

## Quickstart - Fortran Modernization with Himeno

1. This quickstart works with the [performance-demos-fortran](#) GitHub repository, more specifically with [Himeno/serial](#). First, let's produce the Screening Report of Codee (screening command):

```
codee screening --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himeno.f90
```

```
Date: 2024-04-22 Codee version: 2024.2.2
[Fortran] target compiler: <none> (Compiler Agnostic Mode)

[1/1] himeno.f90 ... Done

SCREENING REPORT

---Number of files---
Total | C C++ Fortran
-----|-----
1      | 0 0 1

Lines of code Analysis time # checks Profiling
-----|-----
214      156 ms      6      n/a

CHECKS PER CATEGORY AND PRIORITY LEVELS

| -----Checks per category----- | Priority |
| Scalar Control Memory Vector Multi Offload Quality | L1 L2 L3 |
| -----|-----|
| n/a n/a n/a n/a n/a n/a 6 | 1 0 5 |

Lines of code : total lines of code found in the target (computed the same way as the sloccount tool)
Analysis time : time required to analyze the target
# checks : total actionable items (opportunities, recommendations, defects and remarks) detected
Profiling : estimation of overall execution time required by this target

RANKING OF CHECKERS

Checker Level Priority # Title
-----|-----
PWR063 L1 P12 1 Avoid using legacy Fortran constructs
PWR001 L3 P3 5 Declare global variables as function parameters

SUGGESTIONS

Use 'checks' to find out details about the detected checks:
codee checks --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himeno.f90

Use --target-arch to focus on the checks most relevant to your hardware type [cpu | gpu | mcu],
e.g.:
codee screening --target-arch cpu --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063
himeno.f90

1 file, 7 functions, 5 loops successfully analyzed and 0 non-analyzed files in 156 ms
```

2. Follow the suggestions in order to produce the Checkers Report (option --checks) that lists all the checks applicable to your code.

```
codee checks --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himeno.f90
```

```
Date: 2024-04-22 Codee version: 2024.2.2
[Fortran] target compiler: <none> (Compiler Agnostic Mode)

[1/1] himeno.f90 ... Done

CHECKS REPORT
```

```

himenof90 [PWR063] (level: L1): Avoid using legacy Fortran constructs
himenof90:136:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:164:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:223:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:255:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:275:1 [PWR001] (level: L3): Declare global variables as function parameters

SUGGESTIONS

Use --verbose to get more details, e.g:
codee checks --verbose --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himenof90

Use --level to filter checks with a specific level of priority, e.g:
codee checks --level L1 himenof90

More details on the defects, recommendations and more in the Open Catalog of Best Practices for
Performance:
https://github.com/codee-com/open-catalog/

1 file, 7 functions, 5 loops successfully analyzed and 0 non-analyzed files in 145 ms

```

3. Show the detailed Checkers Report (option **--verbose**). As an example, focus on the checker *PWR063*, related to replacing legacy Fortran constructs. The detailed Codee output, which includes links to the open catalog available in the website, precise location in the source code, etc..., is as follows:

```
codee checks --verbose --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himenof90
```

```

Date: 2024-04-22 Codee version: 2024.2.2
Compiler flags: -O3

[Fortran] target compiler: <none> (Compiler Agnostic Mode)

[1/1] himenof90 ... Done

CHECKS REPORT

himenof90 [PWR063] (level: L1): Avoid using legacy Fortran constructs
PAUSE:
131: pause
Suggestion: Remove the legacy fortran constructs and refactor the code to comply with modern Fortran
standards.
Documentation: https://github.com/codee-com/open-catalog/tree/main/Checks/PWR063
...

1 file, 7 functions, 5 loops successfully analyzed and 0 non-analyzed files in 132 ms

```

4. Follow the Codee suggestion to remove the legacy Fortran construct (PAUSE) detected on line 131. Comment-out or remove that line, since the execution stops right after.
5. Run the checks report again in order to verify that the PWR063 checker is no longer triggered for himenof90:

```
codee checks --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himenof90
```

```

Date: 2024-04-22 Codee version: 2024.2.2
[Fortran] target compiler: <none> (Compiler Agnostic Mode)

[1/1] himenof90 ... Done

CHECKS REPORT

himenof90:136:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:164:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:223:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:255:1 [PWR001] (level: L3): Declare global variables as function parameters
himenof90:275:1 [PWR001] (level: L3): Declare global variables as function parameters

SUGGESTIONS

Use --verbose to get more details, e.g:
codee checks --verbose --check-id PWR001,PWR002,PWR003,PWR007,PWR008,PWR012,PWR063 himenof90

Use --level to filter checks with a specific level of priority, e.g:
codee checks --level L1 himenof90

```

More details on the defects, recommendations and more in the Open Catalog of Best Practices for Performance:  
<https://github.com/codee-com/open-catalog/>

1 file, 7 functions, 5 loops successfully analyzed and 0 non-analyzed files in 149 ms

The PWR063 no longer appears in the output, therefore, no more Fortran legacy constructs are needed to be updated.