



Ductpost

**Полное
Справочное Руководство**

Раздел 1. Файлы Ductpost

Введение

DUCTpost - гибкая система генерации постпроцессоров. Позволяет пользователю станка ЧПУ, очень просто править имеющиеся постпроцессоры и создавать новые.

Нет необходимости тратить время для полного написания постпроцессоров, т.к. в DUCTpost существует своя база из пятидесяти наиболее используемых стоек ЧПУ, которые могут быть изменены и пересохранены используя файлы .opt.

В командной строке Windows, введите: *ductpost-w [управляют типом]> [тип управления].dmp*, после чего DUCTpost сгенерирует файл с текущими настройками по умолчанию. Используя этот файл как основу Вы можете изменить любую его строку или вставить новую.

Файл.opt написан на простом английском языке, строки файла можно писать в любом порядке, однако лучше сохранять логическую последовательность. Этот файл может быть перенесен на любой компьютер.

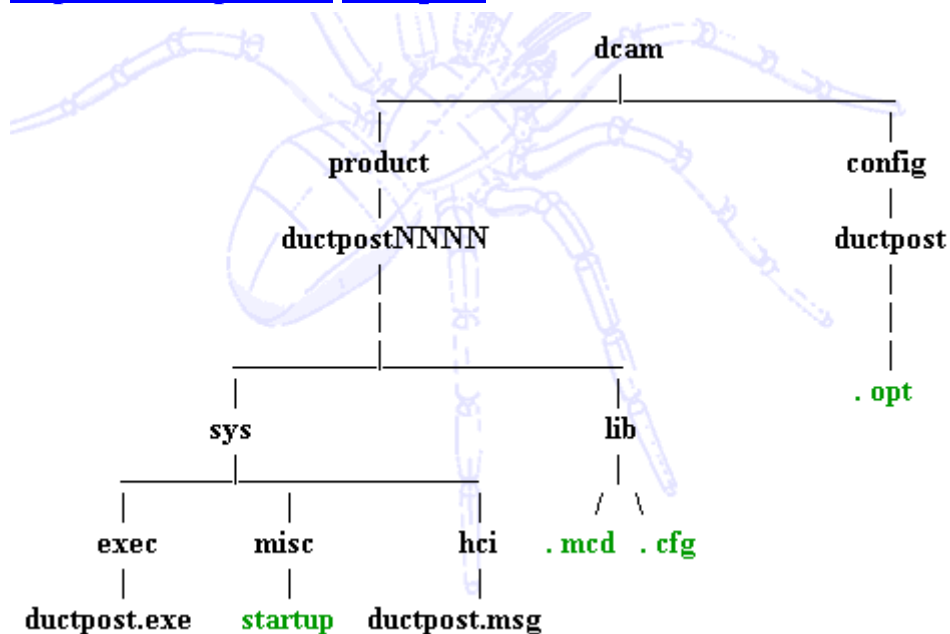
Используя **Файл.opt**, любому слову или группе слов, в файле УП можно присвоить любой формат вывода. Последующие части (начальная строка, смена инструмента, линейные перемещения и т.д.) могут быть определены, например: выводятся или нет номера блоков.

DUCTpost также гибок в записи имени файла. Программа станка ЧПУ, произведенная DUCTpost по умолчанию именуется *jobname.tap*, но может быть *jobname.nc* или *имя задания 01*. Также возможно записать информацию содержащую подготовки рабочего места. Этими средствами управляет файл именуемый *запуск*.

Старый формат *xyz .. cfg* файлы больше не поддержан и будет должен быть преобразован, чтобы использовать **"стандартный"** формат, или один из другого типа управлений, преобразовывая их в файл *опции*.

Старый формат *xyz. cfg* файлов уже не поддерживается и должен быть преобразован в **"стандартный"** формат **Файл.opt**.

Дерево построения Ductpost



Директория **lib** должна включать в себя файлы "**cfg**" или "**mcd**", старших версий Ductpost.

Если **Опция** будут созданы в рабочем каталоге или запуска файлы, то они будут использоваться вместо файлов лежащих в директории **/dcam/config/ductpost**

Файл Startup

Файл Startup постоянно находится в следующем каталоге (**/dcam/config/ductpost/startup**). Однако, это не справедливо для версий от DP1203 где файл Startup **не включен**.

Если файл Startup все же необходим, то его придется создать.

Если файла Startup нет, то вывод значений по умолчанию будет следующим:

Tape file = [**Output Cut-file name**] . **tap**

Set-up file = [**Output Cut-file name**] . **inf**

Print file = [**Output Cut-file name**] . **prt**

ЗАМЕЬТЕ, что PowerMILL игнорирует файл Startup.

Фрмат файла Startup:

Основная форма записи файла Startup по умолчанию:

file part = **output required** (требуемый выход)

file может быть один или " **tape** ", " **setup** ", " **print** "

part может быть " **file** " -для имени файла

" **extension** " -для расширения файла

" **dot** " -для разделителя

output required none (без output (вывода)), или тип " **tty** " для расширения или имени. (" – не всегда требуются, кроме случаев описанных ниже).

