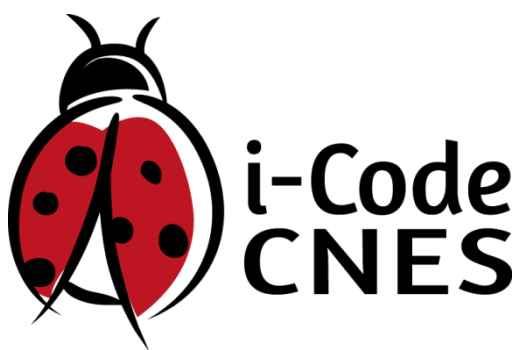


User Manual

i-Code CNES

DNO/DA /AQ - 2017.0002478

Version 3.0.1



<p style="text-align: center;">CNES DNO/DA/AQ</p>	<p>i-Code CNES User Manual</p>	<p>Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 2/59</p>
--	---------------------------------------	---

References

References		Title
R1		Eclipse documentation : http://www.eclipse.org/documentation/
R2	RNC-CNES-Q-HB-80-505	Coding rules Fortran 77 Version 7
R3	RNC-CNES-Q-HB-80-517	Coding rules Fortran 90 Version 5
R4	RNC-CNES-Q-HB-80-501	Common coding rules Version 5
R5	RNC-CNES-Q-HB-80-516	Coding rules SHELL Version 6

Table of contents

1. INTRODUCTION	4
2. I-CODE CNES IDE INTEGRATED INTO ECLIPSE	4
2.1. I-CODE CNES CONFIGURATION.....	4
2.1.1. CONFIGURATION CHOICE	5
2.1.2. RULES FILTER	6
2.1.3. RULES SELECTION	6
2.1.4. RULE'S SEVERITY CONFIGURATION	7
2.1.5. METRICS' THRESHOLD CONFIGURATION	7
2.1.6. RESTORE TO DEFAULT	8
2.2. CHECK RULES AND COMPUTE METRICS	9
2.2.1. ANALYSE LAUNCH	9
2.2.2. RESULTS DISPLAY	10
2.2.3. I-CODE CNES DISPLAY IN THE EDITOR.....	12
2.2.4. RESULTS EXPORT	13
3. I-CODE CNES COMMAND LINE	15
2.3. XML FORMAT	15
2.4. HTML FORMAT.....	15
2.5. HELP DISPLAY	15
4. RULES DESCRIPTION	16
4.1. COMMON RULES	16
4.2. SPECIFIC RULES	26
4.2.1. FORTRAN 77	26
4.2.2. FORTRAN 90	31
4.2.3. SHELL	43
5. COMPUTED METRICS	48
6. USER MESSAGES.....	51
6.1. RULE'S VIOLATION MESSAGES	51
6.1.1. REGLES COMMUNES.....	51
6.1.2. FORTRAN 77	53
6.1.3. FORTRAN 90	55
6.1.4. SHELL	57
7. LIMITATIONS	59
7.1. POWERSHELL DOES NOT ALLOW ANALYSE LAUNCH.....	59
7.2. SHELL: STRINGS SHOULD BE LIMITED WITH « »	59

1. INTRODUCTION

This document is the i-Code CNES user manual. It describes how to use i-Code CNES.



Before using i-Code CNES, you should:

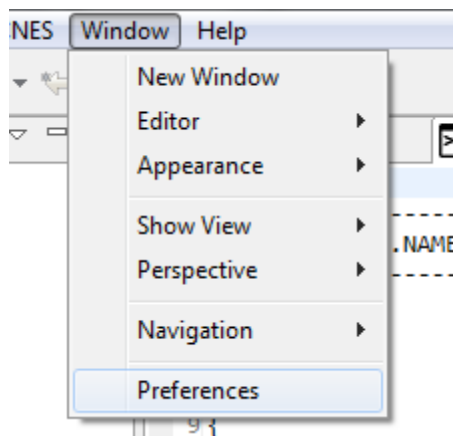
- Know eclipse. Eclipse documentation is available here [R1].
- Read CNES coding rules [R2] [R3] [R4]

2. I-CODE CNES IDE INTEGRATED INTO ECLIPSE

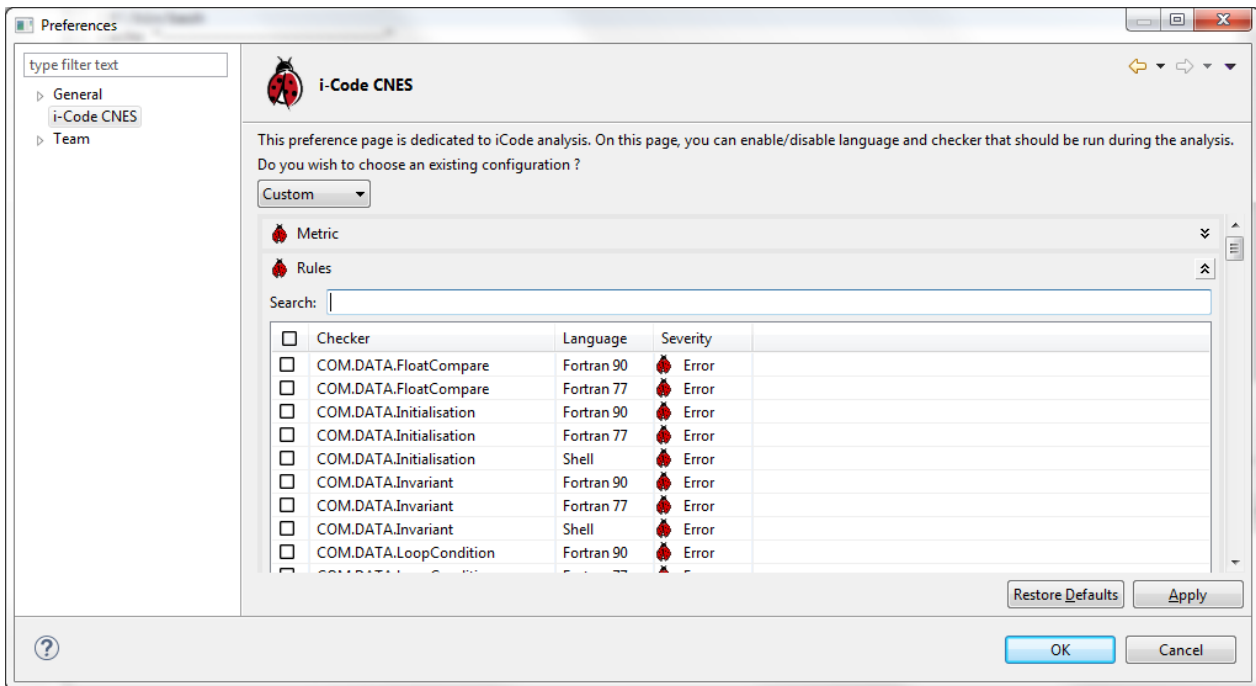
2.1. I-CODE CNES CONFIGURATION

The whole i-Code CNES configuration is accessible from eclipse preference pages.

To open eclipse preference pages, click on Window > Preferences



In the eclipse preference window, select the « i-Code CNES » item on the left.



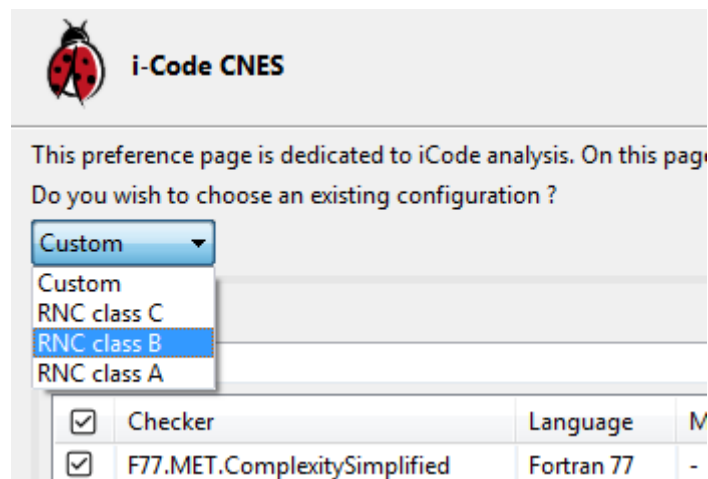
To see all available coding rules, open up « Metrics » or « Rules » tabs.

2.1.1. CONFIGURATION CHOICE

i-Code CNES preference pages offer the user to create his own configuration, selecting the rules he wants to check. Default configurations are of course available.

Au-dessus des volets « Metrics » et « Rules » un menu déroulant propose de sélectionner la configuration souhaitée.

Right above the « Metric » and « Rules » tabs, a combo box permits to choose one of the predefined configuration.



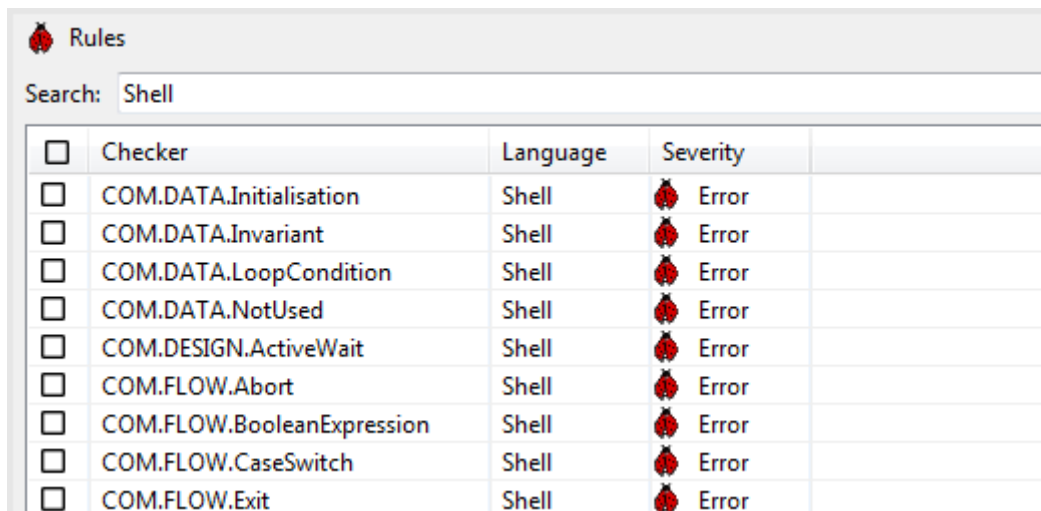
« Custom » allows the user to:

- Activate/desactivate the rules;
- Define each rule severity;
- Define threshold for each metric;

If a predefined configuration is selected, no modification is possible.

2.1.2. RULES FILTER

Rules can be filtered with keyword, using for example the language name: Shell.

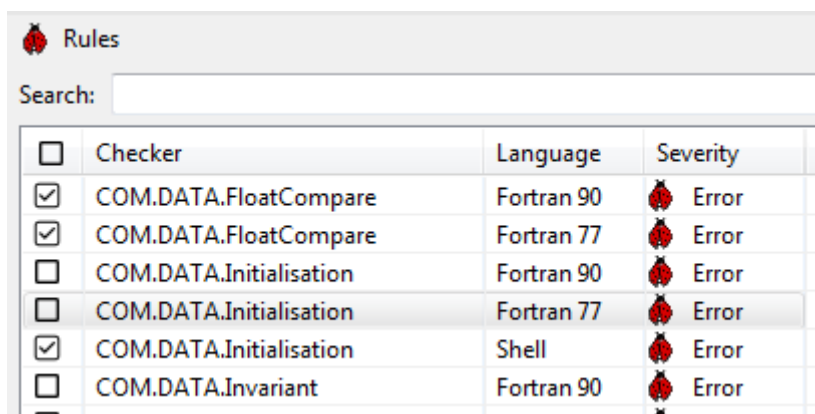


The screenshot shows a window titled 'Rules' with a search bar containing the text 'Shell'. Below the search bar is a table with the following columns: 'Checker', 'Language', and 'Severity'. The table lists several rules, all of which have 'Shell' as the language and 'Error' as the severity. Each rule has a checkbox on the left, which is currently unchecked.

<input type="checkbox"/>	Checker	Language	Severity
<input type="checkbox"/>	COM.DATA.Initialisation	Shell	Error
<input type="checkbox"/>	COM.DATA.Invariant	Shell	Error
<input type="checkbox"/>	COM.DATA.LoopCondition	Shell	Error
<input type="checkbox"/>	COM.DATA.NotUsed	Shell	Error
<input type="checkbox"/>	COM.DESIGN.ActiveWait	Shell	Error
<input type="checkbox"/>	COM.FLOW.Abort	Shell	Error
<input type="checkbox"/>	COM.FLOW.BooleanExpression	Shell	Error
<input type="checkbox"/>	COM.FLOW.CaseSwitch	Shell	Error
<input type="checkbox"/>	COM.FLOW.Exit	Shell	Error

2.1.3. RULES SELECTION

You can select/unselect a rule using the checkbox on the left of the rule considered.



The screenshot shows the 'Rules' window with a search bar that is empty. The table below shows a list of rules. The first checkbox is unchecked, while the checkboxes for 'COM.DATA.FloatCompare' (Fortran 90 and Fortran 77) and 'COM.DATA.Initialisation' (Shell) are checked. The 'COM.DATA.Initialisation' (Fortran 77) row is highlighted.

