Discussion on open GFORGE feature requests 2020-08-26

Dmitriy Yadykin

CHALMERS
UNIVERSITY OF TECHNOLOGY

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Summary

1672	Proceedure for populating the public IMAS database at the Gateway	Daniel Figat	Thomas Johnson
1723	Install UDA/2.3.0 on GW	Fleury Ludovic	Bartek Palak
1755	Install latest FC2K at JET	Jonathan Hollocombe	Francis Casson
1745	Could we have an option in CHEASE to not modify R0 and B0?	Olivier Sauter	Thomas Johnson
996	time interpolation option for get_slice	Gabriele manduchi	Dmitriy Yadykin
1627	Code parameters in the parameter file (saved by autogui)	Dmitriy Yadykin	Dmitriy Yadykin
1724	Loosen dependencies of HDC module	Jonathan Hollocombe	Bartek Palak
1662	Syncing git repositories from ITER	Daniel Figat	Thomas Johnson
953	Checking CPO time dependent data when writing	Fleury Ludovic	Rui Coelho
1650	Simple suggestions to improve robustness of CPO based UAL	Olivier Hoenen	Thomas Johnson
1057	generic framework for non-regression testing of actors	Nobody	David Coster

11 feature requests are open, some of them are ‘obvious’, some are not.

Goal of this discussion:

- to go through not obvious feature requests and decide are they important for the WPCD deliverables for this year and the importance of these feature requests should be raised

Boundary condition: CPT related developments are in general performed in IMAS only.

1057: generic framework for non-regression testing of actors

Initial comment:

Now that we seem to have a functioning actor release procedure, it would be good to have an automatic non-regression testing framework for all actors.
Does the CPT have an idea about how this could be set up?
I think we would need to be able to have:
a generic test workflow consisting of
* ualinit
* actor
* ualslice-collector or ualcollector
and, in addition
* machine/shot/run containing the input for the actor
* machine/shot/run containing the reference output for the actor
* tool for comparing the actor output and the reference output

This could perhaps all be run under Jenkins ...

Thanks,
Dave

Status

- Regression test procedure is developed by Thomas
- more sophisticated (continuous integration framework) exists at ITER (see Mireille comment)

Questions

- Is regression procedure enough for now? Is the full continuous integration platform urgent?
- Is ITER framework(Bamboo based) can be used as is, or is there any better proposal (Jenkins was mentioned several times)

1650:Simple suggestions to improve robustness of CPO based UAL

Initial comment:

I had an an expected error from the UAL, so I started looking into what was going on and found a few things that could be improved...

1. The output variable IDX, from mdsEuitmOpenInterval, is not always set.
   This causes problems later on as the output IDX may be undefined from e.g. euitm_open_.
   I would suggest to set IDX=0 at the start of mdsEuitmOpenInterval.
2. In euitm_open_ : After running mdsEuitmOpen, the first step is to check if the status:
   ```
   if(status) return status;
   ```
   At this point retIdx is not set, thus it remains undefined.
   The second step is to set the output flag:
   ```
   *retIdx = setMdsIdx(shot, run, idx);
   ```
   So, we need to somhow initialise retIdx in case status is "true".
3. In euITM_getequilibrium (line 64 in equilibrium.f90) you call get_dimensions to find the values of dim1, dim2...
   However, this function returns without setting any output variables in case the input idx is "invalid".
   Still the data dim1 is used without checking if it has been filled, which causes a nasty crash.
   Here a simple solution would be to check IDX at the start of euITM_getequilibrium and
   similar routines for any CPO.

Status

- According to CPT (Olivier) these issues are not present for IMAS

Questions

- Is there any obvious test case that can be tried in IMAS?