Таким образом типичные строки могут быть:

setupfile = " **информация** "

setupextension = " " (может быть " **пробел** ")

(Примечание: если ничего не задано, расширение будет " **. none** ")

setup dot = **none**

Это дает **setup file** вызывать данные без **dot** или **extension** .

Файл Startup не зависит от системы ЧПУ или файла option file пока они не заданы.

Пример файла startup:

setup file = none (см. Примечание 1)

machine f11m4x_AUT3 (см. Примечание 2)

tape extension = "f11"

tape dot = "*"

setup dot = "*"

print extension = "prt" (см. Примечание 3)

machine heid (см. Примечание 4)

tape extension = H

setup extension = " " (см. Примечание 5)

setup dot = *

setup file = комментарий

tape file = HEID

cr (см. Примечание 6)

Пояснения к вышеупомянутому примеру:

1. Эта строка является общей, она отменяет все указанное ниже. Эта установка подавляет вывод setup-файла. (# вывод комментария или удаление если требуется setup sheet) .

2. Эта строка специфически связана с системой ЧПУ, в этом примере Option-файл может иметь несколько систем ЧПУ. Вывод этих данных производится в следующих 4-ех строках (кроме того если строка 1 задана, смотри примечание 1).

используйте :- **ductpost f11m4x_AUT3 ltrbotl1**

3. Print file является выводом CLdata-файла из cut-file и расширение по умолчанию будет **prt**
Для получения print file :-

используйте :- **ductpost -p heid ltrbotl1 (Вывод heid . prt)**

или : **- ductpost heid ltrbotl1 -p (Вывод ltrbotl1 . prt)**

4. Эта строка имеет отношение к системе ЧПУ "**heid**" и не взаимодействует с option-файлами системы ЧПУ "**heid**".

используйте :- **ductpost heid ltrbotl1**

5. Использование двойного space string позволяет подавлять расширение здесь и в других местах. Использование "**none**" не приводит к расширению "**.none**".

6. Последняя строка должна быть пустой by a carriage return from the proceeding entry.

PowerMILL и Startup

Startup-файл игнорируется Постпроцессированием для **Tape output** из PowerMILL и все установки могли быть произведены в Selected Toolpath Output Form (выбранной форме вывода пути инструмента).

Установка расширения по умолчанию "**.tap**" выполняется для определения расширения файла вывода, например: **ltrbotl1 . hd**

Не устанавливайте информационный файл если установлен флажок "**Produce info file**" в поле OptionFile и тогда будет вывод **ltrbotl1 . inf**

ПолучитьCLdata print file, установить выход для **Cutter Location**, и выбрать **Print** в поле Format из поля Output Options. В файле вывода отобразится то что было напечатано в поле имени Output File, например: **ltrbotl1 . pcl or ltrbotl1 . prt**

От версии **DP1320 Startup-файл** бал признан и определение в этих пределах осуществляется опцией Post Processor.

Прогон DUCTpost

DUCTpost может быть запущен двумя способами:

1. Как автономная программа запускаемая с командной строки Shell Window:

ductpost [control name] [name cut-file]

где "**name**" **.cut** это имя cut-file произведенного с помощью DUCT, или POWERMILL.

Ductpost производит программу ЧПУ "**name**".**tap** для системы ЧПУ.

Например:

ductpost fanuc test

постпроцесирование файла **test.cut** , для стойки ЧПУ **fanuc**, с выводом его как **test.tap**. (подробнее см. [switches](#) (переключатели).

Или изнутри:

PowerMill2XXX

PowerMill2XXX

Selected Toolpath Output Form

File Type File Mode

Output File
name.tap

Tape File Options
Machine Option File
/dcam/config/ductpost/deckel11.opt
Produce Info File ☒

Output Options
Part Name
2345-name
Units Format Tool Coordinate

Cutfile Toolpath Options
Add Coolant Command ☒ Coolant Type
Add Load Tool Tool Number
Add Tool Length Command ☒ New Tool Length

Здесь, "**name**".**tap** введена (или только "**name**") в поле **Output File**, выбранный постпроцессор для системы ЧПУ введен в поле **Machine Option File**, и написаны остальные требуемые параметры файла - это **Written**, тогда Ductpost сгенерирует программу ЧПУ "**[name]**".**tap**. Если требуются другие расширения файлов, тогда печатайте, например **name**. "**hd**"

Не заданный информационный файл если флажок "**Produce info file**" не задан в поле OptionFile, тогда вывод будет иметь такой вид **ltrbotl1**. **inf**

Полученный CLdata print file, выводится для **Cutter Location**, и выбор **Print** в поле Format из поля Output Options. Файл вывода отобразит то что напечатано в поле Output File, например **ltrbotl1.pcl**

PowerMill3XXX

С более поздними версиями PowerMill, как видно из показанного ниже окна форма Output не претерпела больших изменений.