<input type="checkbox"/>	Checker	Language	Severity
<input checked="" type="checkbox"/>	COM.DATA.FloatCompare	Fortran 90	Error
<input checked="" type="checkbox"/>	COM.DATA.FloatCompare	Fortran 77	Error
<input type="checkbox"/>	COM.DATA.Initialisation	Fortran 90	Error
<input type="checkbox"/>	COM.DATA.Initialisation	Fortran 77	Error
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Shell	Error
<input type="checkbox"/>	COM.DATA.Invariant	Fortran 90	Error

You can select/unselect all the rules using the first checkbox.

Rules

Search:

<input checked="" type="checkbox"/>	Checker	Language	Severity
<input checked="" type="checkbox"/>	COM.DATA.FloatCompare	Fortran 90	Error
<input checked="" type="checkbox"/>	COM.DATA.FloatCompare	Fortran 77	Error
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Fortran 90	Error
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Fortran 77	Error
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Shell	Error
<input checked="" type="checkbox"/>	COM.DATA.Invariant	Fortran 90	Error
<input checked="" type="checkbox"/>	COM.DATA.Invariant	Fortran 77	Error

Check to select or unselect every rules in the table.

2.1.4. RULE'S SEVERITY CONFIGURATION

Rule criticality can be: Error / Warning / Info

It can be modified in the Severity column. No bulk change is available.


Rules

Search:













<input checked="" type="checkbox"/>	Checker	Language	Severity
<input checked="" type="checkbox"/>	COM.DATA.FloatCompare	Fortran 90	Error
<input checked="" type="checkbox"/>	COM.DATA.FloatCompare	Fortran 77	Error
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Fortran 90	Error
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Fortran 77	Warning
<input checked="" type="checkbox"/>	COM.DATA.Initialisation	Shell	Info
<input checked="" type="checkbox"/>	COM.DATA.Invariant	Fortran 90	Warning
<input checked="" type="checkbox"/>	COM.DATA.Invariant	Fortran 77	Error

2.1.5. METRICS' THRESHOLD CONFIGURATION

The thresholds can be modified in the « Minimum » and « Maximum » columns.

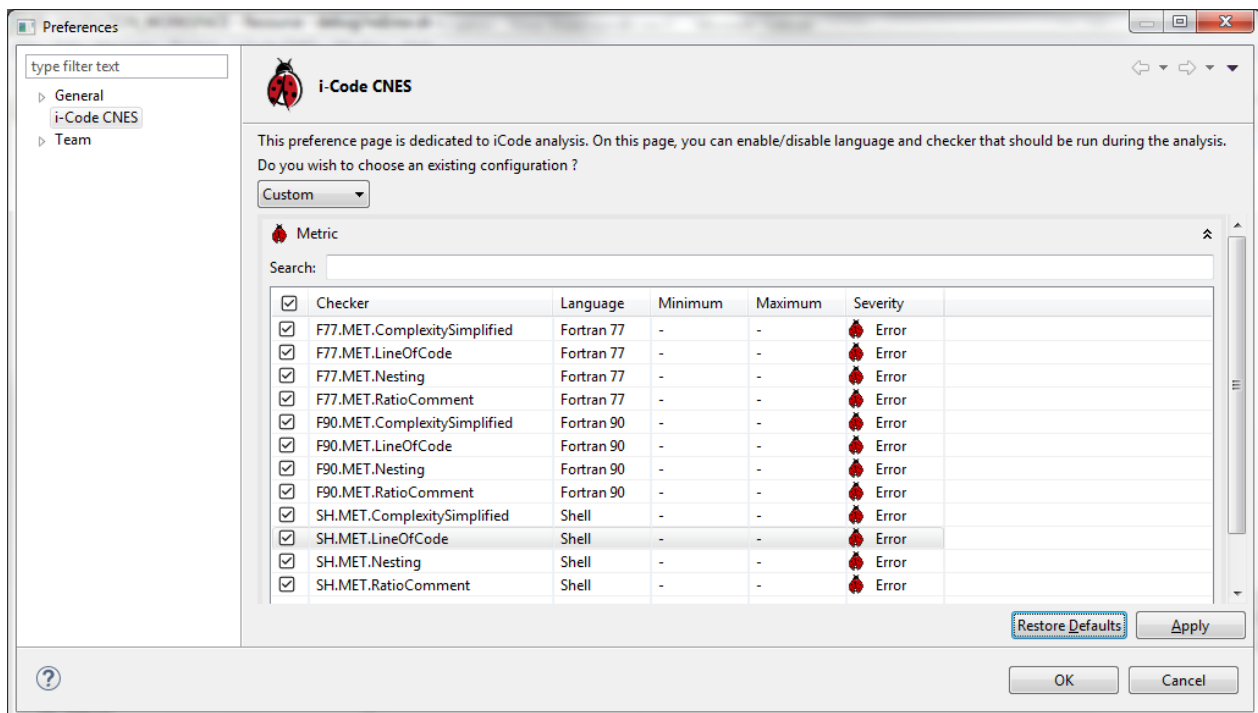
 Metric

Search:

<input checked="" type="checkbox"/>	Checker	Language	Minimum	Maximum	Severity
<input checked="" type="checkbox"/>	F77.MET.ComplexitySimplified	Fortran 77	-	10.0	 Warning
<input checked="" type="checkbox"/>	F77.MET.LineOfCode	Fortran 77	-	60.0	 Warning
<input checked="" type="checkbox"/>	F77.MET.Nesting	Fortran 77	-	5.0	 Warning
<input checked="" type="checkbox"/>	F77.MET.RatioComment	Fortran 77	30.0	-	 Warning
<input checked="" type="checkbox"/>	F90.MET.ComplexitySimplified	Fortran 90	-	10.0	 Error
<input checked="" type="checkbox"/>	F90.MET.LineOfCode	Fortran 90	-	60.0	 Error
<input checked="" type="checkbox"/>	F90.MET.Nesting	Fortran 90	-	5.0	 Error
<input checked="" type="checkbox"/>	F90.MET.RatioComment	Fortran 90	30.0	13	 Error
<input checked="" type="checkbox"/>	SH.MET.ComplexitySimplified	Shell	4.0	10.0	 Info
<input checked="" type="checkbox"/>	SH.MET.LineOfCode	Shell	-	60.0	 Error
<input checked="" type="checkbox"/>	SH.MET.Nesting	Shell	-	5.0	 Info
<input checked="" type="checkbox"/>	SH.MET.RatioComment	Shell	30.0	-	 Error

2.1.6. RESTORE TO DEFAULT

You can go back to the default configuration using the « Restore Defaults » button.

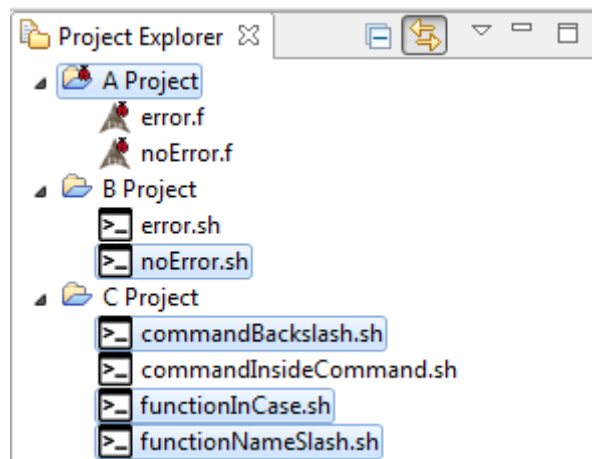


2.2. CHECK RULES AND COMPUTE METRICS

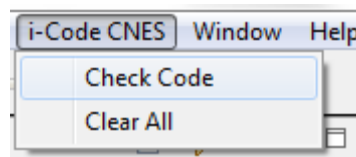
2.2.1. ANALYSE LAUNCH

Required: All the files to analyze are available into *Project Explorer*.

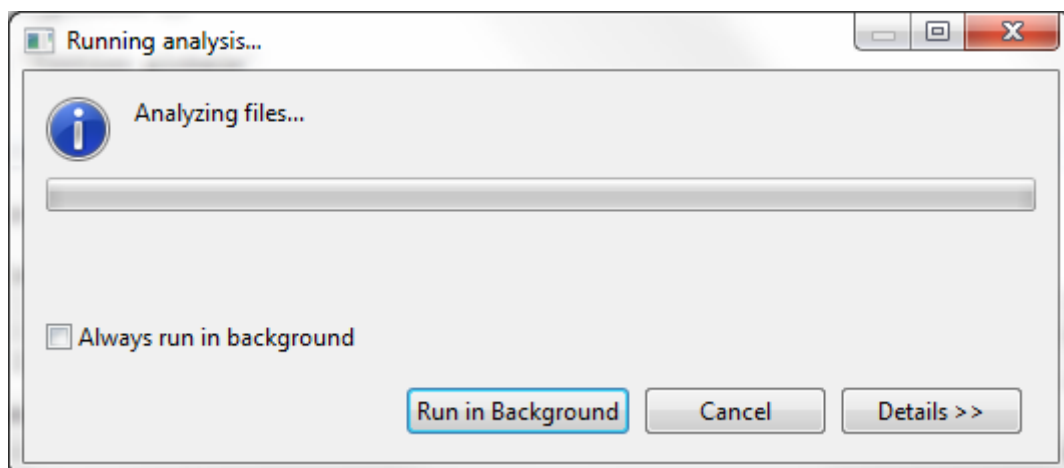
Step 1 : Select all the files you want to analyze into the *Project Explorer*.



Step 2: Launch the analyse from i-code CNES menu: i-Code CNES > Check Code.



The following progress bar appears. When the analyze is done, the results are displayed into « I-Code CNES Violations » and « i-Code CNES Metrics »



2.2.2. RESULTS DISPLAY

2.2.2.1. I-CODE CNES VIOLATIONS

!	Rule	Line	Num...	Message
▶	COM.PRES.Indent	--	50	
▶	F90.DATA.Declaration	--	10	
▶	F90.TYPE.Integer	--	10	
▲	COM.DATA.Invariant	--	6	
▲	fortran90.f90	--	6	
	MAIN PROGRAM	17	--	The variable DOUBLE must be defi...
	MAIN PROGRAM	27	--	The variable DISTANCE must be d...
	MAIN PROGRAM	34	--	The variable x1 must be defined as...
	MAIN PROGRAM	37	--	The variable Q must be defined as ...
	MAIN PROGRAM	43	--	The variable COMPL must be defi...
	MAIN PROGRAM	44	--	The variable TEMP must be define...

This tab displays :

- *Rule*: name of the rule which is not respected
- *Line*: line detection
- *Number of violations*: total number of violations.
- *Message*: error details

Violations on the same rule are displayed into a tree. The first level is the rule, the second level is the location.

2.2.2.1.1. Change display

The display can be changed using toolbar:

Location	Line	!	Num
▶ fortran90.f90	--		115
▲ error.f	--		20
▶ PROGRAM TEST	--		15
▲ subroutine AffecteValeur	--		3
▲ COM.DATA.NotUsed	--	🐞	2
The variable intent is declared and not used.	27		--
The variable intent is declared and not used.	26		--
▶ COM.PROJECT.Header	--	🐞	1
▶ subroutine affectevaleur	--		1
▶ No file header existing.	--		1

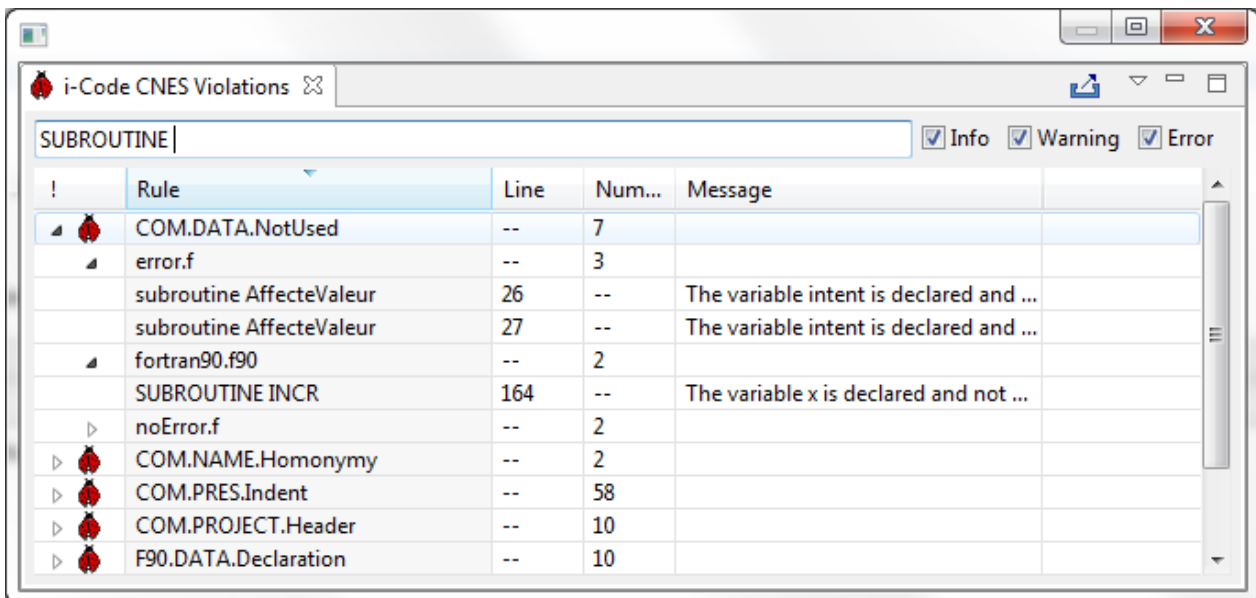
Il est donc possible de choisir une arborescence :

- Quality analyze display : Règles > Fichiers > Fonctions > Violations
- Development display : Fichiers > Fonctions > Règles > Violations

2.2.2.1.2. Results filter

It is possible to filter results in two ways:

Search toolbar: Search field allows to filter by rule name, file name, or keyword.



