



Esprit EDGE - Support de formation

Sujet : Fonctionnement du Post-Procasseur ESPRIT

Auteur : Enora RIVIER
Révisé par : Ayoub MERABET
Dernière révision : 09/06/2026
Pour toutes questions techniques : sav@delta-first.com
Version de Esprit applicable : Esprit EDGE 2026.1

Table des matières

1	Fonctionnement général et algorithmie du post-processeur	3
1.1	Fonctionnement général	3
1.1.1	Notions lexicales	3
1.1.2	Schéma de principe.....	4
1.2	Algorithmie et logique de programmation d'un script post-processeur.....	5
1.2.1	Contexte de développement d'un post-processeur	5
1.2.2	Algorithmie générale de la génération d'un code ISO	5
1.2.3	Bases de programmation d'un post-processeur et éléments de langage	8

Index des Illustrations

Illustration 1: Schéma de principe du post-processeur	4
Illustration 2: Exemple illustré du schéma de principe du post-processeur	4
Tableau 1 : Explication de l'Algorithme générale qui génère un code ISO... ..	6
Illustration 3: Extrait de fichier post-processeur : System Variable et Machine Mode section ..	8
Illustration 4: Illustration 5: Extrait de fichier post-processeur : Formatable Dimension section	9
Illustration 5: Extrait de post-processeur : Symbolics Codes section	9
Illustration 6: Extrait de fichier post-processeur : exemple section.....	10
Illustration 7: extrait de post-processeur : exemple simple, mouvement linéaire 5 axes	10
Illustration 8: extrait de post-processeur : exemple simple, mouvement linéaire 3 axes	10
Illustration 9: Extrait de post-processeur : exemple avancé	11
Illustration 10: Extrait de post-processeur : exemple de langage forcé	11

1 Fonctionnement général et algorithmie du post-processeur

1.1 Fonctionnement général

1.1.1 Notions lexicales

Voici quelques éléments lexicaux qui seront utilisés dans la suite du document :

Logiciel FAO : Fabrication Assistée par Ordinateur. Le logiciel FAO permet, à partir d'une définition numérique de pièce (fichier de conception en trois dimension, plan numérique, ...), de calculer des parcours outil pour piloter des machines à commande numérique. Il est composé, au minimum, d'un processeur chargé de calculer les parcours et d'un post-processeur.

MOCN : Machine Outil à Commande Numérique. Machine de fabrication mécanique automatisée, pilotée par un contrôleur numérique. Ce contrôleur est programmable via un fichier script (fichier codeISO) humainement lisible et éditable.

Code ISO : langage de programmation des fichiers script pour les MOCN. Ce langage permet d'écrire des fichiers de commande pour la MOCN, que cela soit manuellement (par un opérateur humain) ou automatiquement grâce à une FAO.

Post-Processeur : logiciel de transcription des points de parcours calculés par le processeur FAO. Ce logiciel est paramétrable par un fichier de commande appelé fichier post-processeur.

Fichier post-processeur : fichier nécessaire au paramétrage du post-processeur afin d'adapter la génération de fichier code ISO au contrôleur de la machine MOCN. Ce fichier permet aussi de retranscrire les méthodes de programmation exploitées par l'opérateur-programmeur de la MOCN.

1.2 Algorithmie et logique de programmation d'un script post-processeur

1.2.1 Contexte de développement d'un post-processeur

Les fichiers post-processeurs, de part leur complexité et leur spécificité relative à la machine pilotée, sont développés soit avec le constructeur de la machine directement, soit au pied de la machine du client durant des prestations intitulées : « Réglage et Personnalisation de Post-Processeur ».

Bien que le langage ISO soit basé sur des codes similaires sur la plupart des MOCN, il peut varier selon les constructeurs et les versions des armoires à commande numérique. De plus, les réglages effectués par l'intégrateur de la machine ou par le client multiplient encore les variantes de langage possible pour chaque machine. Il n'est pas rare que deux machines du même constructeur ou de même armoire n'est pas le même fichier post-processeur, même si la cinématique est identique.

1.2.2 Algorithmie générale de la génération d'un code ISO

Afin de générer un code ISO, il est nécessaire de comprendre que le post-processeur suit un scénario fixe et systématique. Le programme va franchir des étapes de génération à travers d'exemples.