По умолчанию в DUCTpost встроен ряд постпроцессоров на системы ЧПУ (более подробно см. Доступные системы ЧПУ) .

Если файл имеющий имя **[control]. opt** присутствует в текущей директории, или в директории **/dcam/config/ductpost**, тогда он будет прочтен первым, и **machine name** в заголовке этого option-файла будет использоваться как контроллер. Однако, что-либо заданное в этом **option-файле** будет использоваться вместо встроенного по умолчанию.

Для более подробной информации см. [Option File](#).

DUCTpost также читает файлы сообщений:

/dcam/product/ductpostxxxx/sys/hci/ductpost . msg

(где " **ductpostxxxx** " номер версии DUCTpost). Здесь содержатся сообщения которые используются DUCTpost, которые при необходимости могут быть изменены (Не рекомендуется)

DUCTpost может также использовать **startup file**: (**/dcam/config/ductpost/startup**) для изменения имени tar-файла заданного по умолчанию и его расширения.

DUCTpost также генерирует файл именованный " **name** " **. inf** , содержащий tooling details, и некоторые комментарии в cut-file. При необходимости, этот Вывод может быть подавлен по средством **startup-файла**.

Переключатели:

Ductpost может быть запущен в автономном режиме с командной строки окна MSDos NT Window, с некоторыми дополнительными параметрами.

ductpost fanuc test -t new.hnc -a

Возможные переключатели:

-a используется если cut-file является **ascii**.

-t *name* используется для изменения имя программы ЧПУ

-i *name* используется для изменения имени ***info-файла***.

-p Вывод ***CLdata*** print file

ductpost -v

-v используется для печати номера версии DUCTpost.

-l (строчная буква L) используется для печати списка доступных систем управления

-w [*control*] используется для написания полной конфигурации по умолчанию, системы управления.

Option-Файл

Файл опций используется для переналадки встроенных стандартных постпроцессоров на системы ЧПУ.

Несколько option-файлов могут быть использованы для каждой системы ЧПУ.

Первая строка должна содержать имя системы ЧПУ.

Option-файл обычно находится в следующей директории: **/dcam/config/ductpost/**

Примеры:

Стандартная система ЧПУ Fidia имеет программы без десятичной точки, два десятичных разряда и без пробелов.

Такой вывод выглядит так:

N10G00X1234Y3456Z2345

N20G01X2000Y3000

N30Z-2000

....

Другой станок Fidia имеет в программах десятичные точки, три десятичных разряда и пробелы.

N10 G00 X12.342 Y34.561 Z23.45

N20 G01 X20. Y30.

N30 Z-20.

...

Для внесения данных изменений option-файл должен быть изменен, следующим образом:

```
machine fidia
define format ( X Y Z )
metric formats
decimal point = true
decimal places = 3
trailing zeros = false
end define
define format all
tape position = 1
end define
end
```

Обратите внимание: Первая строка '**machine control**' и последняя строка '**end**' – **единственные фиксированные строки.**

В качестве отступов может быть использованы один или несколько пробелов, **но нельзя использовать табуляцию.**

После строки '**end**' должна быть **пустая линия.**

Состав Option-файла

Структура описания четырех секционная

Формат каждого слова и порядок слов.

Связь между словами и их значениями (ключи и коды).

Целочисленность, реальность, условное обозначение и логические флажки (метки).

Описание блоков, которые соответствуют старту, концу, смене инструментов, круговым и прямолинейным перемещениям.

Типичный option-файл может содержать следующие строки:

machine fanucom

define format (G1 M1)

leading zeros = true

end define

define format (X Y Z I J K)

metric formats

decimal places = 4

end define

define keys

spindle not used

feedrate = F

end define

block start = 5

block increment = 5

define block tool change

N ; T ; "M6"

N ; Z FromZ

end define

end

Где первый блок задает коды имеющие начальные нули - G и M.

Координаты имеют 4 числа после десятичной точки.

Третья часть означает, что чистота вращения шпинделя S не задается, а скорость подачи задается с F.

Следующий блок – нумерация блоков начинается с 5, и повышается на 5.

Последняя часть дает вывод смены инструмента в форме:

N10T2M6

N15Z50.

Для детального изучения состава опций см. [Структурные Опции.](#)

Программы ЧПУ

Программы ЧПУ, обычно состоят из ряда блоков, каждый из которых – это команда на станок.

Пример

N30 X2.3 Y4.56

где: номер блока **30**, движение в точку **X2.3 Y4.56**.

Блок состоит из слов, обычно буква сопровождается цифрой, и хотя большинство записей имеет одну букву на каждое слово, имеются слова в которых используется две или более букв (например, I, J и K - это стандартные буквы, определяющие центр окружности, также некоторые станки используют слова CX, CY и CK). Цифры имеют различные форматы, обычно координаты пишутся с десятичными точками, а остальные без них.