Severity: Checkboxes allows to directly filter on severity

2.2.2.1.3. Results sort

You can sort results using the column headers

2.2.2.2. I-CODE CNES METRICS

i-Code CNES Metrics displays all the metrics computed.

Metric	Total	Mean	Minimum	Maximum
▷ F77.MET.ComplexitySimplified	--	7.25	1.0	26.0
▷ F77.MET.LineOfCode	172.0	42.75	3.0	154.0
▷ F77.MET.Nesting	--	0.5	0.0	2.0
▷ F77.MET.RatioComment	--	--	28.638496	36.363636
▷ F90.MET.ComplexitySimplified	--	2.25	1.0	4.0
▷ F90.MET.LineOfCode	177.0	38.5	8.0	118.0
▷ F90.MET.Nesting	--	0.75	0.0	1.0
▷ F90.MET.RatioComment	--	--	25.0	39.772728

When a metric value is higher than the threshold, it is displayed in red.

This tab displays :

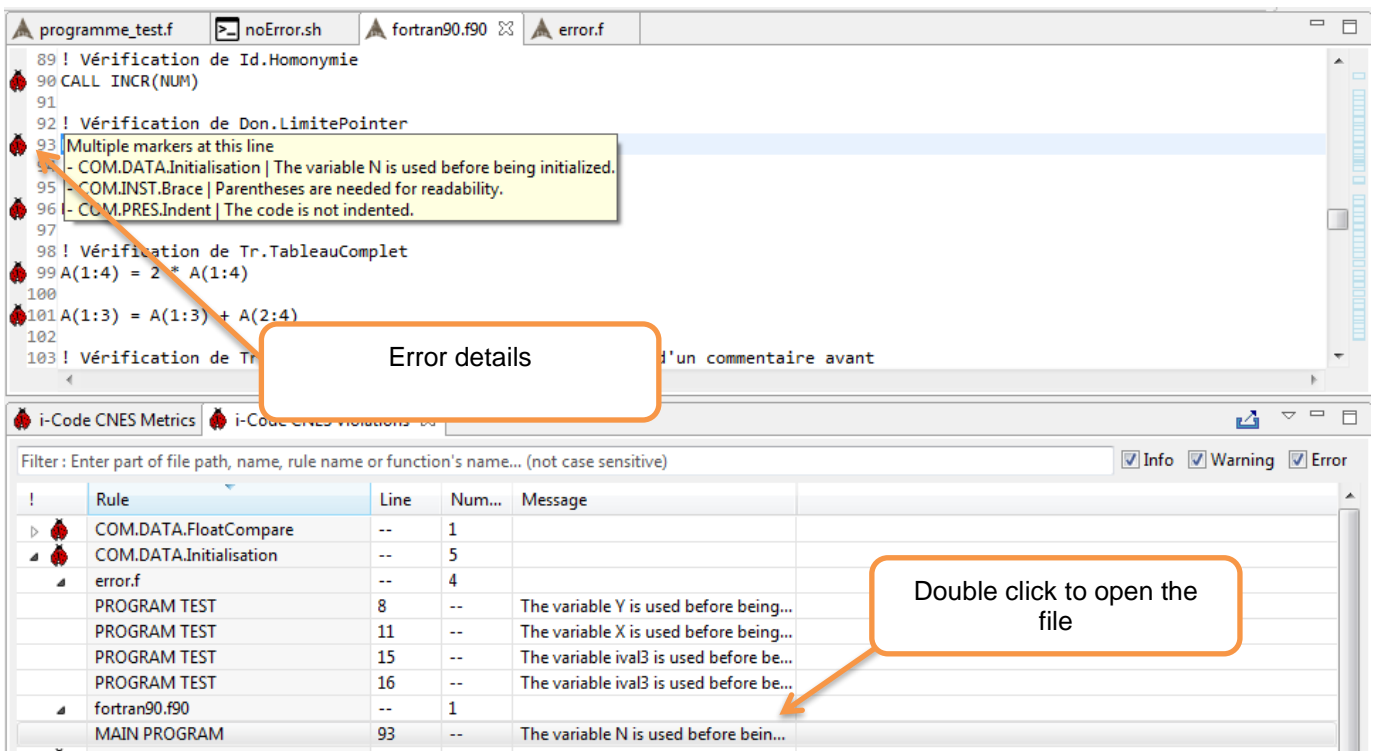
- *Metric* : metric name
- *Total* : total for this metric

- *Mean* : mean for this metric
- *Minimum* : minimum value
- *Maximum* : maximum value

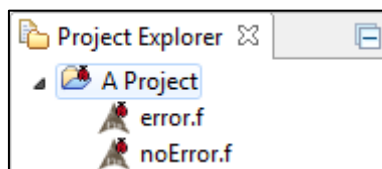
2.2.3. I-CODE CNES DISPLAY IN THE EDITOR

When double click on a violation, the file is opened in the editor at the right line.

A marker on the left side displays the error details.



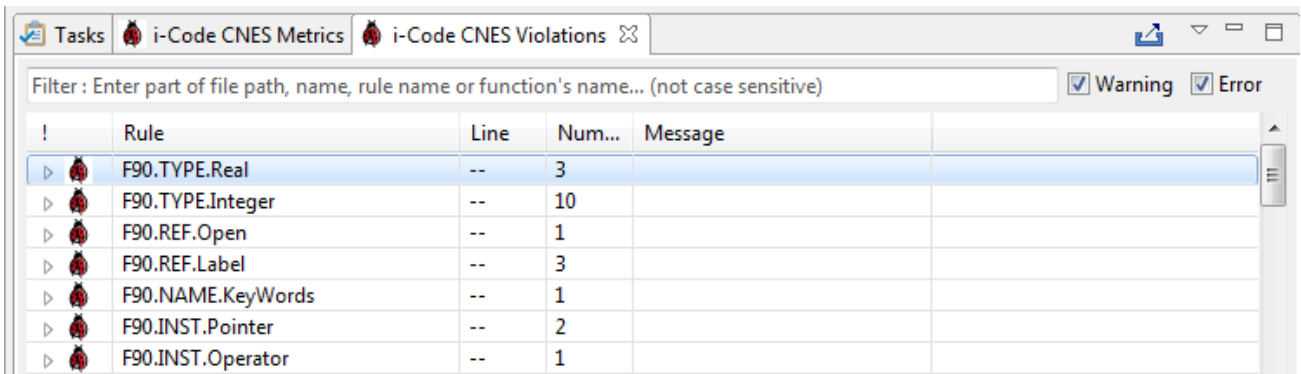
In the Project Explorer, a marker is displayed on the files containing violations.



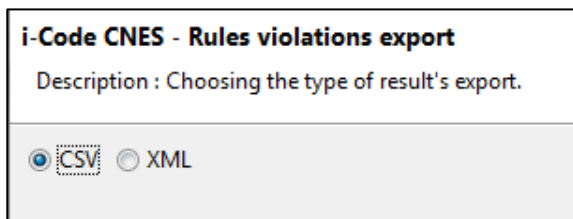
2.2.4. RESULTS EXPORT

Results can be exported as csv or xml file.

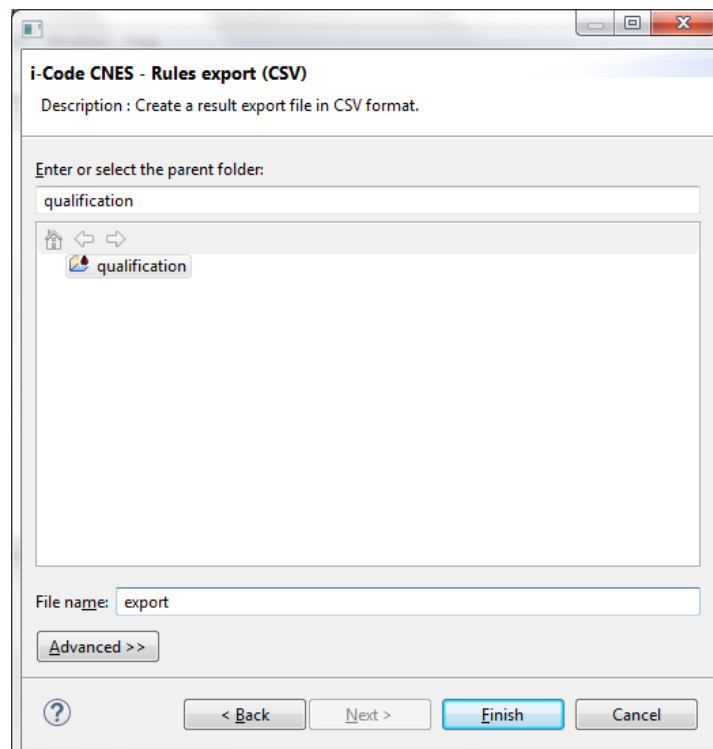
1. Click on Export button, in the top right in « i-Code CNES Violations » or « i-Code CNES Metrics »



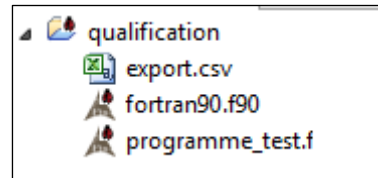
2. Select export format : csv ou xml



3. Precise export file name and location.



4. Results file is created in the selected project



CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 15/59
--------------------------	--------------------------------	---

3. I-CODE CNES COMMAND LINE

To launch an analyze with the command line version, you can use the following commands:

```
icode -f <format> <parameters> -output <output> <files>
```

- <format>: export format (xml/cvs)
- <parameters>: only for xml (see following sections)
- <output>: results output file
- <files> : paths to the files to analyze

Example: icode *.f90 ; icode ./tmp/myfiles.f77 ;

2.3. XML FORMAT

XML format allows to add specific values into the output file.

Optional parameters are the following :

- -project <name> : project name
- -projectVersion <version> : project version.
- -author <name> : author name.
- -configID <id> : configuration id.

Exemples :

- icode -f xml -output result.xml *f
- icode -f xml -author "John Doe" -output result.xml *f ;
- icode -f xml -author "John Doe" -project "MyProject" -output export.xml
./tmp/file.sh

2.4. HTML FORMAT

It is possible to directly convert xml to html using the option -html and -htmlOutput xx.html.

Exemples :

- icode -f xml -output result.xml -html *f
- icode -f xml -author Me -output result.xml -html -htmlOutput file.html *f ;

2.5. HELP DISPLAY

```
icode -help
```

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 16/59
--------------------------	--------------------------------	---

4. RULES DESCRIPTION

4.1. COMMON RULES

Rule	Rule description	i-Code Check
COM.DATA.Array	Obligation d'utiliser les tableaux à deux dimensions de manière à avoir "ligne x colonne" ----- Two dimensions arrays should be declared in this way : line x column	Non
COM.DATA.DeclarationOrder	Fortran 77 : Non applicable / not applicable	Non
	Fortran 90 : Les paramètres des fonctions doivent être déclarés, en premier les données d'entrée, après entrée/sortie puis les données de sortie. La déclaration se fait au travers du mot clé 'INTENT' puis 'IN' pour les données d'entrée, INOUT pour les entrées/sorties et OUT pour les sorties. S'il n'y a pas de paramètre INTENT, ce n'est pas possible de faire la vérification. ----- Fortran 90 : Parameters should declared in the following order : input, input/output, output. Déclaration is made using keyword "INTENT", and then "IN", "INOUT" and "OUT". If "INTENT" keyword is not used, the check is impossible.	Oui
	Shell : non applicable / not applicable	Non
COM.DATA.FloatCompare	Comparaison d'égalité/inégalité (.EQ., ==, .NE., /=) interdite entre des nombres réels (REAL, DOUBLE PRECISION ou COMPLEX). ----- Equality and non equality (.EQ., ==, .NE., /=) are forbidden on reals (REAL, DOUBLE PRECISION, COMPLEX)	Oui
	Shell : non applicable / not applicable	Non
COM.DATA.Initialisation	Les variables doivent être initialisées avant d'être utilisées. Quand une variable est utilisée dans le code, le programme vérifie qu'elle est initialisée (nom de la variable puis signe d'égalité), sinon, il renvoie une erreur. ----- Variables have to be initialized before being used. When a variable is used, the algorithm should checked if it is initialized and if not throw an error.	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 17/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
	Shell : Non implémenté / not available	Non
COM.DATA.Invariant	Les données déclarées dans une subroutine, fonction, etc et jamais modifiées (pas d'occurrence de la variable puis signe d'égalité) doivent être définies comme constantes ----- Data declared into subroutine, fonction, ... and which are never modified then should be declared as constant.	Oui
	Shell : Non implémenté / not available	Non
COM.DATA.LoopCondition	Interdiction de modifier les données de condition de sortie des boucles à l'intérieur de celle-ci ----- It is forbidden to modify loop condition inside the loop.	Oui
COM.DATA.NotUsed	Fortran : Toute variable déclarée doit être utilisée, sinon une erreur est remontée. ----- Fortran : Every declared variable should be used. Shell : Toute variable déclarée doit être utilisée. Limitation : Les assignments dans les options de la commande awk peuvent engendrer des faux-positifs. Une variable est considérée comme utilisé si elle on l'utilise avec \${variable}. ----- Shell: Every declared variable should be used. Limitations: a variable is considered as used if used with \${variable}	Oui
COM.DATA.Using	Interdiction de réutiliser un objet local dans des traitements de type différent. ----- It is forbidden to reuse local object in several different treatments.	Non
COM.DESIGN.Alloc	Fortran : L'allocation et la desallocation des ressources doit être dans le même niveau (fonction, subroutine, ...). Chaque fois que le	PC¹

¹ Les ressources vérifiés sont les blocs de mémoire, pas les fichiers. Si le développeur encapsule les allocations et desallocations dans les subroutines, l'application remonte une erreur.

