

Configuration Database - Feature #1942

Analysis tag in the CDB

04 October 2017 11:00 - Martyniak, Janusz

Status:	Open	Start date:	04 October 2017
Priority:	Normal	Due date:	
Assignee:	Martyniak, Janusz	% Done:	0%
Category:	Database	Estimated time:	0.00 hour
Target version:			

Description

Hello,
I'm opening this feature request by copying various mails about this. This is a part of discussion after we realised that it was not practical/possible to add the tag from within the run control.

Here is my summary a mailed to people interested on July 24th at 16:10 :

Re: Analysis tag in the CDB. A summary of our discussion on Friday

...
...

Wording: sub tags a.k.a dependent tags == BL, CC, ABS tag names which shifter select and RC runs out to the CDB

I design and write a new CDB module Web Service to store an analysis tag with it's sub-tags and the PDF place holder as discussed on Fri and described in this email thread.

This is to be run from the Control Room by experts. Let me stress it again, it is just a set of templates, not related to any runs.

Now:

We write a simple Python script based on existing python APIs (pseudocode)

```
def setAnalysisTagForRun(runnumber):

    bl_tagname = get_beamline_tag_name_for_run(runnumber) # a trivial wrapper around get_beamline
_for_run(runnumber) existing Python API call. We just need a name
    cc_tagname = # ditto for the CC
    abs_tagname = # ditto fro the absorber

# the meat:

    try:
        set_analysis_tag_for_run(runnumber, bl_tagname, cc_tagname, abs_tagname) # has to be writt
en as a python binding to the new CDB module WebService
        """
        stores a relation between the run number and the analysis tag retrieved from the expert a
nalysis tag table which corresponds to the sub tags in the
        argument list. The analysis tag name can be easily retrieved from the CDB by a remote cal
l, and once known used to create a relationship:
        (analysis_tag_name, runnumber). Job done.
        """
    except:
        """
        No analysis tag which corresponds to the set of sub tags listed in the argument.
        Possible cause:
        - shifts have selected a wrong sub-tag combination from the RC
        - experts have not stored the required analysis tag which corresponds to that combination.
        """
        call_Melissa()

#### end of script
```

The script is to be run after run has been taken, once a day or so for all runs (we can get a list of runs automatically).

It would do a job, although is not perfect.

Thoughts ?

J.

History

#1 - 04 October 2017 11:42 - Martyniak, Janusz

Hi,

Current progress (all in the test CDB, no preprod prod etc)

The analysis tag table has been created. The schema is:

```
cdb=> \d analysistag
```

Column	Type	Modifiers
name	character(40)	
blname	character(40)	
ccname	character(40)	
absname	character(40)	
version	integer	
pdfref	character varying(40)	default 'undef'::character varying
creationtime	timestamp without time zone	
validuntiltime	timestamp without time zone	

This table design follows a well known MICE tag tables. A corresponding service with relevant operations has been implemented. Also a python API has been created. The functions are:

```
class AnalysisTagSuperMouse(cdb._analysistag.AnalysisTag)
| Method resolution order:
|   AnalysisTagSuperMouse
|   cdb._analysistag.AnalysisTag
|   cdb._base._CdbBase
|   __builtin__.object
|
| Methods defined here:
|
|   __init__(self, url='', wsdl_dir='/cdb/analysisTagSuperMouse?wsdl')
|       Construct a AnalysisTagSuperMouse.
|
|       @param url: the url of the server in the form 'http://host.domain:port'
|
|       @exception CdbPermanentError: Unable to contact CDB server or invalid URL
|
|   set_tag(self, tag)
|       Set the analysis tag template in the database.
|       @param tag: Python dictionary which defines the tag.
|       The following keys are expected to be present in the dictionary, otherwise an exception is thrown
|
|   :
|
|       name - the name of the analysis tag
|       blname - beamline tag name to go with this tag
|       ccname - same for the coolingchannel tag
|       absname - same for absorber
|       version - tag version
|       pdf_ref - a reference to the pdf file. This is an optional key, if missing its value is set to
|       "Global_"+ tag['name']+"_v"+str(tag['version'])+".root"
|
| -----
| Methods inherited from cdb._analysistag.AnalysisTag:
|
|   get_status(self)
|       Get the status of the service.
|
|       @return a string containing the status of the service
|
|       @exception CdbTemporaryError: The problem maybe transient and retrying the
|       request MAY succeed
```

```

| @exception CdbPermanentError: An unexpected internal error
|
| get_tag(self, tagname, version=0)
| Get the analysis tag for a given tag name. The version should be omitted to get the latest version
.
| :param tagname: name of the tag
| :param version: tag version (latest, if 0)
| :return: a tag dictionary
|
| set_url(self, url)
| Set the client to use the given CDB server.
|
| @param url: the URL of the CDB wsdl
|
| @exception CdbTemporaryError: Unable to contact CDB server
| @exception CdbPermanentError: Invalid URL
|
| -----