Типичная информация содержащаяся в программе ЧПУ:

Block number	N.....e.g. N25	Номер блока
Feedrate	F.... e.g. F300 or F300.	Скорость подачи
Spindle speed	S.....e.g. S850	Частота вращения шпинделя
Tool Number	T.....e.g. T2	Номер инструмента

Станочные функции:

Machine stop	M0	Станок стоп
Spindle on clockwise	M3	Вращение шпинделя по часовой стрелке
Spindle on anticlockw.	M4	Вращение шпинделя против часовой стрелки
Spindle off	M5	Шпиндель выкл.
Tool change	M6	Смена инструмента
Coolant on	M8	Охлаждение вкл.
Coolant off	M9	Охлаждение выкл.
End of tape	M2	Конец записи
End of prog	M30	Конец программы

Подготовительные функции:

Rapid linear move	G0	Ускоренное линейное перемещение
Feedrate linear move	G1	Линейное перемещение с подачей
Circle clockwise	G2	Круговое движение по ЧС
Circle anticlockwise	G3	Круговое движение против ЧС
Compensation on	G41	Компенсация вкл.
Compensation off	G40	Компенсация выкл.
Metric coordinates	G20 или G70	Метрические координаты
Inch coordinates	G21 или G71	Дюймовые координаты
Absolute coordinates	G90	Абсолютные координаты
Incremental coordinates	G91	Относительные координаты

Форматы

Слова с десятичной точкой, обычно имеют максимальное количество знаков после «запятой» (чаще всего: 3-для метрической СК, 4-для дюймовой СК) Слова могут иметь впереди стоящие или позади стоящие нули.

Например, X3.45 X3450 X000345 X3.450 X0003.45

Все записи для значения 3.45

Основные коды **M** и **G** имеют максимум две цифры(например, **G01** или **G1** ; **M03** или **M3**)

Пример записи

%	Начало записи
:0001	Номер программы
N10G28G91X0Y0Z0	Перемещение в стартовую позицию (данная особенность – для Fanuc)
N30T1M6	Смена инструмента
N40G0G90X-25.Y-40.S800 M3	Ускоренное перемещение в абсолютные координаты X-25 Y-40, вкл. вращение шпинделя
N50G43Z3.H1M8	Коррекция на длину инструмента, вкл. охлаждения, перемещение вниз
N60G1Z-5.F100	Перемещение с подачей в рабочую плоскость
N70G41Y40.D1F200	Вкл. Компенсации режущего инструмента
N80X-5.	
N90G2G17X25.Y10.I0J-30.	Круговое перемещение в X25 Y10, центр I0 J-30 от начала дуги
N100G1Y-10.	
N110G2X-5.Y-40.I-30.J0	
N120G1X-25.	
N130G0G40Z20.	Выкл. компенсации инструмента
N140G28G91Z0	Возврат в опорную точку
N150G49H0	Выкл. компенсации длины инструмента
N300G28X0Y0	Возврат в опорную точку XY
N310T3M6	Смена инструмента
N320G0G54G90X100.Y-35.S1500M3	
N330G43Z3.H3	
N340G1Z1.F40	
N350G83G99Z-10.R1.Q3.66F150	Цикл глубокого сверления
N360G28G91Z0	Возврат в опорную точку по Z
N370G49H0	Выкл. компенсации длины инструмента
N380G28X0Y0	Возврат в опорную точку по XY
N390M30	Конец программы

Файл сообщений

Ductpost содержит сообщения которые пишутся и печатаются в файле во время прогона ductpost. Эти сообщения могут быть изменены или транслированы, как требуется Вам.

Это используется, если Вы хотите передать сообщения на другом языке, во время выполнения прогона на программы (например: " **Enter program number** " в " **Введите номер программы** ") или в формате файла **.inf** .

Наиболее часто транслируемые строки файла сообщений:

102 "Enter program number :"	(«Введите номер программы»)
103 "Enter cutfile name :"	(«Введите имя cutfile»)
109 "Incomplete cldata file"	(«Не скомплектован файл cldata»)
111 "Enter part number :"	(«Введите номер детали»)
127 "Problem opening files"	(«Проблема открытия файла»)
172 "Error(getrec): Could not determine Cldata precision"	(«Ошибка (getrec): Невозможно точно определить Cldata»)
186 " x too small in block %(I10)"	(«В блоке слишком малое значение X»)
187 " x too large in block %(I10)"	
188 " y too small in block %(I10)"	
189 " y too large in block %(I10)"	
190 " z too small in block %(I10)"	
191 " z too large in block %(I10)"	
192 " Control not available "	(«Контроль не возможен»)

Для внесения изменений файл: **/dcam/product/ductpostNNNN/sys/hci/english.msg** должен быть скопирован в **/dcam/product/ductpostNNNN/sys/hci/ductpost.msg**.