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 18/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
	<p>mot 'DEALLOCATE' est trouvé, le programme vérifie qu'il y a le mot 'ALLOCATE' et que les deux utilisent la même ressource.</p> <p style="text-align: center;">-----</p> <p>Fortran : Allocation and deallocation of ressources should be in the same level. For each "DEALLOCATE" keyword found, i-Code CNES checks that the keyword "ALLOCATE" is also found and checks that they are relative to the same resource.</p>	
	<p>Shell : non applicable / not applicable</p>	Non
COM.DESIGN.ActiveWait	<p>Fortran : dans une boucle, les instructions SLEEP, WAIT et PAUSE sont interdites.</p> <p style="text-align: center;">-----</p> <p>Fortran : in a loop, SLEEP, WAIT and PAUSE instructions are forbidden.</p>	Oui
	<p>Shell : La boucle WHILE [1] et le mot READ sont interdits</p> <p style="text-align: center;">-----</p> <p>Shell : WHILE[1] and keyword READ are forbidden.</p>	Oui
COM.FLOW.Abort	<p>Fortran : Le mot STOP est interdit.</p> <p style="text-align: center;">-----</p> <p>Fortran : Keyword STOP is forbidden</p> <p>Shell : Les mot KILL, PKILL et KILLALL sont interdit Limitation : Ne prend pas en compte les return,break et exit, même s'ils peuvent interrompre l'exécution de commandes.</p> <p style="text-align: center;">-----</p> <p>Shell : Keyword KILL, PKILL and KILLALL are forbidden.</p> <p>Limitation: i-Code CNES do not consider keyword "RETURN", "BREAK" and "EXIT".</p>	Oui
COM.FLOW.BooleanExpression	<p>Dans une instruction conditionnelle (IF, DO) il n'est pas possible de définir plus de cinq expressions conditionnelles (AND, OR, NEQV, XOR, EQV, NOT, LT, <, LE, <=, GT, >, GE, >=, EQ, ==, NE, /=).</p> <p style="text-align: center;">-----</p> <p>In a condition (IF,DO), maximum number of operands is 5 (AND, OR, NEQV, XOR, EQV, NOT, LT, <, LE, <=, GT, >, GE, >=, EQ, ==, NE, /=)</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 19/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
COM.FLOW.CaseSwitch	Fortran77 : Non Applicable / not applicable	Non
	Fortran90 : Obligation de finir l'instruction SWITCH avec DEFAULT, afin de traiter tous les cas possibles. ----- Fortran90 : SWITCH instruction should define the DEFAULT case as last one.	Oui
	Shell : Obligation de finir l'instruction CASE avec *), afin de traiter tous les cas possibles ----- Shell : CASE instruction should finish with *)	Oui
COM.FLOW.CheckArguments	Fortran : Obligation de contrôler les paramètres passés à un programme ----- Fortran : Every paramaters should be checked.	Non
	Shell : voir /see SH.FLOW.CheckArguments	
COM.FLOW.CheckCodeReturn	Fortran : Obligation de tester tous les retours de fonction ----- Fortran : Function return values should be checked.	Oui
	Shell : voir/see SH.FLOW.CheckCodeReturn	Non
COM.FLOW.CheckUser	Fortran : Obligation de vérifier l'identité de l'utilisateur qui exécute un programme ----- Fortran : The identity of the user running the program should be checked.	Oui
	Shell : voir SH.FLOW.CheckUser	Non
COM.FLOW.Exit	Fortran : Interdiction d'implémenter plusieurs points de sortie dans les fonctions, procédures ou méthodes. ----- Fortran : Function, procedure and method should have a unique exit point.	Oui
	Shell : Non implémenté / not available	Non

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 20/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
COM.FLOW.ExitLoop	Interdiction d'implémenter plus d'une sortie dans les boucles. ----- Loops should have a unique exit point.	Oui
COM.FLOW.FileExistence	Fortran : Avant d'ouvrir ou créer (mot clé OPEN, READ, WRITE) un fichier, doit apparaître l'instruction INQUIRE avec le flag EXIST sur le même fichier. ----- Fortran : File opening (OPEN, READ, WRITE) should be done after using INQUIRE and EXIST instruction of this file.	Oui
	<u>Shell :</u> Avant d'accéder à un fichier (> nom_fichier), il faut faire apparaître la vérification : if [! -f \$nom_fichier] <u>Limitations :</u> <ul style="list-style-type: none"> • Pas de détection dans les commandes \$(...) ou `...` • Une variable (or redirections standards) sur laquelle est réalisée une redirection peut être interprétée comme un fichier ----- Shell: A file access (> file_name) should be preceded by the check : if [! -f \$file_name] Limitations: <ul style="list-style-type: none"> • No detection in commands \$(...) or `...` • A redirection or a redirected variable could be considered as a file 	Oui
COM.FLOW.FilePath	Dans l'instruction OPEN, il est interdit d'utiliser directement le nom du fichier (fichier.txt). Le chemin d'accès doit être défini au travers d'une variable qui contient le chemin vers le fichier. ----- In the OPEN instruction, it is forbidden to use directly the file name (file.txt). Path should be defined through a variable.	Oui
	Shell : Non implémenté / not available	Non
COM.FLOW.Recursion	Fortran77 : Non Applicable / not applicable	Non
	Fortran90 : Interdiction d'utiliser la récursivité. En Fortran, une fonction récursive est définie comme suit : RECURSIVE FUNCTION (params)	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 21/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
	<p style="text-align: center;">-----</p> <p>Fortran 90 : Recursivity is forbidden. In Fortran, i-Code CNES checks that « RECURSIVE FUNCTION(params) » is not found.</p>	
	Shell : Non implémenté / not available	Non
COM.INST.BoolNegation	<p>La double négation est interdite sur les expressions booléennes. Les négations sont définies avec le mot. NOT. Donc les expressions suivantes ne sont pas permises :</p> <p style="text-align: center;"> .NOT. (.NOT. a) -> (a) .NOT. (a .AND. .NOT. b) -> .NOT. a .OR. b </p> <p style="text-align: center;">-----</p> <p>Double negation is forbidden on Boolean expressions. Negations are defined with the keyword « NOT ». Following expressions are not allowed:</p> <p style="text-align: center;"> NOT. (.NOT. a) -> (a) .NOT. (a .AND. .NOT. b) -> .NOT. a .OR. b </p>	Oui
	Shell : Non implémenté / not available	Non
COM.INST.Brace	<p>Fortran : Toute expression doit être parenthésée, ainsi le nombre de parenthèses ouvertes doit être supérieur ou égal au nombre d'opérateurs utilisés (+, -, *, /, **)</p> <p style="text-align: center;"> a +b * c -> a + (b * c) </p> <p style="text-align: center;">-----</p> <p>Fortran : Every expression should be braced.</p>	Oui
	Shell : Non implémenté / not available	Non
COM.INST.CodeComment	<p>Fortran : Interdiction de commenter le code. Tout mot clé (ASSIGN, BACKSPACE, BLOCK DATA, CALL, ...) dans une ligne de commentaire est une erreur. Le header (commentaires juste avant ou juste après de la déclaration de la fonction ou subroutine) peut contenir ces mots.</p> <p style="text-align: center;">-----</p> <p>Fortran : Code comments are forbidden. I-Code CNES checks all keywords in comments, except in the header.</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 22/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
	<p>Shell : Interdiction de commenter le code. Le premier mot après le symbole de commentaire (#), ne doit pas être un mot clé (cd, date,...), ni une affectation de variable (var=).</p> <p>Limitation : un texte courant commençant par un mot clé du langage sera incorrectement détecté comme du code commenté.</p> <p>Ex : # date de mise a jour # set the starting value</p> <p style="text-align: center;">-----</p> <p>Shell : Code comments are forbidden. I-Code CNES checks all keywords in comments (after #).</p> <p>Limitation: False positive will be raised due to common word used as keyword, for example "date".</p>	Oui
COM.INST.GOTO	<p>Fortran: L 'instruction GO TO est interdite.</p> <p style="text-align: center;">-----</p> <p>Fortran : "GO TO" is forbidden</p>	Oui
	<p>Shell: non applicable / not applicable</p>	Non
COM.INST.Line	<p>Fortran 77 : non applicable / not applicable</p>	Non
	<p>Fortran 90 et Shell : Chaque ligne doit contenir maximum une expression. En Fortran il est possible d'implémenter plusieurs instructions dans une ligne grâce au point-virgule.</p> <p style="text-align: center;"> a = b + c ; d = e * f -> a = b + c d = e * f </p> <p style="text-align: center;">-----</p> <p>Fortan 90 and shell: Each line should be limited to only one expression.</p>	Oui
	<p>Shell : Interdiction de réaliser plusieurs instructions sur une même ligne.</p> <p>Limitation : L'utilisation de mots clefs dans les messages de commandes non encadrés de guillemets peut engendrer des erreurs sur l'analyse du fichier.</p> <p style="text-align: center;">-----</p> <p>Shell Limitation: The use of keywords in command messages without using « » could produce false positive.</p>	

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 23/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
COM.INST.LoopCondition	Fortran : Dans une instruction de boucle, les comparaisons d'égalité (.EQ., ==) ou de différence (!=, .NE.) sont interdites. ----- Fortran: In a loop instruction, comparison using (.EQ., ==, .NE., \=) are forbidden.	Oui
	Shell : Non implémenté / not available	Non
COM.NAME.Homonymy	Fortran : Une variable doit d'avoir un nom unique. Chaque fois qu'une variable est trouvée, le programme vérifie que le nom de cette variable n'est pas déjà utilisé dans le programme. ----- Fortran : Variable name should be unique.	Oui
	Shell : Non implémenté / not available	Non
COM.PRES.Data	Obligation de commenter par une description détaillée les objets importants. ----- Important objects should be well commented.	Non
COM.PRES.Indent	Le code doit être indenté par espaces. Une ligne doit commencer à la même colonne que la ligne précédente. Après l'instruction DO, IF, WHILE, WHERE, SELECT et TYPE la ligne doit commencer dans une colonne supérieure, à l'exception de la fin de l'expression (dénnoté par END). Exemple : <pre style="color: green;"> DO i = 2, nb somme = somme + x(i) IF (isnan(somme)) THEN print *, 'somme is a NaN' moy = -1.0 END IF END DO </pre> ----- Source code should be indented with space (see example above)	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 24/59
--------------------------	--------------------------------	---

Rule	Rule description	i-Code Check
	Shell : Non implémenté / not available	Non
COM.PRES.LengthLine	<p>Une ligne de code doit contenir un maximum de 100 caractères. Le caractère 101 doit être écrit à la ligne suivante.</p> <p style="text-align: center;">-----</p> <p>A line of source code should contain 100 characters maximum.</p>	Oui
	Shell : Non implémenté / not available	Non
COM.PROJECT.Analyser	<p>Obligation de passer un outil d'analyse statique sur tous les codes sources d'un projet.</p> <p style="text-align: center;">-----</p> <p>All source code should be analyzed with checkers.</p>	Non
COM.PROJECT.CodeCloning	<p>Interdiction de dupliquer / cloner du code.</p> <p style="text-align: center;">-----</p> <p>Duplicated code is forbidden.</p>	Non
COM.PROJECT.Header	<p>Fortran : Obligation de définir et d'appliquer les entêtes/cartouche de chaque module et fonctions en début de projet.</p> <p><i>Shell : nommer COM.PRES.Header</i></p> <p style="text-align: center;">-----</p> <p>Fortran : Headers are mandatory for each module and function</p>	Oui
COM.PROJECT.Warnings	<p>Obligation d'afficher tous les warnings et de les corriger</p> <p style="text-align: center;">-----</p> <p>Warning should be displayed and fixed.</p>	Non
COM.TYPE.Expression	<p>Fortran : Dans une expression (une expression est définie par un opérateur comme +, -, *, /, **), Les variables doivent être de même type : soit REAL, soit INTEGER, etc.</p> <p style="text-align: center;">-----</p> <p>Fortran : in a expression (defined by operator like +, -, /, *, **) all variables should have the same type : REAL, INTEGER, ...</p>	Oui
	Shell : Non implémenté / not available	Non
	<i>Shell : Les variables d'une même expression doit être du même type : integer ou char/string définis selon première initialisation.</i>	Non

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 26/59
--------------------------	--------------------------------	---

