIF the item associated with a particular foreign definition, even though it should not usually be necessary to do so.

There are several non-exclusive possibilities for adapter dictionaries that IUCr might choose to support, including:

- IUCr or any third party creates adapter dictionaries for (a subset of) other domains in complete, standalone form, reserving name prefixes with IUCr for the purpose just like might be done for any other ancillary dictionary in the IUCr domain.
- IUCr or any third party creates adapter dictionaries for (a subset of) other domains in an indirect form primarily referencing and relying on a foreign dictionary, reserving name prefixes with IUCr for the purpose just like might be done for any other ancillary dictionary in the IUCr domain.
- Individual CIF producers use **[local]** names, either with a formal dictionary of their own creation, or else simply with an implicit dictionary.
- Individual CIF producers express per-CIF adapter dictionaries in a virtual, parameterized, indirect form described by data items included in the CIF and referencing a foreign dictionary (analogous to XML's provisions for binding namespace prefixes to schemas).

> *jamesrhester wrote:*Questions:
> 1. Are all CIF-writing programs and/or dictionary writers expected to understand that the _astro prefix is reserved? If so, in what way is this different to the current prefix system? Or is the method of translation of datanames specified in each adapter dictionary?

For formally curated adapter dictionaries, the prefix part is not different or separate from the current prefix system. That's part of the point: we use a mechanism we have already established instead of creating a new one. I don't see why it would be useful or desirable to create a separate, parallel, mechanism.

For adaptation via [local] data names, all the same provisions apply as for any other such names. Again, nothing fundamentally new is required.

If support for virtual adapter dictionaries were adopted, then the parameters describing those dictionaries would need to include everything necessary to map local data names to those defined in the referenced foreign dictionary. For per-CIF virtual dictionaries it would not be essential to use reserved prefixes. CIF processors wishing to validate data items against such adapter dictionaries would need to be able to understand the indirection involved in order to retrieve the foreign dictionary and map data names into it.

Although the name-mapping mechanism would be expressed either implicitly or explicitly in adapter dictionaries, it's not directly relevant to most use cases, as discussed above. CIF-consuming programs generally should not need to know about it, though the parser might need the information internally, especially if it validates.

> *jamesrhester wrote:*2. How are the adapter dictionaries constructed? Who does this work and makes sure that it agrees with all other adapter dictionaries that might be relevant to the same datablock?

Whoever wants badly enough to use foreign items will create and maintain appropriate adapter dictionaries, just as dictionary creation works now. However, where complete and direct item-by-item correlation is what is wanted, adapter dictionaries could easily be created via a program, provided only that target dictionaries are written in DDLs that we are prepared to read. If a given target dictionary is *not* written in a DDL that we are prepared to read then some sort of standalone adapter dictionary is required no matter what.

I'm not sure what concern you see with multiple adapter dictionaries relevant to the same block. At least, I don't see what new problem adapter dictionaries would present in this regard.

> *jamesrhester wrote:*3. Are all CIF reading programs expected to read the adapter dictionaries in order to understand how to obtain foreign discipline datanames?

Foreign-domain data names are not directly relevant to most CIF consumers under this proposal. That, too, is part of the point. It is the intention that adapter dictionaries should (or at least could) contain all data needed to validate CIFs using the datanames defined therein, either directly or indirectly, just like any other CIF dictionary. Being defined in an adapter dictionary inherently makes data names local, even if they are based on or refer directly to items in a dictionary belonging to a different domain.

If a mechanism for indirect adapter dictionaries were accepted, such as either of the two described above, then CIF readers wishing to validate items drawn indirectly from foreign dictionaries would need to obtain the foreign dictionary and use the defined name mapping to validate local items against it. That information would be carried by the adapter, though programs could conceivably shortcut for centrally-curated adapters. Whether any given program provides such a facility is at the discretion of the author.

> *jamesrhester wrote:*As you can probably tell I don't think I've completely understood the subtleties of the proposal, so if you have a chance John could you take us through an example usage scenario?

Inasmuch as you may be saying that you perceive flaws in the proposal, please go ahead and lay them out. Though the idea seems good to me at the moment, maybe I'm blind to some egregious shortcoming. Until such time as I am persuaded that there is a better alternative, however, I'll answer criticisms as best I can.

The main potential subtlety I have recognized is already discussed above: that computing the data name by which to look up a foreign-domain item is generally *not* the problem at hand (though the proposal can nevertheless provide the data needed to do that).

Otherwise, perhaps you are looking for subtleties that aren't there. Certainly, some of the key alternatives for adapter dictionaries would allow nominally interdomain CIFs to be created and used exactly as any CIF is today. Consumers would not necessarily even need to recognize their cross-domain nature. That's one of the objectives of the proposal; to exercise the simplest options, we don't need anything new.

The major underlying principles are these:

- Because domains are fundamentally organizational units for dictionaries, cross-domain relationships should be handled at dictionary level. Therefore, this is a proposal for binding foreign items into the local domain via dictionaries, not for (directly) referencing foreign-domain items.
- As much as possible, we should use mechanisms and facilities that already exist instead of creating new ones.
- Valid uses of the cross-domain support should produce the minimum possible breakage of current CIF software (and in particular, the system avoid defining standard aliases for local-domain data names).

Additionally, it is desirable that

- domains have as much independence as possible, including to control which foreign data items, if any, they want to formally support in their CIFs, and that
- the system not depend on foreign domains to use the same DDLs that the local domain does (though doing so may facilitate supporting inter-domain CIFs).

The very simplest alternative would produce adapter dictionaries containing definitions such as this (please forgive my uncouth use of DDL1):

Code: Select all

```
data_astro_lunar_surface_temp
    _name                       '_astro_lunar_surface_temp'
    _category                     astro_lunar
    _type                         numb
    _list                         no
    _enumeration_range            0.0:
    _definition
;               The lunar surface temperature, expressed in Kelvin.
                Drawn from item _lunar_surface_temp of IAU-domain core
                dictionary astro-cif.dic, version 1.0.1.
;
```

At the other extreme, use of a per-cif virtual adapter dictionary might look something like this:

Code: Select all

```
data_example
loop_
  _foreign.dictionary_prefix
  _foreign.domain
  _foreign.dictionary_uri
  _foreign.dictionary_expected_version
  _foreign.name_mapping
  'astro' 'IAU' 'http://www.iau.org/cif/astro-cif.dic' '1.0.1' 'remove-prefix'

_astro_lunar.surface_temp 390(2)
_cell.length_a 12.345(6)
```

Or even like this:

Code: Select all

```
data_example
loop_
  _foreign.dictionary_prefix
  _foreign.domain
  _foreign.dictionary_uri
  _foreign.dictionary_expected_version
  _foreign.name_mapping
  'astro::' 'IAU' 'http://www.iau.org/cif/astro-cif.dic' '1.0.1' 'remove-prefix'

_astro::_lunar.surface_temp 390(2)
_cell.length_a 12.345(6)
```

There are a variety of intermediate possibilities, all of them mutually compatible in that different alternatives could be used for different foreign dictionaries, or even for the same foreign dictionary in different cifs.

Top