Тогда файл **ductpost.msg** будет использоваться вместо файла заложенного по умолчанию.

Каждое сообщение выдается номером сопровождаемым **форматированной строкой заключенной в двойные кавычки**. Если в сообщении более одной строки, то следующие строки добавляются используя + «**следующая строка в двойных кавычках**»

Пример:

```
176 "-----"
+ "----- Setup Sheet -----"
+ "-----"
```

Формат каждого CLdata контролирует файл сообщений:

%(I10)	значение 10 целых чисел
%(F5.1)	5 реальных цифр с одной цифрой после «запятой»
%(A10)	10 строк символов
%(A-10)	строки могут быть выравнены по левому краю

Невозможно изменить переменные если написан только их формат.

Пример вывода строк в информационном файле:

```

175 " -----"
+ " ----- Setup Sheet -----"
+ " -----"
+ " "
+ " -----"
+ " | Date : %(A-11) |"
+ " | Time : %(A-11) |"
+ " | Programmer : %(A-11) |"
+ " | Machine : %(A-11) |"
+ " |-----|"
+ " | Block Tool Number Length Radius Comments |"
+ " |-----|"

```

(где Date и Time – дата и время, Programmer - текущий login name, Machine – тип станка из option-файла)

```

176 " | | | | |(A-35)| " (Формулировка комментария DUCT pprint)
177 " | | | | |"
178 " |(I5)|(I2) |(I2) |(I2))| |(F5.1) | |(F5.1) |"
(номер блока, номер инструмента, смещение радиуса, смещение длины, длина, радиус)
179 " |-----|"
180 "

```

```

=====
+ " | | | | |"
+ " | Maximum Minimum |"
+ " | | |"
+ " | X % (F10.3) |(F10.3) |"
+ " | | |"
+ " | Y % (F10.3) |(F10.3) |"
+ " | | |"
+ " | Z % (F10.3) |(F10.3) |"
+ " | | |"
+ " |
=====

```

(Максимальное и минимальное значение X Y и Z из cutfile)

```

193 " | | | | | Tip radius % (F10.3) |" (Радиус скругления зуба)
194 " | | | | | Feedrate % (F10.3) |" (Скорость подачи)
195 " | | | | | Plunge Rate % (F10.3) |" (Скорость погружения)
196 " | | | | | Safe Z % (F10.3) |" (Начальное положение Z)
197 " | | | | | Rapid Height % (F10.3) |" (Значение Z после 1-го прохода по Z)
198 " | | | | | Spindle speed % (F10.3) |" (Частота вращения шпинделя)
199 " | Part number |% (A-35) |" (Номер детали)

```

Раздел 2. Форматы опций

Форматы слов

Строка программы ЧПУ содержит набор слов: **G1 X... Y... Z... F...** и т. д.

Каждое слово может быть отформатировано описанием, заложенным в option-файле. Слова могут быть описаны индивидуально или группами.

Но необходимо внести в список форматов, изменения их первоначального статуса.

Форматы изменения описаны ниже:

*Каждое описание формата должно начинаться с: **define format (...)** и заканчиваться **end define**.*

Формулировка **define format** должна сопровождаться именем(-ами) одного или нескольких слов в скобках, разделенных пробелами.

Примеры:

```
define format ( X Y Z )
field width = 8
leading zeros = false
decimal point = true
end define
```

```
define format ( G M )
leading zeros = true
decimal point = false
end define
```

Что дает вывод координат **X Y Z** в форме: **X3.123 Y78.9 Z400**.

Коды **G** и **M** выводятся в форме: **G01 G90 M03 M10**

Если все слова должны иметь общий формат, то используется следующий синтаксис:

```
define format all
tape position = 0
end define
```

Что приводит к отсутствию пробелов между словами в программе. (Более подробно см. [the document on spacing.](#))

(Чаще всего между словами ставится один пробел, что обеспечивает удобство чтения программы, однако это удлиняет строки и увеличивает размер файла).

Возможные определения форматов

Заданный формат слова	Пример использования	Краткое пояснение
address letter	address letter = "MSG, "	Слово начинается MSG,
address width	address width = 5	4 знака включая пробел, или 5 знаков
field width	field width = 8	До 8 цифр dddd.d включая точку, или 8 цифр. (125 max.)
tape position	tape position = 1	1 пробел перед словом
scale factor, <i>или</i> scale divisor	scale factor = 2 <i>или</i> scale divisor = 4	Умножение значение на 2 или деление на 4

sign	sign = if negative	Вывод только отрицательного знака
modal, или not modal	modal	См. Примечания ниже
permanent, или not permanent	not permanent	См. Примечания ниже
metric formats, и/или imperial formats	metric formats	Вывод метрических или дюймовых форматов
decimal places	decimal places = 3	3 знака после точки
decimal point	decimal point = false	Десятичная точка не нужна
leading zeros	leading zeros = true	См. Примечания ниже
trailing zeros	trailing zeros = true	См. Примечания ниже
exponent width	exponent width = 2	X3.45E+04