4.2. SPECIFIC RULES

4.2.1. FORTRAN 77

Règle	Vérification	Couverture au standard
F77.BLOC.Common	Interdiction d'utiliser des COMMON blanc. ----- White COMMON are forbidden.	Oui
F77.BLOC.Else	Dans une instruction IF, le dernier ELSE IF doit toujours être suivi d'un ELSE. ----- In an instruction IF, the last ELSE IF should be follow by ELSE.	Oui
F77.BLOC.File	Obligation d'utiliser les instructions OPEN et CLOSE pour accéder aux fichiers. ----- File access whould be done using OPEN and CLOSE instructions.	Non
F77.BLOC.Function	Obligation d'utiliser les parenthèses d'argument pour l'instruction FUNCTION, même s'il n'y a pas d'argument ----- Parameter's braces sohuld be used when defining a FUNCTION whenever there is no parameter.	Oui
F77.BLOC.Loop	Les boucles DO imbriquées doivent avoir des indicateurs de fermeture différents. Ce n'est pas possible de partager le label. ----- Nested DO loop should have different end instructions (one for each).	Oui
F77.DATA.Array	Obligation de déclarer explicitement les dimensions des tableaux. Par contre, est possible d'utiliser la notation * pour la dernière dimension mais toujours avec la justification d'un commentaire avant. A(*), A(4, *), A(4, *, *), A(4, 4, *), mais A(*, 4)	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 27/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	<p style="text-align: center;">-----</p> <p>Arrays dimension should be declared explicitly. The use of * is tolerated for the last one if justified with a comment.</p>	
F77.DATA.Common	<p>Obligation d'utiliser l'instruction INCLUDE pour déclarer les COMMON dans les unités de programmes qui les référencent.</p> <p style="text-align: center;">-----</p> <p>INCLUDE must be used to declare COMMON in program units referencing it.</p>	Oui
F77.DATA.Double	<p>Dans une initialisation de constante ou dans l'évaluation d'une expression arithmétique, l'utilisateur souhaite que cette constante soit évaluée en double précision, la présence d'un exposant double précision (lettre D) est obligatoire.</p> <p style="text-align: center;">-----</p> <p>In a constante initialisation or in an expression evaluation, the use of (D) (as double precision is mandatory.</p>	Oui
F77.DATA.Initialisation	<p>Obligation d'initialiser toutes les variables avant leurs utilisations avec l'instruction DATA ou BLOCKDATA.</p> <p style="text-align: center;">-----</p> <p>Vairable initialisation should be done using DATA or BLOCKDATA</p>	Oui
F77.DATA.IO	<p>Les unités logiques implicites définies par * sont interdites.</p> <p>READ (*,f) [iolist], READ f [,iolist], WRITE (*,f) [iolist], PRINT f [,iolist]</p> <p style="text-align: center;">-----</p> <p>Implicit logical units defined by * are forbidden.</p>	Oui
F77.DATA.LoopDo	<p>Obligation d'utiliser un type ENTIER comme paramètre de contrôle des boucles DO.</p> <p style="text-align: center;">-----</p> <p>DO loop condition parameter should use INTEGER.</p>	Oui
F77.DATA.Parameter	<p>Interdiction d'utiliser des constantes, expression calculé ou appel de fonction comme paramètres de fonction.</p> <p>CALL function (3, x*y, f(z), var)</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 28/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	<p style="text-align: center;">-----</p> <p>The use of constants, computed values or function calls as function parameters is forbidden.</p>	
F77.ERR.OpenRead	<p>Obligation des tester le statut de retour des instructions OPEN et READ, de préférence à l'aide du paramètre " IOSTAT = ", et vérifier le value de cette variable.</p> <p style="text-align: center;">-----</p> <p>The check of eturn status for instructions OPEN and READ is mandatory. It is recommended to use "IOSTAT = " to do it.</p>	Oui
F77.INST.Assign	<p>Interdiction d'utiliser l'instruction ASSIGN.</p> <p style="text-align: center;">-----</p> <p>ASSIGN instruction is forbidden.</p>	Oui
F77.INST.Dimension	<p>Interdiction d'utiliser l'instruction DIMENSION.</p> <p style="text-align: center;">-----</p> <p>DIMENSION instruction is forbidden.</p>	Oui
F77.INST.Equivalence	<p>Interdiction d'utiliser l'instruction EQUIVALENCE.</p> <p style="text-align: center;">-----</p> <p>EQUIVALENCE instruction is forbidden.</p>	Oui
F77.INST.Function	<p>Il faut utiliser l'instruction FUNCTION avec une déclaration explicite de type, à la définition de la fonction.</p> <p style="text-align: center;">-----</p> <p>FUNCTION instruction should be used with an explicite type declaration, at function definition.</p>	Oui
F77.INST.If	<p>Interdiction d'utiliser le IF arithmétique :</p> <p style="text-align: center;">IF (Expression arithmétique) e1,e2,e3</p> <p>Où les « eN » sont des étiquettes.</p> <p style="text-align: center;">-----</p> <p>Arithmetic IF is forbidden.</p>	Oui
F77.INST.Include	<p>Avec une instruction INCLUDE, le fichier inclus, ne peut pas inclure instructions exécutables (ASSIGN, GOTO, IF, ELSE, CONTINUE, STOP, PAUSE ; DO, READ, WRITE, PRINT,</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 29/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	REWIND ; BACKSPACE, ENDFILE, OPEN, CLOSE, INQUIER, CALL, RETURN, END) ----- If instruction INCLUDE is used, the included file should not contain executable instructions. (ASSIGN, GOTO, IF, ELSE, CONTINUE, STOP, PAUSE ; DO, READ, WRITE, PRINT, REWIND ; BACKSPACE, ENDFILE, OPEN, CLOSE, INQUIER, CALL, RETURN, END)	
F77.INST.Pause	Interdiction d'utiliser l'instruction PAUSE. ----- PAUSE instruction is forbidden.	Oui
F77.INST.Return	L'instruction RETURN(i) est interdite dans les sous- programmes. ----- RETURN(i) instruction is forbidden in subprogram.	Oui
F77.INST.Save	Interdiction d'utiliser l'instruction SAVE hormis pour des variables locales avec une justification par commentaire. ----- SAVE instruction is forbidden except for local variables with justification in a comment.	Oui
F77.MET.Line	Interdiction de dépasser 72 caractères par ligne. ----- Each line should contain less than 72 characters.	Oui
F77.NAME.GenericIntrinsic	Obligation d'utiliser les noms génériques des fonctions intrinsèques. ----- The use of generic names for intrinsic functions is mandatory.	Oui
F77.NAME.Intrinsic	Interdiction de réutiliser les noms des fonctions intrinsèques. Quand une fonction définie par le développeur a le même nom qu'une fonction intrinsèque (définie aux standards), l'application lance une erreur. -----	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 30/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	Intrinsèque function names reuse is forbidden.	
F77.NAME.KeyWords	Interdiction de réutiliser les mots-clés u Fortran77 pour les variables ----- Fortran 77 keywords should not be reused for variable names.	Oui
F77.NAME.Label	Restriction des étiquettes aux instructions FORMAT et CONTINUE. -----	Oui
F77.PROTO.Declaration	Obligation de déclarer les fonctions externes (lesquelles qui ne sont pas dans le même fichier) par le mot EXTERNAL avant de leur utilisation. ----- External functions should be declared with EXTERNAL keyword.	Oui
F77.REF.IO	Obligation d'identifier les unités logiques par un nom symbolique. READ (5, *) NOMBRE -> READ (STDIN, *) NOMBRE ----- Logical unit should be identified with symbolic name.	Oui
F77.REF.Open	Obligation de définir les paramètres FILE, STATUS et POSITION de l'instruction OPEN ----- OPEN parameters FILE, STATUS and POSITION should be defined.	Oui
F77.REF.Parameter	Interdiction de transmettre comme paramètre d'une subroutine les variables qui sont déjà dans un bloc COMMON accessible par la subroutine et le programme qui l'appel. COMMON /CONTROL/ A, B, C, D ... PROGRAM ESSAI	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 31/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	<pre style="color: red;">CALL MY_SUB1 (A, B, C, D) ... END PROGRAM ESSAI SUBROUTINE MY_SUB1 (C _A, B, _C, C_D) ----- It is forbidden to set as subroutine parameter variables which are already in a COMMON bloc accessible from the subroutine and the program which calls it.</pre>	
F77.TYPE.Basic	<p>Obligation d'utiliser les types standards (INTEGER, REAL, DOUBLE PRECISION, COMPLEX, LOGICAL, CHARACTER) uniquement. Autres types non standards seront considérées erreurs INTEGER*4, LOGICAL*n</p> <p style="text-align: center;">-----</p> <p>Only standard types (INTEGER, REAL, DOUBLE PRECISION, COMPLEX, LOGICAL, CHARACTER) are allowed.</p>	Oui
F77.TYPE.Hollerith	<p>Les données et les constantes de type HOLLERITH sont interdites.</p> <p>Une donnée Hollerith est de la forme : numeroH par exemple 0.8H</p> <p style="text-align: center;">-----</p> <p>Data and constant of type Hollerith are forbidden.</p>	Oui

4.2.2. FORTRAN 90

Règle	Vérification	Couverture au standard
F90.DESIGN.Free	<p>Obligation de libérer la mémoire allouée dans le même niveau conceptuel.</p> <p style="text-align: center;">-----</p> <p>Allocated memory should be free in the same conceptual level.</p>	Non

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 32/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
F90.DESIGN.Include	<p>L'inclure d'un fichier est interdit. Si l'INCLUDE contient un fichier écrit en F90, le programme retourne une erreur.</p> <pre style="text-align: center;"> INCLUDE 'file_to_include.F90' ----- File inclusion is forbidden. </pre>	Oui
F90.DESIGN.Interface	<p>Le contenu des modules doit être limité aux clauses USE, PRIVATE et PUBLIC</p> <pre> MODULE interface_syslog Implicit none PRIVATE Interface Subroutine f_syslog(cdata) USE message_syslog Type(opendata_type) End subroutine End interface PUBLIC f_syslog End module interface_syslog ----- Modules should only contains USE, PRIVATE and PUBLIC clauses. </pre>	Oui
F90.DESIGN.IO	<p>Le nombre d'unité dans une fonction OPEN doit dépendre d'une autre fonction ou un tableau.</p> <pre> Integer :: f_unit = 15 integer :: f_unit OPEN (UNIT = f_unit, ...) -> f_unit = getNumber() OPEN(UNIT = f_unit, ...) ----- </pre>	PC²

² La règle demande de vérifier ces trois cas : 1) Des primitives d'allocation et de libération de numéros d'unité. 2) Une primitive de réservation d'un numéro d'unité donné, pour permettre l'utilisation de sous-programmes Fortran 77 utilisant des numéros fixés. 3) Des constantes nommées pour l'entrée et la sortie standard. Notre vérification inclut les options 2 et 3.

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 33/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
F90.DESIGN.Obsolete	<p>In an OPEN function, units number should depend on another function or array.</p> <p>Cette règle vérifie les clauses suivantes :</p> <ul style="list-style-type: none"> - Ne pas utiliser le GOTO calculé. - Ne pas utiliser la syntaxe : CHARACTER*N - Ne pas utiliser le IF arithmétique - Dans une boucle DO, ne pas utiliser de variables réelles ni comme indice, ni pour les bornes de l'intervalle de contrôle, ni pour le pas d'incréméntation. - Ne pas utiliser de terminaison de boucle DO autre que END DO ou CONTINUE. - Ne pas faire de branchements sur ENDIF. - Ne pas utiliser l'instruction PAUSE. - Ne pas utiliser l'instruction GOTO assigné. - Ne pas utiliser l'affectation d'étiquette de FORMAT - Ne pas utiliser le descripteur H (Hollerith) dans les formats. <p style="text-align: center;">-----</p> <p>This rule checks the following points:</p> <ul style="list-style-type: none"> • Do not use computed GO TO • Do not use syntax: CHARACTER*N • Do not use arithmetic IF • In a DO loop, do not use REAL variables, neither as indice, nor as interval limits, nor as increment step • Do not exit a DO loop with other than END DO or CONTINUE • Do not branch on ENDIF • Do not use PAUSE • Do not use assigned GOTO • Do not use FORMAT label • Do not use descriptor H (Hollerith) in formats 	Oui
F90.BLOC.File	<p>Tout fichier ouvert doit être fermé. Le programme cherche l'instruction CLOSE pour chaque fichier ouvert en amont.</p> <p style="text-align: center; color: green;"> OPEN (unit = f_unit, ...) ... CLOSE (unit = f_unit, ...) </p> <p style="text-align: center;">-----</p>	Oui