Dans le tableau suivant se présente le déroulé d'une pièce simple avec un outil et une opération :

Les exemples intitulés **Hard** sont des points de passage obligatoires et systématiques comme : le début et la fin de programme, les changements d'outil, le début et la fin d'une opération d'usinage. Ainsi, les programmes auront toujours un *ex_mainStart* et un *mainEnd*.

Les exemples intitulés **Soft** sont des points de passage variables en fonction de la morphologie et de la longueur du parcours d'outil. Ainsi, par exemple, un *Ex_Rapid* va être déclenché pour chaque mouvement en rapide à générer et un *Ex_Linear* pour chaque mouvement en avance travail à générer.

Déroulé de l'algorithme	Explicatif
1 Hard EX_STARTCODE ; %_N_1001_MPF	Début de programme d'usinage
2 Hard EX_CYCLEDEFINITION	Exemple d'initialisation
3 Hard EX_MAINSTART N28 MCALL N30 TRANS N32 TRAFEOF N34 G90 G17 G54 M5 N36 FFWON N38 SOFT N40 COMPCAD N42 CFTCP N46 FGROU(X1,Y,Z,B,C) N44 JERK_OFF N48 TOL(0.01) N50 M50 N52 RETRAIT	
4 Hard EX_BEFORECYCLE	
5 Hard EX_FIRSTTOOLCHANGE MSG("OP1:SURFACAGE") N54 T1 D01 N56 M6 N58 TRAORI(2) N60 CUT2DF N62 G54 N64 G642 N66 M26 N68 M28	Premier changement d'outil
6 Hard EX_SETWORKCOOR	
7 Hard EX_CYCLESTART(Contour)	
8 Soft EX_ROTARYSTART(SP2)	
9 Hard EX_CHANGEWORKSYSTEM(SP2) N70 G54 N72 APPROCHE N74 ROT X0 N76 AROT Y45. N78 AROT Z0	
10 Soft EX_RAPID(SP2)	Mouvement en avance rapide
11 Soft EX_RAPID(SP2)	
12 Soft EX_RAPID(SP2)	
13 Hard EX_STARTPOINT(Contour) N80 G54 S16000 M3 M8 M74 N82 G0 Z250. N84 G0 Y-104.866 N86 G0 X51.033 N88 G0 C0 B45.	Initialisation du premier point d'usinage du cycle d'usinage
14 Soft EX_RAPID(Contour) N90 Z65.033	Mouvement en avance rapide
15 Soft EX_LINEAR(Contour) N92 G1 Z61.033 F1500	Mouvement en avance travail
16 Soft EX_LINEAR(Contour) N94 Y92. F2500	
17 Soft EX_CIRCLE(Contour) N96 G2 X67.033 Y92. I8. J0	Mouvement en interpolation circulaire
18 Soft EX_LINEAR(Contour) N98 G1 Y-92.	
19 Soft EX_CIRCLE(Contour) N100 G3 X76.246 Y-92. I4.607 J0	
20 Soft EX_LINEAR(Contour) N102 G1 Y104.866	
21 Soft EX_LINEAR(Contour) N104 Z63.033 F1500	
22 Soft EX_RAPID(Contour) N106 G0 Z65.033	
23 Soft EX_RAPID(Contour) N108 X51.033 Y-104.866	
24 Soft EX_RAPID(Contour) N110 Z63.033	
25 Soft EX_LINEAR(Contour) N112 G1 Z59.033	
26 Soft EX_LINEAR(Contour) N114 Y92. F2500	
27 Soft EX_CIRCLE(Contour) N116 G2 X67.033 Y92. I8. J0	
28 Soft EX_LINEAR(Contour) N118 G1 Y-92.	
29 Soft EX_CIRCLE(Contour) N120 G3 X76.246 Y-92. I4.607 J0	
30 Soft EX_LINEAR(Contour) N122 G1 Y104.866	
31 Soft EX_LINEAR(Contour) N124 Z61.033 F1500	
32 Soft EX_RAPID(Contour) N126 G0 Z65.033	
33 Soft EX_RAPID(Contour) N128 X51.033 Y-104.866	
34 Soft EX_RAPID(Contour) N130 Z61.033	
35 Soft EX_LINEAR(Contour) N132 G1 Z57.033	

