

C Instrumentation Framework - Feature #6780

Implement CIF in Python (script and module)

02/03/2016 05:17 PM - Evgeny Novikov

Status:	New	Start date:	02/03/2016
Priority:	High	Due date:	
Assignee:	Ilya Shchepetkov	% Done:	0%
Category:	Core	Estimated time:	0.00 hour
Target version:			
Published in build:			
Description			
At the moment different Python tools use CIF, so, best practices should be implemented in a Python library. In addition, please see https://forge.ispras.ru/issues/9346#note-3 .			
BTW, it has sense to specify a list of compiler options that CIF should use for preprocessing, source code querying and instrumentation (just for C-backend!) rather than a list of compiler options that are not supported by CIF.			
Related issues:			
Related to C Instrumentation Framework - Feature #6829: Make CIF to look like...		Open	02/09/2016
Blocked by Klever - Bug #6769: External commands failures are not included in...		Closed	02/03/2016
Blocks Klever - Feature #6644: Debug CIF just when debugging		New	01/29/2016

History

#1 - 04/27/2016 03:56 PM - Evgeny Novikov

BTW, each plugin should provide a unique output file for CIF, e.g. some file placed into a plugin working directory, to avoid races between different plugins (CIF outputs all intermediate and output files near an output file). At the moment SA uses original output files that lead to intermediate CIF files appear near original input files.

#2 - 09/22/2017 05:17 PM - Evgeny Novikov

- Subject changed from *Implement AVTG common library to invoke CIF and process its output* to *Implement VTG common library to invoke CIF and process its output*

- Category changed from **Abstract tasks generation* to *Tasks generation*

#3 - 10/17/2017 10:49 AM - Evgeny Novikov

Copied from the [#8504](#) description:

When CIF failed, there are too many different errors in output log. But in fact, only first error is informative. It is offered to show the first error and skip others.

#4 - 10/19/2017 12:38 PM - Ilya Zakharov

- Assignee set to *Alexey Polushkin*

- Priority changed from *Normal* to *Urgent*

- Target version set to *1.0*

#5 - 10/25/2017 12:39 PM - Evgeny Novikov

- Assignee deleted (*Alexey Polushkin*)

- Priority changed from *Urgent* to *High*

- Target version deleted (*1.0*)

Indeed this isn't so important issue while [#8504](#) is implemented without any common CIF library.

#6 - 09/28/2018 04:58 PM - Evgeny Novikov

- Priority changed from *High* to *Urgent*

- Category deleted (*Tasks generation*)

- Description updated

- Subject changed from *Implement VTG common library to invoke CIF and process its output* to *Implement common Python library to invoke CIF and*

process its output

- Project changed from Klever to C Instrumentation Framework

#7 - 12/17/2018 08:55 AM - Evgeny Novikov

- Blocks Bug #9346: Remove "-include" options after second stage added

#8 - 12/17/2018 09:11 AM - Evgeny Novikov

- Description updated

- Subject changed from Implement common Python library to invoke CIF and process its output to Implement CIF in Python (script and module)

#9 - 12/17/2018 09:11 AM - Evgeny Novikov

- Subject changed from Implement CIF in Python (script and module) to Implement CIF in Python (script and module)

#10 - 12/17/2018 03:30 PM - Evgeny Novikov

- Assignee set to Ilya Shchepetkov

- Category set to Core

Ilya is the most appropriate developer of CIF in Python.

#11 - 06/04/2019 04:20 PM - Evgeny Novikov

- Blocks deleted (Bug #9346: Remove "-include" options after second stage)

#12 - 06/18/2019 12:22 PM - Evgeny Novikov

- Priority changed from Urgent to High

Although, the suggested improvement will help to several CIF users, it is not extremely important.

#13 - 06/19/2019 02:28 AM - Ilya Shchepetkov

- Related to Feature #6829: Make CIF to look like GCC added

#14 - 02/10/2020 06:18 PM - Evgeny Novikov

- Blocks deleted (Feature #6731: Pass source code just through C-backend if aspects are empty)