Règle	Vérification	Couverture au standard
	Every open file should be closed.	
F90.DATA.Array	<p>La dimension du tableau doit être respectée. Il faut obligatoirement utiliser les paramètres qui ont été déclarés à la création du tableau lors de l'appel dans une fonction ou une procédure.</p> <p>Exemple :</p> <p style="color: red;">Subroutine s1(tab) -> Subroutine s1(tab, x)</p> <p style="color: red;">Integer :: tab(2) Integer :: tab(x)</p> <p style="text-align: center;">-----</p> <p>Array dimensions should be respected.</p>	Oui
F90.DATA.ArrayAccess	<p>Dans un tableau d'indirection il n'est pas possible de spécifier plusieurs fois le même élément.</p> <p style="text-align: center;">Integer,dimension(3) :: a</p> <p style="text-align: center;">Integer.dimension(3) :: b</p> <p style="text-align: center;">a = (/ 1,1,3 /)</p> <p style="text-align: center;">b(a) = (/ 1,2,3 /)</p> <p style="text-align: center;">-----</p> <p>In an array of indirections, it is not possible to define the same item several times.</p>	Oui
F90.DATA.Constant	<p>Les constantes qui apparaissent dans plusieurs sous-programmes doivent être définies dans un module.</p> <p>Subroutine s1()</p> <p style="color: red;">Real PI = 3,141519</p> <p>End subroutine s1</p> <p style="text-align: right;">module precision</p> <p style="text-align: center;">-> real PI =</p> <p>3.141519</p> <p>Function f1()</p> <p style="text-align: right;">end module precision</p> <p style="color: red;">Real PI = 3,141519</p> <p>End function</p> <p style="text-align: center;">-----</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 35/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	Constants which appear in several subprograms should be defined into a module.	
F90.DATA.ConstantFloat	<p>Les constantes littérales numériques doivent être suivies par le paramètre de sous-type</p> <pre>Integer,parameter :: DOUBLE=selected_real_kind(15) Real (DOUBLE) :: x = 0.1_DOUBLE</pre> <p>-----</p> <p>Numerical constants (see above) should be followed by subtype parameter.</p>	Oui
F90.DATA.Declaration	<p>Il est obligatoire de mettre l'instruction IMPLICITE NONE après chaque en-tête de méthode. De plus chaque variable devra au préalable être déclaré avant leur utilisation. Cette règle n'est pas prise en compte pour les fonctions et les tableaux.</p> <p>-----</p> <p>The use of IMPLICITE NONE instruction is mandatory for each method header.</p> <p>Limitations : this rule is not checked for function and array</p>	Oui
F90.DATA.Float	<p>Est interdit d'utiliser le format * en sortie (instruction WRITE) pour les nombres flottants</p> <pre>Real :: X Write (std_out, *), x</pre> <p>-----</p> <p>It is forbidden to use * output format (WRITE instruction) for REAL</p>	Oui
F90.DATA.Parameter	<p>Les fonctions intrinsèques SELECTED_REAL_KIND et SELECTED_INT_KIND doivent être regroupées dans un module.</p> <pre>MODULE precision Integer, parameter :: DOUBLE = SELECTED_REAL_KIND(15) END MODULE</pre> <p>-----</p> <p>Functions SELECTED_REAL_KIND and SELECTED_INT_KIND should be grouped in the same module.</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 36/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
F90.ERR.Allocate	<p>L'allocation (ALLOCATE) et libération (DEALLOCATE) doit contenir le paramètre STAT. Ensuite, la valeur du STAT doit être testé après l'instruction.</p> <pre style="text-align: center;"> ALLOCATE(x, STAT = iom) IF (iom > 0) THEN ... ----- Allocation and deallocation should contain "STAT" parameter. Then, « STAT » value should be tested. </pre>	Oui
F90.ERR.OpenRead	<p>Les instructions OPEN et READ qui travaille avec des fichiers doivent contenir le paramètre IOSTAT, et vérifier le value de cette variable. Pour vérifier ça, l'instruction OPEN doit contenir l'attribut FILE. Un exemple qui ne respecte pas la règle est :</p> <pre style="text-align: center;"> OPEN (UNIT = FILE_UNIT, FILE = C_ARG) </pre> <p>Parce qu'il n'y a pas d'attribut IOSTAT, et le value de IOSTAT n'est pas testé. Le même exemple correct sera :</p> <pre style="text-align: center;"> OPEN (UNIT = FILE_UNIT, FILE = C_ARG, IOSTAT=IOS) IF (IOS .NE. 0) ... </pre> <p>Pour l'instruction READ, comme l'attribut FILE n'existe pas, la vérification sera sur ces reads que le UNIT désigne une unité logique qui a été ouverte par un OPEN avant a le code. En continuation avec l'antérieur exemple :</p> <pre style="text-align: center;"> READ (*, *) //pas vérifiable, lecture du clavier READ (UNIT = FILE_UNIT, FMT = 9011, IOSTAT = IOS) // verifiable IF (IOS .LT. 0) ... ----- OPEN and READ instructions, which work on files, should contain IOSTAT parameter and check this variable value. To check this, OPEN instruction should contain FILE attribute. </pre>	Oui
F90.INST.Associated	<p>Entre le mot ASSOCIATED et la déclaration il faut avoir l'instruction NULLIFY.</p>	Oui

Règle	Vérification	Couverture au standard
	<p style="text-align: center;">déclaration</p> <pre> graph TD decl[déclaration] --> assoc1[associated] decl --> nullify[nullify] decl --> assoc2[associated] assoc1 --> v1[V] nullify --> assoc3[associated] nullify --> assoc4[associated] assoc3 --> v2[V] assoc4 --> v3[V] assoc2 --> x[X] </pre> <p>Example :</p> <pre> Real, pointer :: ptr ... NULLIFY(ptr) ASSOCIATED(ptr) ----- </pre> <p>Between declaration and ASSOCIATED instruction, it is mandatory to call NULLIFY.</p>	
F90.INST.Entry	<p>L'instruction ENTRY est interdite</p> <pre> Subroutine s1 ENTRY rien ----- </pre> <p>ENTRY instruction is forbidden.</p>	Oui
F90.INST.Equivalence	<p>Interdiction d'utiliser l'instruction EQUIVALENCE.</p> <pre> INTEGER total (3,2) INTEGER sum (6) EQUIVALENCE (sum, total) ----- </pre> <p>EQUIVALENCE instruction is forbidden.</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 38/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
F90.INST.If	<p>L'instruction IF logique est interdite quand est suivi par un mot autre que EXIT, CYCLE, GOTO et RETURN.</p> <pre style="text-align: center;"> IF (x == 0) THEN GO TO 1000 End IF ----- </pre> <p>IF followed by one of these instructions : EXIT, GOTO, CYCLE, RETURN is forbidden.</p>	Oui
F90.INST.Intent	<p>Chaque paramètre des sous-programmes doit avoir le mot clé INTENT à sa déclaration.</p> <pre style="text-align: center;"> Function f1(x, y, z) Integer, INTENT(IN) :: x Integer, INTENT(IN) :: y Integer, INTENT(OUT) :: z ----- </pre> <p>Each subprogram parameter should be declared with INTENT keyword</p>	Oui
F90.INST.Nullify	<p>Après une desallocation il y a obligation d'utiliser l'instruction NULLIFY sur la même unité logique.</p> <pre style="text-align: center;"> DEALLOCATE (C, stat = iom) NULLIFY (C) ----- </pre> <p>After deallocate, the use of NULLIFY into the same logical unit is mandatory</p>	
F90.INST.Only	<p>Interdiction d'utiliser le mot clé ONLY sans commentaire avant qui explique son utilisation.</p> <pre style="text-align: center;"> Use mes_fonctions_intrinseques , ONLY :: getuid -> my_getuid ----- </pre> <p>The use of keyword ONLY without comment is forbidden</p>	Oui
F90.INST.Operator	<p>Ne pas utiliser la notation ancienne pour les opérateurs relationnels. Substituer .EQ., .NE., .LT., .LE., .GT., .GE. pour ==, /=, <, <=, >, >=</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 39/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	<p style="text-align: center;">-----</p> Do not use old notation. Replace .EQ., .NE., .LT., .LE., .GT., .GE. by ==, /=, <, <=, >, >=	
F90.INST.Pointer	<p>Le POINTER est interdit à exception des cas suivantes :</p> <ul style="list-style-type: none"> - pour créer des structures de données complexes (i.e. liste de chaînes, arbre, etc.) ; - pour manipuler des références à des tableaux (référence à un tableau alloué dans un sous-programme, recopie par échange de pointeurs,...) et à des parties de tableaux; - pour utiliser de l'allocation dynamique dans les composants de types dérivés. <p>Le programme lance, donc, une erreur quand il trouve l'attribut POINTER et il référence une variable simple.</p> <p style="color: red;">real, pointer :: ppi</p> <p style="text-align: center;">-----</p> <p>POINTER is forbidden except in those cases:</p> <ul style="list-style-type: none"> • Complex data structure creation • Manage array references • Dynamic allocation into composants with derivated types 	Oui
F90.NAME.GenericIntrinsic	<p>Ne pas se servir des fonctions intrinsèques spécifiques (INT,IFIX,IDINT,REAL,FLOAT,SNGL,ICHAR,CHAR,AINT,DINT,ANINT, DNINT,NINT, IDNINT, IABS,ABS,DABS,CABS,MOD,AMOD,DMOD,ISIGN,SIGN,DSIGN,IDIM,DIM,DDIM,DPROD,MAX0,AMAX1,DMAX1, AMAX0,MAX1,MIN0,AMIN1,DMIN1,AMIN0,MIN1,AIMAG,CONJG,SQRT,DSQRT,CSQRT,EXP,DEXP,CEXP,ALOG,DLOG,CLOG,ALOG10, DLOG10,SIN,DSIN,CSIN,COS,DCOS,CCOS,TAN,DTAN,ASIN,DASIN,ACOS,DACOS,ATAN,DATAN,ATAN2,DATAN2,SINH,DSINH,COSH,DCOSH,TANH,DTANH), utiliser les génériques (INT,REAL,AINT,ANINT,NINT,ABS,MOD,SIGN,DIM,MAX,MIN,SQRT, EXP,LOG,LOG10,SIN,COS,TAN,ASIN,ACOS,ATAN,ATAN2,SINH,COSH,TANH).</p> <p style="color: red;">resultat = AMOD (argument, diviseur)</p> <p style="color: green;">resultat = MOD (argument, diviseur)</p> <p style="text-align: center;">-----</p> <p>Do not use specific functions (see above), use generic ones.</p>	Oui
F90.NAME.KeyWords	<p>Les variables du code ne peuvent être nommées comme les mots clés en Fortran (ALLOCATABLE, ALLOCATE ;</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 40/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard														
	<p>ASSIGN, BACKSPACE, ...). De plus les noms de fonctions doivent être différents des fonctions intrinsèques (ABS, ACHAR, ACOS, ...)</p> <p><i>integer, parameter :: DATA</i></p> <p><i>integer, parameter :: MY_DATA</i></p> <p>-----</p> <p>Fortran keywords should not be used as variable names.</p>															
F90.PROTO.Overload	<p>La surcharge des opérateurs est interdite. Celle-ci est définie par l'instruction INTERFACE OPERATOR (symbole) et ensuite par leur utilisation.</p> <p>-----</p> <p>Overraded operators are forbidden</p>	Oui														
F90.REF.ARRAY	<p>Dans une expression, quand on veut représenter l'utilisation total d'un tableau, il est obligatoire de se servir de la notation (:)</p> <p><i>Y = A*X + B</i></p> <p><i>Y(:) = A(:)*X + B</i> où Y et A sont des tableaux</p> <p>-----</p> <p>The use of a whole array should be represented using the notation (:)</p>	Oui														
F90.REF.Interface	<p>Le sous-programme appelé doit être visible</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;"><i>subroutine Pas_1</i></td> <td style="width: 50%;"><i>subroutine Pas_1</i></td> </tr> <tr> <td><i>call Mouv(3.0, resul)</i></td> <td><i>call Mouv(3.0, resul)</i></td> </tr> <tr> <td><i>end subroutine Pas_1</i></td> <td><i>contains</i></td> </tr> <tr> <td><i>subroutine Mouv(oper0, resul)</i></td> <td><i>-> subroutine Mouv(oper0, resul)</i></td> </tr> <tr> <td><i>...</i></td> <td><i>...</i></td> </tr> <tr> <td><i>end subroutine Mouv</i></td> <td><i>end subroutine Mouv</i></td> </tr> <tr> <td></td> <td><i>end subroutine Pas_1</i></td> </tr> </table> <p>-----</p> <p>Called subprogram should be visible.</p>	<i>subroutine Pas_1</i>	<i>subroutine Pas_1</i>	<i>call Mouv(3.0, resul)</i>	<i>call Mouv(3.0, resul)</i>	<i>end subroutine Pas_1</i>	<i>contains</i>	<i>subroutine Mouv(oper0, resul)</i>	<i>-> subroutine Mouv(oper0, resul)</i>	<i>...</i>	<i>...</i>	<i>end subroutine Mouv</i>	<i>end subroutine Mouv</i>		<i>end subroutine Pas_1</i>	Oui
<i>subroutine Pas_1</i>	<i>subroutine Pas_1</i>															
<i>call Mouv(3.0, resul)</i>	<i>call Mouv(3.0, resul)</i>															
<i>end subroutine Pas_1</i>	<i>contains</i>															
<i>subroutine Mouv(oper0, resul)</i>	<i>-> subroutine Mouv(oper0, resul)</i>															
<i>...</i>	<i>...</i>															
<i>end subroutine Mouv</i>	<i>end subroutine Mouv</i>															
	<i>end subroutine Pas_1</i>															
F90.REF.Label	<p>L'END doit être suivi par le type (FUNCTION, SUBROUTINE, ...) et le nom</p>	Oui														

Règle	Vérification	Couverture au standard
	<pre> function f ... end function -> function f ... end function f </pre> <p>-----</p> <p>END keyword should be followed by the type (function, subroutine...) and the name.</p>	
F90.REF.Open	<p>Toute instruction OPEN doit avoir les paramètres FILE, STATUS, IOSTAT et POSITION</p> <pre> OPEN (UNIT=f_unit, FILE=c_args, STATUS='old', POSITION='rewind', IOSTAT=ios) </pre> <p>-----</p> <p>Every OPEN instruction should have parameters FILE, STATUS, IOSTAT and POSITION</p>	Oui
F90.REF.Variable	<p>Une même variable doit être référencée sous le même nom dans un sous-programme.</p> <pre> Call incr(i) ... subroutine incr(j) i = i + 1 end subroutine incr -> call incr(i) ... subroutine incr(j) j = j + 1 end subroutine incr </pre> <p>-----</p> <p>A variable should be referenced with the same name in subprogram.</p>	Oui
F90.TYPE.Derivate	<p>Toute déclaration de type doit être dans un module.</p> <p>-----</p> <p>Every type declaration should be placed into a module</p>	Oui
F90.TYPE.Integer	<p>Les paramètres INTEGER doivent être suivis par l'expression SELECTED_INT_KIND.</p> <pre> Integer, parameter :: LONG integer, parameter :: LONG = SELECTED_INT_KIND(5) </pre>	Oui