36	Soft EX_LINEAR(Contour)	N134 Y92. F2500	
37	Soft EX_CIRCLE(Contour)	N136 G2 X67.033 Y92. I8. J0	
38	Soft EX_LINEAR(Contour)	N138 G1 Y-92.	
39	Soft EX_CIRCLE(Contour)	N140 G3 X76.246 Y-92. I4.607	
	J0		
40	Soft EX_LINEAR(Contour)	N142 G1 Y104.866	
41	Soft EX_LINEAR(Contour)	N144 Z59.033 F1500	
42	Soft EX_RAPID(Contour)	N146 G0 Z65.033	
43	Soft EX_RAPID(Contour)	N148 X51.033 Y-104.866	
44	Soft EX_RAPID(Contour)	N150 Z59.033	
45	Soft EX_LINEAR(Contour)	N152 G1 Z55.033	
46	Soft EX_LINEAR(Contour)	N154 Y92. F2500	
47	Soft EX_CIRCLE(Contour)	N156 G2 X67.033 Y92. I8. J0	
48	Soft EX_LINEAR(Contour)	N158 G1 Y-92.	
49	Soft EX_CIRCLE(Contour)	N160 G3 X76.246 Y-92. I4.607	
	J0		
50	Soft EX_LINEAR(Contour)	N162 G1 Y104.866	
51	Soft EX_LINEAR(Contour)	N164 Z57.033 F1500	
52	Soft EX_RAPID(Contour)	N166 G0 Z65.033	
53	Soft EX_RAPID(Contour)	N168 X51.033 Y-104.866	
54	Soft EX_RAPID(Contour)	N170 Z57.033	
55	Soft EX_LINEAR(Contour)	N172 G1 Z53.033	
56	Soft EX_LINEAR(Contour)	N174 Y92. F2500	
57	Soft EX_CIRCLE(Contour)	N176 G2 X67.033 Y92. I8. J0	
58	Soft EX_LINEAR(Contour)	N178 G1 Y-92.	
59	Soft EX_CIRCLE(Contour)	N180 G3 X76.246 Y-92. I4.607	
	J0		
60	Soft EX_LINEAR(Contour)	N182 G1 Y104.866	
61	Soft EX_LINEAR(Contour)	N184 Z55.033 F1500	
62	Soft EX_RAPID(Contour)	N186 G0 Z250.	
63	Hard EX_CYCLEEND(Contour)		Fin de cycle
64	Hard EX_TOOLCANCEL	N188 M9	
		N190 M5	
		N192 TRAFOOF	
		N194 ROT	
		N196 RETRAIT	
65	Hard EX_MAINEND		
66	Hard EX_ENDCODE	N198 G53 B0 C0	Fin de programme
		N200 M30	

Tableau 1: Explication de l'Algorithme générale qui génère un code ISO

1.2.3 Bases de programmation d'un post-processeur et éléments de langage

Comme expliqué au paragraphe précédent, le post-processeur suit une logique de « points de passages » pour générer le code ISO. Le script est donc à l'image de cette logique. Le fichier post- processeur présente ainsi quatre sections principales :

```

***** System Variables *****
POSTNAME           : SIEMENS
MACHINETOLERANCE   : 0.001
MAXCIRCCLERADIUS   : 9999
MAXNCODE           : 99999999
NCODEDEFAULT       : 2
NCODEINCREMENT     : 2
PROGRAMNUMBERDEFAULT : 1000
REMOVESPACES       : 0
RUNTIMECOMMENTS    : 0
STEPTOLERANCE      : 0.001
UPPERCASECOMMENTS  : 1
ROTARY_CLEARANCE   : ROTARY_ENABLED
TOOLCHANGE_HIGHZ   : 99999.999
SAFE_CYCLESTART    : 0

***** Machine modes *****

COORDINATEMODE     : ABSOLUTE
CIRCLEMODE         : BY360
CENTERMODE         : INCREMENTFROMSTART
MEASUREMENTMODE    : METRIC
ROTARYMODE         : FULL
RPLANEMODE        : CLEARANCE

```

Illustration 3: Extrait de fichier post-processeur : System Variable et Machine Mode section