953:Checking CPO time dependent data when writing

Initial comment:

In Python i found myself with a nasty problem when writing/reading data that is time dependent. In a CPO with ntime time slices, we can assign just the first time slice and not fill the remaining ones and when writing with cpo.put() no warning message is issued. Reading the cpo on the other hand will fail miserably (also in ISE for instance, not only in python).
Is it possible to implement a checker on the put() such that one ensures that all time dependent fields that are used have all time slices filled to avoid the reading problem ?

simple code found below

Rui

Status:

- Implementation is possible but requires considerable amount of work

Questions:

- how urgent/general is this issue?
- can such check be done on the code level (fill all time bases with dummy data)?

1662: Syncing git repositories from ITER

Initial comment:

We have a number of git repositories at ITER that we need to have a mirror of on gforge.
I've started the process by creating the corresponding gforge repos and created mirrors. The repositories are
https://*g2tjohns*@gforge6.eufus.eu/git/hcd-wf
https://*g2tjohns*@gforge6.eufus.eu/git/ids-tools
https://*g2tjohns*@gforge6.eufus.eu/git/hcd2core-source
https://*g2tjohns*@gforge6.eufus.eu/git/hcd2core-prof
https://*g2tjohns*@gforge6.eufus.eu/git/coresourcecomb
and the corresponding repos at ITER
ssh://git@git.iter.org/wf/hcd-wf.git
ssh://git@git.iter.org/imex/ids-tools.git
ssh://git@git.iter.org/heat/hcd2core-sources.git
ssh://git@git.iter.org/heat/hcd2core-profiles.git
ssh://git@git.iter.org/tran/core_sources-combiner.git
This is how the mirrors were created:
git clone --mirror <ITER-URL>
cd <git-dir-created-by-clone>
git push --mirror <gforge-URL>
Can someone please automate the syncing of these repositories? At this point, ITER is the master and gforge the slave.

Status:

- work is started, but not finished

Questions:

- should the importance of this request be raised
- will it be different with GFORGE NEXT? Should we wait for installation?

1627:Code parameters in the parameter file (saved by autogui)

Initial comment:

This is a 'cosequence' of the discussions with ETS developers and users. The request is to provide better way to store and manage code parameters in the parameter file (or somewhere else). Right now, the way code parameters are stored is not 'user friendly', i.e. making parameter file hard to read. It is also not easy to edit code parameters in the parameter file as they are stored following (probably) xml-like format.
As for the user-friendliness the present proposals are either to:
- move code parameters to the end of the parameter file (not to 'pollute' the main section)
- or to store them n the separate file (assosiated with the 'main' file in some way)

As for the ways to make code parameters 'easy editable' there are no clear ideas, we probably need to clarify possible solutions first

Regards
Dmitriy

Status:

- implementation design is not defined, probably will take some time to design and implement

Questions:

- is this urgent for this year?

1627:time interpolation option for get_slice

Initial comment:

I think it would be useful to add time interpolation option to the actors using get_slice ual routine. At present this option is set to 2 (previous time slice) and this can not be changed.

Regards
Dmitriy

Status:

- time interpolation option is implemented in IMAS AL
- option implementation need to be finalized
- probably workflow modification will be needed

Questions:

- should the importance of this tracker be raised

1745:Could we have an option in CHEASE to not modify R0 and B0?

Initial comment:

In the ETS the CHEASE actor updates the R0 and B0 in the vaccum field structure. Since R0 is a static variable in the equilibrium IDS, it is only stored once - the value of R0 during later time steps are ignored.
This causes problem when running the ETS with a moving boundary. Then CHEASE is suggesting that R0 moves around, which generates variations in B0. The result is that B0 is stored by the UAL, but R0 is only stored on the first call and consequently we get an incorrect variation in R0*B0.
The suggestion to solve this issue is that CHEASE either don't touch the input R0 and B0. Or if this is not acceptable, then we can have an XML option to not change R0 and B0.

Status:

- initial discussion is started, implementation is not done

Questions:

- should the importance of this request be raised?
- should the imnplementation be planned for one of the nearest sprints?