<p align="center">CNES DNO/DA/AQ</p>	<p align="center">i-Code CNES User Manual</p>	<p>Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 42/59</p>
---	--	--

Règle	Vérification	Couverture au standard
	<p align="center">-----</p> <p>INTEGER parameters should be followed by SELECTED_INT_KIND expression.</p>	
F90.TYPE.Real	<p>Les paramètres REAL doivent être suivis par l'expression SELECTED_REAL_KIND.</p> <p>Real, parameter :: LONG</p> <p>real, parameter :: LONG = SELECTED_REAL_KIND(5)</p> <p align="center">-----</p> <p>REAL parameters should be followed by SELECTED_REAL_KIND</p>	<p align="center">Oui</p>

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 43/59
--------------------------	--------------------------------	---

4.2.3. SHELL

Règle	Vérification	Couverture au standard
SH.DATA.Constant	Obligation de définir les constantes en utilisant <code>`typeset -r`</code> . ----- Constants should be defined using <code>`typeset -r`</code> .	Non
SH.DATA.IFS	Interdiction de modifier la variable IFS ----- It is forbidden to modify IFS variable	Oui
SH.DATA.Integer	À la déclaration d'un integer doit apparaitre <code>`typeset -i`</code> ----- Integer declaration should be done with <code>`typeset -i`</code>	Oui
SH.DESIGN.Bash	La première ligne d'n script doit être <code># !/bin/bash</code> , <code># !/bin/ksh</code> ou <code># !/bin/false</code> . ----- Script firs line should be <code># !/bin/bash</code> , <code># !/bin/ksh</code> or <code># !/bin/false</code> .	Oui
SH.DESIGN.Exit	Interdiction pour un programme de prendre fin avant les programmes qu'il a lancés en tâches de fond. ----- It is forbidden for a program to end before the processes it launches.	Non
SH.DESIGN.Options	Le cas <code>getopts</code> doit être suivi par un case où ses options sont évaluées. ----- Getopts should be followed by a case where options are evaluated.	Oui
SH.ERR.Args	Obligation d'afficher un message particulier lorsqu'un script ne reconnaît pas une option. Ce message doit inclure le synopsis d'utilisation du script. ----- It is mandatory to display a message when a script does not recognize an option. This message should include the way to use the script (<code>-help</code>)	Non
SH.ERR.Help	Les options doivent être gérées par un <code>getopts</code> ou <code>getopt</code> suivi par un case. Une de ces options doit être <code>-h</code> ou <code>-help</code> .	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 44/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	<p>Limitation : si un "h" apparaît dans une chaîne de caractères dans le traitement du case, alors que l'option -h n'est pas gérée, il n'y aura pas de violation.</p> <p style="text-align: center;">-----</p> <p>Options should be managed with getopt or getopt followed by a case. One of these cases should be -h or --help</p>	
SH.ERR.NoPipe	<p>Avant le premier pipe d'un script il faut avoir 'set -o pipefail'</p> <p><u>Limitations</u> :</p> <ul style="list-style-type: none"> • Un symbole " " apparaissant dans une chaîne de caractères ou dans les options d'une commande provoquera une violation, sauf suivant un printf ou un sed. • Les utilisations du pipe qui suivent un caractère "#" ne seront pas détectées (ex : grep -v ^# grep "\$POINT_DE_MONTAGE ") <p style="text-align: center;">-----</p> <p>Use 'set -o pipefail' before using pipes.</p> <p>Limitations:</p> <ul style="list-style-type: none"> - The use of " " into a string or command line options will raise an error except in a printf or sed - The use of " " followed by '#' are not detected 	Oui
SH.ERR.String	<p>Dans les cas de if ou while avec les traitements de chaînes de caractères, il faut avoir le traitement de chaînes vides, désignée par</p> <p>[[]]</p> <p><u>Limitation</u> :</p> <p>L'appel de ls à l'intérieur d'une commande \$(...) ou `...` est susceptible d'être ignoré.</p> <p style="text-align: center;">-----</p> <p>The use of « if » or « while » with strings, empty strings should be managed.</p>	Oui
SH.FLOW.CheckArguments	<p>Dans une fonction, la première instruction, sera la vérification des paramètres. L'instruction sera de la forme suivante : if [# -ne 0]</p> <p>Limitation : Un argument d'une même fonction sera signalé autant de fois qu'il est présent dans la fonction</p> <p style="text-align: center;">-----</p> <p>In a function, first instruction should be the parameters check.</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 45/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
SH.FLOW.CheckCodeReturn	Shell : Non implémenté / not available <i>Pour toutes les fonctions existantes et appelées dans le script, faudra de vérifier son return avec l'aide de \$#</i> <i>La fonction CD est aussi considérée.</i>	Non
SH.FLOW.CheckUser	Si le script demande des droits d'administrateur, après la vérification de l'utilisateur il faut demander l'action directe de l'admin. ----- If root rights are needed, the administration intervention should be asked right after the user check	Oui
SH.INST.Basename	Ne pas se servir de \$0, utiliser 'basename \$0' ----- Do not use \$0, replace it by 'basename \$0 '	Oui
SH.INST.Continue	Shell : Non implémenté / not available	Non
SH.INST.Copy	Obligation d'utiliser des arguments de même nature pour les commandes 'cp'. La nature des arguments est soit de type fichier ou soit de type répertoire. ----- 'cp' command should be used with arguments of the same type (file or directory)	Non
SH.INST.Find	L'instruction LS est interdite ----- LS instruction is forbidden	Oui
SH.INST.GetOpts	Cette règle vérifie les options passées comme paramètre dans le script. L'application lance un warning à la dernière ligne s'il ne trouve pas le mot GETOPS dans le script. Ce celui qui permet contrôler les paramètres passes. <u>Limitation :</u> Après l'appel de la commande getopt, la violation de \$1 est ignoré jusqu'à la déclaration d'une nouvelle fonction dans le fichier. ----- GETOPS should be used to check parameters	Partiel
SH.INST.Interpreter	La première ligne d'un script doit être l'accès à l'interpréter.	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 46/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
	<p style="text-align: center;">-----</p> <p>First line should be the access to interpreter.</p>	
SH.INST.Keywords	<p>Une variable ne peut pas être le nom d'un mot clé (if, while, then...)</p> <p><u>Limitation :</u></p> <p>L'appel d'option de commandes possédant un mot clé lèvent de faux-positifs (ex : l'option if de la commande dd).</p> <p style="text-align: center;">-----</p> <p>Keyword should not be used as variable name.</p>	Oui
SH.INST.Logical	<p>Après le symbole logique && ou seulement est permises des instructions 'echo' et 'exit'</p> <p style="text-align: center;">-----</p> <p>After « && » or « », only echo and axit are allowed.</p>	Oui
SH.INST.Move	<p>Interdiction d'écraser un fichier en utilisant la commande 'mv'.</p> <p style="text-align: center;">-----</p> <p>It is forbidden to erase a file using « mv » command.</p>	Non
SH.INST.POSIX	<p>Utiliser les commandes POSIX aux scripts. Liste dans le RNC-CNES-Q-HB-80-516</p> <p><u>Limitation :</u></p> <p>L'appel de la commande écho suivi d'une redirection (ex : echo &>2) peut engendrer des erreurs sur l'analyse.</p> <p style="text-align: center;">-----</p> <p>POSIX commands should be used.</p>	Oui
SH.INST.SetShift	<p>Les instructions SET et SHIFT sont interdites.</p> <p style="text-align: center;">-----</p> <p>Instructions SET and SHIFT are forbidden.</p>	Oui
SH.INST.Variables	<p>Toute variable doit être écrite comme \${nom_variable}</p> <p style="text-align: center;">-----</p> <p>Every variable should be written as : \${variable_name}</p>	Oui

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 47/59
--------------------------	--------------------------------	---

Règle	Vérification	Couverture au standard
SH.IO.Redirect	Les redirections non standards doit être commentées, et le commentaire doit contenir le nom explicite de la redirection. ----- Nonstandard redirections should be commented, the comment should contains the redirection name.	Oui
SH.MET.LimitAWK	Pour toute instruction AWK, ne pas dépasser les 5 actions (denotées par {...}) ----- AWK instruction should not exceed 5 actions.	Oui
SH.MET.LimitSed	Pour toute instruction SED,ne pas dépasser les 5 actions (denotées par -e, --expresion, -f, ...) et dans chaque action ne pas dépasser les 5 lignes. ----- SED instruction should not exceed 5 actions.	Oui
SH.MET.PipeLine	Toute pipeline de ligne de commandes doit être commenté a priori. ----- Every comand pipeline should be commented.	Oui
SH.REF.Export	Shell : Non implémenté / not available	Non
SH.REF.Inheritance	Interdiction pour les scripts SHELL d'hériter des alias ou des fonctions définis par l'utilisateur et d'utiliser des commandes interactives dans un script. ----- Scripts should not inherit alias or user defined functions.	Non
SH.SYNC.Signals	Shell : Non implémenté/ not available	Non

5. COMPUTED METRICS

Metric	File computation
MET.Nesting	<p>Nombre maximum de niveaux d'imbrications de if, switch,while, for, ... dans une fonction/méthode. Ce nombre est 0 s'il y a aucune imbrication..</p> <pre> SUBROUTINE changer_coordonnees(Deplacement, NbPoints, Points) ! ! --- Cette routine effectue une modification de coordonnees sur un tableau de point ! --- en appliquant un deplacement sur les 3 axes x, y et z ! IMPLICIT NONE INTEGER :: c ! colonne INTEGER :: l ! ligne INTEGER, parameter :: NbDim = 3 INTEGER, intent(in) :: NbPoints DOUBLE PRECISION, intent(inout), dimension(NbDim,NbPoints) :: Points DOUBLE PRECISION, intent(in), dimension(NbDim) :: Deplacement ! On applique a chaque point une valeur de deplacement selon les 3 axes 1 do c=1, NbDim, 1 2 do l=1, NbPoints, 1 Points(c, l) = Points(c, l) + Deplacement(c) end do end do END SUBROUTINE </pre> <p>NB.Imbric = 2</p> <p style="text-align: center;">-----</p> <p>Maximum number of nested levels if a function/method. This number is 0 if there is no imbrication.</p>
MET.Cyclomati c	C'est le nombre de décisions du code (nb if, case, while, catch...).

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 49/59
--------------------------	--------------------------------	---

	<pre> SUBROUTINE changer_coordonnees(Deplacement, NbPoints, Points) ! ! --- Cette routine effectue une modification de coordonnees sur un tableau de point ! --- en appliquant un déplacement sur les 3 axes x, y et z ! IMPLICIT NONE INTEGER :: c ! colonne INTEGER :: l ! ligne INTEGER, parameter :: NbDim = 3 INTEGER, intent(in) :: NbPoints DOUBLE PRECISION, intent(inout), dimension(NbDim,NbPoints) :: Points DOUBLE PRECISION, intent(in), dimension(NbDim) :: Deplacement ! On applique a chaque point une valeur de déplacement selon les 3 axes 1 do c=1, NbDim, 1 2 do l=1, NbPoints, 1 Points(c, l) = Points(c, l) + Deplacement(c) end do end do END SUBROUTINE </pre> <p>NB.Cycomatique = 2</p> <p style="text-align: center;">-----</p> <p>Number of decisions into the source code.</p>
MET.LineOfCode	C'est le nombre total les lignes dans du code source du composant logiciel sauf les lignes vides et les lignes de commentaires

```

1  SUBROUTINE changer_coordonnees(Deplacement, NbPoints, Points)
!
! --- Cette routine effectue une modification de coordonnees sur un tableau de poin
! --- en appliquant un deplacement sur les 3 axes x, y et z
2  !
      IMPLICIT NONE
3
4     INTEGER :: c          ! colonne
5     INTEGER :: l          ! ligne
6
7     INTEGER, parameter :: NbDim = 3
8     INTEGER, intent(in) :: NbPoints
9
10    DOUBLE PRECISION, intent(inout), dimension(NbDim,NbPoints) :: Points
11    DOUBLE PRECISION, intent(in), dimension(NbDim) :: Deplacement
12
13    ! On applique a chaque point une valeur de deplacement selon les 3 axes
14    do c=1, NbDim, 1
15        do l=1, NbPoints, 1
16            Points(c, l) = Points(c, l) + Deplacement(c)
17        end do
18    end do
19
20    END SUBROUTINE

```

NB.Line = 15

Number of lines into the source code (except white line and comment line)

MET.RatioCom
ment

C'est la proportion de commentaires dans le code source du composant logiciel. L'entête se prend en compte dans le calcul du taux de commentaire (tant si elle est avant ou après la définition).