Tout d'abord, les sections *System variable* et *Machine Mode* permettent un réglage général du post-processeur.

```

32 ***** Formatable Dimensions *****
33
34 SEQUENCENUMBER      : N   Y 4 N N 0   N 4 N N 0   0   - - - -
35 PROGRAMNUMBER      : _   Y 4 N N 0   Y 4 N N 0   0   - - - -
36 XABSOLUTE          : X   N 4 Y N 4   N 5 Y N 3   1   1 N 0 0
37 YABSOLUTE          : Y   N 4 Y N 4   N 5 Y N 3   1   1 N 0 0
38 ZABSOLUTE          : Z   N 4 Y N 4   N 5 Y N 3   1   1 N 0 0
39 ##XCENTERABSOLUTE  : XCENTERABSOLUTEV N 4 Y N 4   N 5 Y N 3   7   1 N 0 7
40 ##YCENTERABSOLUTE  : YCENTERABSOLUTEV N 4 Y N 4   N 5 Y N 3   7   1 N 0 7
41 ##ZCENTERABSOLUTE  : ZCENTERABSOLUTEV N 4 Y N 4   N 5 Y N 3   7   1 N 0 7
42 XCENTERABSOLUTE    : I   N 4 Y N 4   N 5 Y N 3   1   1 N 0 1
43 YCENTERABSOLUTE    : J   N 4 Y N 4   N 5 Y N 3   1   1 N 0 1
44 ZCENTERABSOLUTE    : K   N 4 Y N 4   N 5 Y N 3   1   1 N 0 1
45 DIAMETERCOMPENSATION : D   Y 2 N N 0   Y 2 N N 0   0   1 N 0 0
46 FEEDRATE           : F   N 3 Y N 2   N 5 N N 0   0   1 N 0 0
47 CHIPFEEDRATE       : F_  N 3 Y N 5   N 4 Y N 4   0   1 N 0 0
48 INVERSETIMEFEED    : F__ N 2 Y N 3   N 2 Y N 3   0   1 N 0 0
49 LENGTHCOMPENSATION : D_   Y 2 N N 0   Y 2 N N 0   0   1 N 0 0
50 SPINDLESPEED        : S   N 4 N N 0   N 4 N N 0   0   1 N 0 0
51 TOOLNUMBER          : T   N 2 N N 0   N 2 N N 0   0   1 N 0 0
52 AAXIS               : C   N 4 Y N 3   N 4 Y N 3   1   1 N 0 0
53 BAXIS               : B   N 4 Y N 3   N 4 Y N 3   1   1 N 0 0
54 CIRCULARADIUS       : R_  N 4 Y N 4   N 5 Y N 3   1   1 N 0 0
55 OFFSET              : Q_  N 4 Y N 4   N 5 Y N 3   1   1 N 0 0
56 WORKSYSTEM          : G   N 3 N N 0   N 3 N N 0   0   - - - -

```

Illustration 4: Extrait de fichier post-processeur : Formatable Dimension section

Ensuite, les parties *Formatable Dimension* et *Symbolics Codes* permettent la déclaration des éléments de langage pour la MOCN. Les *Formatable Dimensions* seront les éléments qui présenteront une valeur à leur suite (coordonnées X,Y,Z, par exemple). Les *Symbolics Codes* présenteront tout autre mot clef du langage ISO (instructions de déplacements G0, G1, instructions de sélection de plan d'usinage, etc.)

```

***** Symbolic Codes *****
STRINGCHARACTER      : '
COMMENTSTART         : ;
COMMENTEND           :
UPPERCASECOMMENTS    : 1
MOTIONRAPID          : G0
MOTIONLINEAR         : G1
MOTIONCW             : G2
MOTIONCCW            : G3
DWELL                 : G4
XYPLANE              : G17
ZXPLANE              : G18
YZPLANE              : G19
INCH                  : G20
METRIC                : G21
ZERORETURN           : G28
COMPENSATIONOFF       : G40
COMPENSATIONLEFT     : G41
COMPENSATIONRIGHT    : G42
COMPENSATIONPLUS     : G43
COMPENSATIONMINUS    : G44
LENGTHCANCEL         : G49
COORDINATEPRESET     : G53
WORKSYSTEM1          : G54
WORKSYSTEM2          : G55
WORKSYSTEM3          : G56
WORKSYSTEM4          : G57