Примечания:

address letter	Обычное значение – это одна буква, " X " или " G ", но может быть до 19 знаков (макс.) .
address width	Может быть от 0 до 19 макс.
field width	Может быть от 0 до 125 макс
tape position = 1	Размещение одного пробела перед каждым словом программы: например N10 G01 X2.345 Y4.56 Z0.567 F600 (По умолчанию может быть tape position = 0 без пробелов)
scale factor или scale divisor	Обычно масштабные факторы умножения и деления = 1 (one), но бывает необходимым умножить или разделить значения на какое-либо число, или сменить знак. (Значение может быть только целым числом) (scale factor = -1)
sign	Использование: sign = if negative для вывода в координатах, только знака минус, sign = always если требуются знаки + / - . sign = none для кодов G и F где знаки не требуются.
modal или not modal	Слово является модальным – если команда действует во всех последовательных строках, пока не отменяется подобной. Обычно коды G и координаты X, Y, Z – модальные, а коды I, J, K – не модальные
permanent	Постоянно требуемые в применении при выводе. Самый общий пример – номер блока (N). Слово должно быть введено во всех строках программы. Примечание : Для " heidiso " каждый блок должен заканчиваться символом " * " для EOB.
metric formats или imperial formats	Иногда требуется иметь различные метрические и дюймовые форматы. По умолчанию - metric formats , но это также зависит от CLdata См. <u>Целые, реальные числа и знаки</u>
leading zeros и trailing zeros	Спереди или позади стоящие нули, могут использоваться для определения положения десятичной точки. (например, выводом значения 345.1 может быть 00345100)

<i>exponent width</i>	По умолчанию – 0, потому не имеет ни какого значения. Однако если значение дано, то слово выводится в экспоненциальном виде (см. выше)
------------------------------	---

Общий пример:

(Примечание: отступ – это пробелы, а не табуляция)

```
define format ( X )
  address letter  = "X"
  address width  = 1
  field width    = 8
  tape position  = 1
  scale factor   = 1
  scale divisor  = 1
  sign           = always
  modal
  metric formats
  decimal places = 3
  decimal point  = true
  leading zeros  = false
  trailing zeros = false
  imperial formats
  decimal places = 4
  decimal point  = true
  leading zeros  = false
  trailing zeros = false
end define
```

В этом случае **X** появляется как **X+3.456** в метрической форме, и как **X+3.4562** в дюймовой.

```
define format ( G1 )
  address letter  = "G"
  address width  = 1
  field width    = 2
  sign           = none
  not modal
  metric formats
  decimal places = 0
  decimal point  = false
  leading zeros  = true
  trailing zeros = true
  imperial formats = metric formats
end define
```

В этом случае **G1** появляется как **G01** как в метрических так и в дюймовых СК.

Some Typical Integers, Reals and Characters (Некоторые типичные целые, реальные числа и знаки)

В файле конфигурации используемого постпроцессора может быть множество флажков с заданными значениями контроля генерируемой программы:

flag name = value

Значение должно быть согласовано с типом флажка, иначе выводится сообщение об ошибке, и постпроцессирование прерывается, или может игнорироваться.

Например:

block increment = 10

maximum feedrate = 1000. # обратите внимание на точку

message output = true

данная запись является правильной, а при неправильной записи, например: **block increment = 10.** или **maximum feedrate = 1000** выводится **сообщение об ошибке**

Флажки (метки) разных типов и их функции:

Плавающая точка:

arc radius limit = n.	Ограничение Максимального радиуса дуги при выводе программы ЧПУ и согласуемый с возможностями станка. (До версии DP1335 был real 16)
arc minimum radius = n.	Ограничение Минимального радиуса дуги при выводе программы ЧПУ и согласуемый с возможностями станка. (Новая DP1357)
diameter = n.	Очень редко используется и был введен для системы ЧПУ Lathe. М/с специфический параметр (Linear = Pi * dia * A / 360)
maximum feedrate = n.	Максимально допустимая скорость подачи – это действует на линейную подачу и любые другие Плунжерное/Cutting/Ускоренное. В случае задания подачи более установленной, это значение исправляется до допустимого значения.
minimum feedrate = n.	Минимально допустимая скорость подачи – любая подача заданная менее этого значения исправляется до допустимого значения.
maximum segment = n.	Определяет где в файл программы должна быть сделана вставка в feet (фут). (См. раздел Сегментация)
maximum tape length = n.	Максимальная длина одной части программы ЧПУ. По умолчанию неограниченна (set to 0) – Если размер программы слишком велик, она автоматически разбивается на отдельные файлы.
rapid feedrate = n.	Максимальная ускоренная подача – Это НЕ линейная подача (G0) и используется в вычислении машинного времени. Должен быть согласован с возможностями станка.
plunge threshold angle = n.	Угол наклона – при котором Плунжерное перемещение определяется и рассчитывается как плунжерное. (По умолчанию - 0.0 Вертикаль – Используйте только в исключительных случаях)
withdrawal amount = n.	Расстояние отхода инструмента при работе – при перенастройке осевых вращений.