```

An example table after filling the same tag 3 times looks like this:

name	version	pdfref	blname	creationtime	validuntiltime	ccname	absname
myanatag	1		myblname	2017-08-14 12:35:45.43	2017-08-14 12:36:18.654	myccname	myabsname
myanatag	1		myblname	2017-08-14 12:36:18.654	2017-08-14 12:51:16.117	myccname	myabsname
myanatag	1		myblname	2017-08-14 12:51:16.117		myccname	myabsname

(3 rows)

#2 - 04 October 2017 11:53 - Martyniak, Janusz

And finally this is my mail from July 26th 2017 (which received no comments)

Hi Melissa and Adam,

When coding the analysistag setter code to allow experts to fill it in, I encountered a small issue which I would like to clarify.

Some background:

The cooling channel tag mechanism is designed in such a way that the existing tag keyed by the tag name is overwritten with a new tag data (magnet currents etc) and it's validity timestamp is set to infinity (technically == NULL). The old tag record is shadowed by inserting its validity time to the start time of a new record. This guaranties that there is one and only one 'live' tag of a given tag name.

We don't check in the CDB if the expert data are different from ones we are just hiding, these could be identical, we still shadow the old ones, the change is harmless in this case. The point to make here is that the tag name plays a key role here: if it exists, add a new one and shadow the old one, otherwise add a new one.

In our system, we agreed to have the following columns:

```
analysistagname | bl_name | cc_name | abs_name | version | pre_ref | creation | validity DEFAULT NULL |
```

now, the expert inserts (I'm omitting quotes for brevity):

```
my_tag, bl_tag1, cc_tag1, abs_tag1, 1, my_tag_v1_pdf.root
```

The effect is (subject to auto-generating the root file name, no important here)

```
my_tag | bl_tag1 | cc_tag1 | abs_tag1 | 1 | my_tag_v1_pdf.root | new() | NULL
```

A question:

If the expert uploads a new record, when should it be considered as a new one (modifying the NULL above) and when should it be considered as a new one ?

Possibilities:

- minimalistic : only analysistagname field matters, the rest is could be anything - if it matches we update: so we could possibly get 2 rows like this:

```
my_tag | bl_tag1 | cc_tag1 | abs_tag1 | 1 | my_tag_v1_pdf.root | prev_time | now()
my_tag | bl_tag2 | cc_tag1 | abs_tag1 | 1 | my_tag_v1_pdf.root | now() | NULL
```

Please note, I only changed the beamline tag name, did not bother to change anything else, even the version. (but they are free to be changed).

Is this a required behaviour ?

- treat the analysis tag and the sub tags as unseparable, so the second row above should be prevented from being inserted and the first row should be left unchanged ? In this case for a change to be of any use we have to change the only remaining column value which is a version number which would automatically set the pdf file name as a combination of the analysis tag and the version, as we agreed. If we keep the same version, it would correspond to a 'silent' change as in case of a CC tag as described at the top of this mail.

Another question is the purpose of the version column. In the non-minimalistic case, an update for the same tag name list would allow to insert a new pdf file reference by means of a version number (the analysis tag name stays the same). This would serve a purpose of updating the pdf file itself, when one decides that the old one was incorrect. Am I right ?

Adam said that we should not bother with the MAUS version, since MAUS knows its version. But still, the model above has no room for more than one pdf file for a given set of tags.

Any comments ?

cheers, JM