```

Illustration 5: Extrait de post-processeur : Symbolics Codes section

Nous trouverons enfin la partie dédiée aux « exemples » comme expliqué au [Paragraphe 1.2.2.](#)

```

431 EX_CYCLESTART      :
432
433 EX_CYCLEEND        : meme_outil=(meme_outil+1)
434
435 EX_COMPENSATION     : BD N G1 COMPENSATIONSIDE* X Y Z C B F
436
437 EX_COMPENSATIONOFF : BD N MOTIONCODE* G40* X Y Z C B F
438
439 EX_COMPOFF_R       : BD N G0 G40* X Y Z C B F
440
441 EX_RAPID            : IF (TC=1) EXITEXAMPLE ENDIF
442
443                   $Rotate
444                   : BD N G0 X Y Z
445
446 EX_LINEAR           : BD N G1 X Y Z C B F
447
448 EX_RAPID_SAXIS     : BD N G0 X Y Z C B
449
450 EX_LINEAR_SAXIS    : BD N G1 X Y Z C B F
451

```

Illustration 6: Extrait de fichier post-processeur : exemple section

Une fois les éléments de langage ISO déclarés dans les parties *Formatable Code section* et *Symbolics Codes section*, les exemples vont expliciter au post-processeur les éléments à écrire et dans quel ordre. Par exemple, la ligne suivante précise que pour un mouvement linéaire, il faut écrire le numéro de ligne (N), l'instruction G1, les valeurs des coordonnées X, Y, X, C, B et la valeur de l'avance F.

```
EX_LINEAR           : N G1 X Y Z C B F
```

Illustration 7: extrait de post-processeur : exemple simple, mouvement linéaire 5 axes

Cette ligne serait en adéquation avec le pilotage d'une machine cinq axes (3 axes linéaires X,Y et Z et 2 axes rotatifs C et B). Si nous avons à piloter une machine 3 axes (X,Y et Z seulement) il serait donc écrit :

```
EX_LINEAR           : N G1 X Y Z F
```

Illustration 8: extrait de post-processeur : exemple simple, mouvement linéaire 3

L'exemple suivant présente d'autres éléments de langage :

```

EX_CIRCLE                : if (dregdiff(3))
                        : BD N WORKPLANE CODE CIRCLEDIRECTION*      X* Y* Z* I*
J* F ##if(nextclfile(189)=1) $Avancetangentielle else F endif
                        : else
                        : BD N WORKPLANE CODE CIRCLEDIRECTION*      X* Y* Z* I*
J* F ##if(nextclfile(189)=1) $Avancetangentielle else F endif
                        : endif

```

Illustration 9: Extrait de post-processeur : exemple avancé

- Tout d'abord, une structure logique *if*, *else*, *endif* est présente dans cet exemple. En fonction de condition définie, il est donc possible de créer des embranchements logiques. Dans l'exemple ci-dessus, il est donc écrit deux choses différentes selon le résultat de *dregdiff(3)*¹ ;
- Des étoiles apparaissent à côté des dimensions X,Y et suivantes, ce qui permet de forcer la sortie des valeurs. En effet, sans ces étoiles, le post-processeur ne ré-écrit pas les dimensions, qui n'ont pas évoluées, afin d'alléger le code. L'exemple ci-dessus n'écrit donc pas systématiquement la valeur de Z sauf s'il y a une variation en Z ;
- Les derniers éléments notables de cet exemple sont les lignes précédées du symbole *##*. Ce double dièse permet d'indiquer au post-processeur d'ignorer tout ce qui suit, jusqu'à la fin de la ligne. Il n'exécutera donc pas le test *if(nextclfile(189)=1) \$Avancetangentielle else F endif*,

Un autre exemple permet de mettre en lumière un autre élément de langage, les lignes dites « écrites endur » :

```

EX_MAINSTART            : BLANKLINE
                        : N 'MCALL'
                        : N 'TRANS'
                        : N 'TRAFOOF'

```

Illustration 10: Extrait de post-processeur : exemple de langage forcé

Les mots écrits entre guillemets simples sont des mots ré-écrit tel quel. Cette fonctionnalité permet d'écrire des fonctions ou mots clefs ISO, comme ici des initialisations machines. Ces éléments, retranscrit automatiquement, sont souvent des éléments permanents de tout programme généré par le post-processeur.

¹ Dregdiff permet de tester la variation d'une dimension, ici la 3ème qui correspond à Z.