Примечание: Все вышеупомянутые параметры задаются с ' .' (точкой)

Целочисленные переменные:

block increment	Возрастание номеров блоков (строк) программы ЧПУ
block start	Значение используется для задания номера первого блока (строки)
counter increment	Количество на которое изменяется переменная " counter (счетчик) ".
counter start	Стартовое значение для переменной " counter (счетчик) "
cycle output	Используется для определения типа вывода циклов. (Не очень достоверна – пользуйтесь выбором в <i>PowerMill</i>)
integer 69	Используется для задания скорости подачи при плунжерном фрезеровании (по умолчанию = 0 – около 85% = 2 – фактически заданная подача)
maximum block number *	Максимальный номер блока (строки), после которого производится сброс на «0».
maximum tape blocks *	Максимальное количество блоков (строки) в программе
minimum tape blocks *	Минимальное количество блоков (строки) в программе
program id start	Значение по умолчанию progid . (Обычно по умолчанию = 1)
segment type	Критерий для разбиения и сегментации программы 0 – по длине программы в футах (feet) 1 – по длине перемещения инструмента 2 – по номеру блока 3 – по количеству символов
special zero	Если = 2 , то значение специального ноля выводится как ' zero = "0.0" ' Какое бы значение не было, это действие может быть игнорировано и переопределено к нужному ' zero = "0.0" '
split move	Критерий для разбивки перемещений 0 -- перемещения по XYZ не разбиваются (Для много-осевых) 1 – разбиение перемещений на "XY" и "Z"
tape split retract distance *	Расстояние, на которое отводится инструмент при разбиении программы.
tool reset coordinates	Установка критериев для перемещений после смены инструмента (Не используйте многоосевые перемещения инструмента.) 0 – Без спец. действий. Output depends on CLdata input 1 – Вывод координат X и Y. 2 – Вывод координат X, Y и Z 3 – Вывод X, Y в одной строке, Z в следующей 4 – Вывод Z в одной строке, затем X Y в следующей

* см. Раздел «Разбиение программы»

Logical flags :-

block order	= false Вывод слов и т.п. в порядке их задания списком " <u>word order</u> " (порядок слов) в исходном файле. = true Output Вывод слов и т.п. в порядке их появления в блоке определенном в option-файле.
full circle	= true Определение вывода полной окружности в круговых перемещениях. = false Вывод линейных перемещений для окружностей
go home output	= true Вывод движения «иди домой». = false Подавление движения «иди домой».
incremental centre +	= false Вывод координат центра круга I, J, K в абсолютных значениях. = true Вывод координат I, J, K как относительное значение между начальной точкой и центром.
message output	= true Вывод сообщений cldata в файле программы. = false Подавление сообщений cldata в файле программы.
retract at angular limit ++	= false Мульти-осевой возврат и переконфигурация не осуществляется и постпроцессирование прерывается сообщением об ошибке, если угловые и линейные пределы пути инструмента превышены. = true Возврат и переконфигурация будут позволяться, если возможно вставить дополнительное движение, кроме заданных в PowerMill.
spindle w motion spindle x motion spindle y motion spindle z motion	= false Шпиндель совмещается с осью W = false Шпиндель совмещается с осью X = false Шпиндель совмещается с осью Y = true Шпиндель совмещается с осью Z
spindle azimuth rotation ++	= false Вращение оси в единицах стола = true Вращение оси в единицах инструментального барабана
spindle elevation rotation ++	= false Вращение оси в единицах стола = true Вращение оси в единицах инструментального барабана
tlo output	= false Подавление коррекции на длину инструмента при его смене. = true Вывод коррекции на длину инструмента при его смене.
use partid	= true Используется имя детали из cut-файла. = false Вызывает id детали с клавиатуры. (Не из P'Mill)
use progid	= true По умолчанию используется (1), или заданный номер программы. = false Вызывает id программы с клавиатуры. (Не из P'Mill)
tape split on tool change +++	= false Нормальный вывод программы = true Вывод программы со сменой инструмента разбитой в позиции смены. (Используется для вывода подпрограмм)
	(Значения по умолчанию) : (Альтернативные значения)

"+" – см. Раздел «Вывод окружностей»

"++" – см. Раздел «Мультиосевые файлы (4 оси - 5 осей)

"+++" – см. Раздел «Подпрограммы»

Строки:

machine name	Пример " Fanuc6m version 1.2 " печатается при старте постпроцессирования в перфокартах
print header	Пример " Delcam Postprocessor " печатается при старте постпроцессирования в перфокартах
point	Пример " . " Символ для использования десятичной точки
zero	Пример " 0.0 " Строка символов для использования вывода плавающей точки нуля