<pre> 1 SUBROUTINE changer_coordonnees(Deplacement, NbPoints, Points) 1 ! 2 ! --- Cette routine effectue une modification de coordonnees sur un tableau de poin 3 ! --- en appliquant un deplacement sur les 3 axes x, y et z 4 ! 2 ! IMPLICIT NONE 3 INTEGER :: c ! colonne 4 INTEGER :: l ! ligne 5 INTEGER, parameter :: NbDim = 3 6 INTEGER, intent(in) :: NbPoints 7 DOUBLE PRECISION, intent(inout), dimension(NbDim,NbPoints) :: Points 8 DOUBLE PRECISION, intent(in), dimension(NbDim) :: Deplacement 5 ! On applique a chaque point une valeur de deplacement selon les 3 axes 9] do c=1, NbDim, 1 10] do l=1, NbPoints, 1 11] Points(c, l) = Points(c, l) + Deplacement(c) 12] end do 13] end do 14] 15] END SUBROUTINE </pre> <p>RATE.Comment = 5 / 15 = 0.33</p> <p style="text-align: center;">-----</p> <p>Comment proportion in the source code. Header is taken into account.</p>	
--	--

6. USER MESSAGES

6.1. RULE'S VIOLATION MESSAGES

These messages are displayed into i-Code *CNES violations* when the analyse is done.

6.1.1. REGLES COMMUNES

Règle	Message
COM.DATA.DeclarationOrder	The parameters are not defined in the right order. The order shall be: in, in/out, out.
COM.DATA.FloatCompare	It's not allowed to compare float variables(" <i>variable</i> ") with equality.
COM.DATA.Initialisation	The variable " <i>variable</i> " is used before being initialized.

Règle	Message
COM.DATA.Invariant	The variable " <i>variable</i> " must be defined as constant.
COM.DATA.LoopCondition	The variable " <i>variable</i> " is modified inside the loop.
COM.DATA.NotUsed	The variable " <i>variable</i> " is declared and not used
COM.DESIGN.ActiveWait	This process contains an active wait. SH : There is an active wait in this point.
COM.DESIGN.Alloc	The resource named " <i>variable</i> " has not been allocated and deallocated in the same algorithmic level.
COM.FLOW.Abort	The keyword STOP is not allowed.
COM.FLOW.BooleaenExpression	Using more than five conditions in an expression is not allowed. SH : It is not allowed use five or more conditional expressions in the same instruction.
COM.FLOW.CaseSwitch	A DEFAULT case is needed in a switch case instruction. SH : The default case of the case switch condition is missing.
COM.FLOW.CheckCodeReturn	The return code of the function "function" is not checked.
COM.FLOW.CheckUser	The user identity is not verified in the main program.
COM.FLOW.Exit	There is more than one exit in the function.
COM.FLOW.ExitLoop	There is more than one exit in the loop.
COM.FLOW.FileExistence	The existences of the file "file" must be checked with the instruction INQUIRE before being opened or created. SH : The existence of the file " + name + " has not been checked.
COM.FLOW.FilePath	It is not allowed to use directly the file name. Store the file path in a variable. Use the variable instead.
COM.FLOW.Recursion	The use of recursivity is not allowed.
COM.INST.BoolNegation	Double negation is not allowed.
COM.INST.Brace	Parentheses are needed for readability.
COM.INST.CodeComment	Commented code is not allowed. It shall be suppressed. SH : The keyword " + ytext() + " is used in a comment.

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 53/59
--------------------------	--------------------------------	---

Règle	Message
	A variable is assigned in a comment.
COM.INST.GOTO	The keyword GOTO is not allowed.
COM.INST.Line	More than one instruction per line is not allowed.
COM.INST.LoopCondition	A loop condition shall be written with inequality (.LE.,<=, or .GT.,>=)
COM.NAME.Homonymy	Names must be unique. The name "variable" is already defined in this file.
COM.PRES.Indent	The code is not indented.
COM.PRES.LengthLine	There are more than 100 characters in this line.
COM.PROJECT.Header	<ul style="list-style-type: none"> - No file header existing. This module/function should have a header with a brief description. - No file header (file name not found). This module/function should have a header with a brief description. - The module/function should have a header with a brief description. SH : The function should have a header with a brief description.
COM.TYPE.Expression	Mixed types " <i>type_variable_1</i> " with " <i>type_variable_2</i> " in the same expression

6.1.2. FORTRAN 77

Règle	Message
F77.BLOC.Common	Unnamed COMMON is not allowed.
F77.BLOC.Else	The IF instruction shall finish with an ELSE after the last ELSE IF.
F77.BLOC.Function	When calling a function, the brackets following the function name are mandatory.
F77.BLOC.Loop	Loops shall have distinct ends.
F77.DATA.Array	The dimension of the array "variable" is not well declared. The * shall be used for the last dimension.
F77.DATA.Common	The INCLUDE instruction shall be used to reference the needed common bloc.
F77.DATA.Double	The double precision variable is not correctly initialized. It misses the character D in its declaration.

F77.DATA.Initialization	The variable "variable" shall be initialized with DATA or BLOCK DATA before its use.
F77.DATA.IO	The use of * with logical units is not allowed.
F77.DATA.LoopDo	The control variable in a loop shall be an integer.
F77.DATA.Parameter	"variable" belongs to parameter types forbidden when calling a function: a constant, an expression to be evaluated, a call to another function
F77.ERR.OpenRead	The status of OPEN/READ shall be tested with the parameter IOSTAT.
F77.INST.Assign	The instruction ASSIGN Is not allowed.
F77.INST.Dimension	The instruction DIMENSION Is not allowed.
F77.INST.Equivalence	The instruction EQUIVALENCE is not allowed.
F77.INST.Function	It misses the type declaration in FUNCTION header.
F77.INST.If	The arithmetic if is not allowed.
F77.INST.Include	The executable instruction "variable" is not allowed in the include file.
F77.INST.Pause	The instruction PAUSE is not allowed.
F77.INST.Return	The instruction RETURN(i) is not allowed.
F77.INST.Save	The instruction SAVE is only permitted for local variables
F77.MET.Line	There are more than 72 characters in this line.
F77.NAME.GenericIntrinsic	It should be used the generic name of the intrinsic function instead of "variable"
F77.NAME.Intrinsic	It is not allowed to use the name of an intrinsic function.
F77.NAME.KeyWords	The variable "variable" is a keyword in Fortran77 language.
F77.NAME.Label	The use of labels is not allowed except with the instructions FORMAT and CONTINUE.
F77.PROTO.Declaration	The function "variable" shall be declared.
F77.REF.IO	The logical entities shall be declared using a symbolic name.
F77.REF.Open	The instruction OPEN shall be called with the parameters FILE, STATUS and POSITION.
F77.REF.Parameter	It is not allowed to provide as a parameter the variables of an accessible bloc COMMON. The variable "variable" is used in a wrong way.

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 55/59
--------------------------	--------------------------------	---

F77.TYPE.Basic	"variable" is not a basic type. Basic types are INTEGER, REAL, DOUBLE PRECISION, COMPLEX, LOGICAL and CHARACTER.
F77.TYPE.Hollerith	Type Hollerith is not allowed. "variable" shall be a CHARACTER.

6.1.3. FORTRAN 90

Règle	Message
F90.BLOC.File	The file "variable" is not correctly closed.
F90.DATA.Array	The dimension's array must be declared as parameters' function.
F90.DATA.ArrayAccess	Array " variable1" initialized using other array named "variable2 " with repeated values.
F90.DATA.Constant	The constants shall be declared and initialized in a module.
F90.DATA.ConstantFloat	Float constant "variable" shall be declared using the subtype_parameter: <name>_<subtype_parameter>
F90.DATA.Declaration	The variable must be declared. The sequence IMPLICITE NONE must be declared after the method.
F90.DATA.Float	It is not allowed to use the format * for reals like "variable".
F90.DATA.Parameter	It misses the use of intrinsic function SELECTED_REAL_KIND or SELECTED_INT_KIND for the subtype specification.
F90.DESIGN.Include	Is it possible to use a module instead of this inclusion?
F90.DESIGN.Interface	Interface Module shall only contain: INTERFACE, USE, IMPLICIT instructions as well as PRIVATE or PUBLIC declaration.
F90.DESIGN.IO	The value of the logic unity should be a integer or a variable initialised directly.
F90.DESIGN.Obsolete	The instruction calculated GOTO is not allowed. The instruction PAUSE is not allowed. The alternate return statement is not allowed. There is a branch on an END IF statement. It is not allowed. The use of CHARACTER* is not allowed. The instruction HOLLERITH is not allowed inside FORMAT. Error in "variable" used.

	<p>The instruction ASSIGN contains the label for the FORMAT instruction.</p> <p>Arithmetical IF is not allowed.</p> <p>A DO loop shall end with END DO.</p> <p>The variable "variable" is a real used in a do loop. Use only INTEGER.</p> <p>Each loop shall have its own END DO. Shared END DO is forbidden.</p>
F90.ERR.Allocate	The status of the ALLOCATE or DEALLOCATE instruction is not checked
F90.ERR.OpenRead	<p>- There is no parameter IOSTAT in the OPEN/READ instruction.</p> <p>- The return of IOSTAT is no checked in the OPEN/READ instruction.</p>
F90.INST.Associated	The pointer « variable » is not set to null before the use of the instruction ASSOCIATED.
F90.INST.Entry	The instruction ENTRY is not allowed.
F90.INST.Equivalence	The instruction EQUIVALENCE is not allowed.
F90.INST.If	Logical IF (without THEN and ENDIF) is only allowed with EXIT, CYCLE, GOTO, RETURN statements.
F90.INST.Intent	It misses the attribute INTENT for the parameter "variable"
F90.INST.Nullify	It misses the instruction NULLIFY after the DEALLOCATION of "variable".
F90.INST.Only	The instruction ONLY must be preceded by a comment.
F90.INST.Operator	The symbolic notation (==, /=, <=, <, >=, >) must be used instead of (.EQ., .NE., .LT., .LE., .GT., .GE.). Error in "variable".
F90.INST.Pointer	This use of POINTER is not allowed.
F90.NAME.GenericIntrinsic	Use the generic name of the intrinsic functions instead of "variable".
F90.NAME.KeyWords	The variable "variable" is a keyword in Fortran90 language.
F90.PROTO.Overload	Overloading operator is not allowed. Overload of "variable"
F90.REF.ARRAY	It should be used the notation(:) to specify the entire use of the arrays: "list_variables".
F90.REF.Interface	The function "function" is not visible in this point.

CNES DNO/DA/AQ	i-Code CNES User Manual	Réf : DNO/DA/AQ - 2017.0002478 Date : 24/05/2018 Page : 57/59
--------------------------	--------------------------------	---

F90.REF.Label	It misses the name of the subprogram. It must finish with END TYPE_PROGRAM NAME.
F90.REF.Open	It misses one or more parameters In OPEN instruction. Mandatory parameters are FILE, STATUS, IOSTAT, POSITION.
F90.REF.Variable	The variable "variable" is used with different names inside the subprogram.
F90.TYPE.Derivate	The variable " " must be defined inside the module structure.
F90.TYPE.Integer	It misses the declaration SELECTED_INT_KIND in the initialisation of "variables"
F90.TYPE.Real	It misses the declaration SELECTED_REAL_KIND in the initialisation of "variables"

6.1.4. SHELL

Règle	Message
SH.DATA.IFS	The environment variable IFS can't be modified.
SH.DATA.Integer	The integer variables must be defined using the typeset -i declaration.
SH.DESIGN.Bash	The first line must declare the interpreter (/bin/bash, /bin/ksh or /bin/false)
SH.DESIGN.Options	It is mandatory to use getopts and getopt and to provide the -h, -help, -v and -version options at least.
SH.ERR.Help	The help option (-h or --help) must be implemented.
SH.ERR.NoPipe	When the pipe is used in the script the option set -o pipefail is mandatory.
SH.ERR.String	The empty strings must be taken into account
SH.FLOW.CheckArguments	The number of parameters received has not been checked.
<i>SH.FLOW.CheckCodeReturn</i>	<i>The function's return function_name has not been checked.</i>
SH.FLOW.CheckUser	The user has not been checked.
SH.INST.Basename	The use of the keyword basename before \$0 is mandatory.
<i>SH.INST.Continue</i>	<i>The keyword CONTINUE is not allowed.</i>
SH.INST.Find	The use of LS is not allowed. Use FIND instead.
SH.INST.GetOpts	It is mandatory to use getopts & getopt in the script.
SH.INST.Interpreter	The first line must declare the interpreter

Règle	Message
SH.INST.Keywords	The keywords <i>variable</i> cannot be used as a variable.
SH.INST.Logical	The abbreviation and && must be followed only by ECHO or EXIT.
SH.INST.POSIX	The keyword <i>POSIX_word</i> is not allowed.
SH.INST.SetShift	The keyword SET/SHIFT is not allowed.
SH.INST.Variables	The variable <i>variable_name</i> is not correctly declared (must be declared using \${ } or " " notation)
SH.IO.Redirect	Thenon-standard redirection must be preceded by a comment.
SH.MET.LimitAWK	The AWK expression has more than 5 actions
SH.MET.LimitSed	The SED expression has more than 5 actions/lines
SH.MET.PipeLine	Every pipeline must be preceded by a comment.
<i>SH.REF.Export</i>	<i>The keyword EXPORT is no allowed.</i>
<i>SH.SYNC.Signals</i>	<i>The keyword TRAP must be followed by a variable, not an integer.</i>

7. LIMITATIONS

7.1. POWERSHELL DOES NOT ALLOW ANALYSE LAUNCH

Alternative : use cmd.exe

Issue : <https://github.com/dupuisa/i-CodeCNES/issues/73>

7.2. SHELL: STRINGS SHOULD BE LIMITED WITH « »

Issue: <https://github.com/dupuisa/i-CodeCNES/issues/31>

♦♦♦♦ END OF DOCUMENT ♦♦♦♦