Другие:

azimuth axis direction и elevation axis direction	= positive Направление вращения по ЧС рассматривается как положительный конец вокруг оси вращения = negative Изменение направления вращения
azimuth axis units и elevation axis units	= degrees Измерение угла поворота в градусах. = linear units Измерение угла поворота в линейных единицах. (Используется совместно с параметром diameter)
coordinates	= absolute Вывод координат в абсолютных значениях. = incremental Вывод координат в относительных значениях.
spline type +	= bspline Вывод данных сплайна в формате B-spline = polynomial Вывод данных сплайна в формате polynomial .
units	= metric Программа выводится в метрических единицах (даже если в cut-файле заложены дюймы). = imperial Программа выводится в дюймовых единицах (даже если в cut-файле заложены метрические). = input Программа выводится в единицах согласно cut-файла.
	(обычно по умолчанию) : (Альтернатива) : (+ М/с специфичн)

Для более полной информации см. раздел «Список Целых чисел»

Ключи и коды

Ключи:

Ключи зависят от определенных форматов и слов заложенных в постпроцессоре.

Например "**x coordinate**" обычно связан со словом X. Что определяется следующим синтаксисом:

```
define keys
  x coordinate = X
end define
```

Чаще всего ключи редко меняются от их настроек по умолчанию, поэтому "**define keys**" редко используется в option-файле.

Однако, в некоторых случаях может понадобиться изменение функции ключа(-ей), например для переключения осей (ось Y в Z и Z в Y).

Также можно указывать что слово не используется для отдельной функции (например **spindle = not used**)

Важно: Имена групп используется для определения ключа, а не символа. Так **auxfun = M1** *не* **auxfun = M**. Имена групп могут быть найдены в исходном файле.

Некоторые имена ключей и кодов могут быть сокращены, так "**x coordinate**" может быть записан как "**x coord**"

Рекомендуется эти ключи или коды определять используя в их опциях символы (например **key i** вместо **I : tool length offset** вместо **G6 43**)

Пример:

```
define keys
  z coordinate = Y
  y coordinate = X
  x coordinate = Z
  feedrate    not used
end define
```

Здесь изменены обычные слова для координат, и значение скорости подачи не выводится.

Полный список:

aux function	azimuth axis	blocknumber	clearplane
cycle dwell	circle angle	drill hole depth	drill peck depth
dwell	elevation axis	feedrate	feed per rev
key i	key j	key k	leader
message end	message start	opskip	preparatory function
program id	radius	spindle	tool length
tool length offset	tool number	tool radius offset	x coordinate
y coordinate	z coordinate	x feedrate	y feedrate
z feedrate	x vector	y vector	z vector
3rd rotation axis	error		

Коды

Коды predetermined пунктом **whose value** не изменяются и находятся в *секции определения кодов* (подобно указанным выше **секциям определения ключей**).

Они используются в выводе кодов системы ЧПУ в программу, эти коды являются стандартными адресными символами G и M.

Каждый код должен быть связан с выводимым словом и значением (например **G1 0** для вывода кода ускоренного перемещения (**G0**)).

Они определяются следующим образом:

define codes

function name = *формат метки слова* **значение функции кода**
end define

Пример:

define codes

rapid = **G1 0**
linear = **G1 1**
comp on left = **G2 41**
comp on right = **G2 42**
comp off = **G2 40**
spindle on cw = **M1 3**
coolant on = **M2 8**

end define

G1 и **G2** стандартные имена групп для **G-кодов**, **M1** и **M2** для **M-кодов**.

В одной строке может быть несколько **G** или **M-кодов**.

Это справедливо для кодов линейного и ускоренного перемещений **G0**, **G1**, и компенсации режущего инструмента **G41**, **G42**, **G40**.

(**Примечание:** It is not permissible to have **two G codes** of the same group on a line [e.g. **G2 40** ; **G3 17** ; **G2 80**] as **G2 80** will overwrite **G2 40**)

Полный список:

Следующие функции **code names** распознаются постпроцессором и задаются соответствующие **G-коды**, если заданы в исходном тексте или в option-файле:

absolute data	bore 1	bore 2	bore 3
bore 4	bore 5	break chip	change tool
circle ccw	circle cw	clamp off	clamp on
compensation off	compensation on left	compensation on right	constant surface speed
coolant off	coolant on	coolant on flood	coolant on mist
coolant on tap	cycle retract	deep drill	drill
dwell	end of drill	end of prog	end of tape
feedrate per	feedrate per rev	from	gear range 1

minute

gear range 2

linear

spin coolant off

spindle on cw

stop

xy plane

gear range 3

metric data

spin cool on ccw

spindle off

tap

xz plane

imperial data

opt stop

spin cool on cw

spindle rpm

tool length offset

zy plane

incremental data

rapid

spindle on cc

spline

Формирование блоков

Блок – это строка или набор строк появляющихся в программе ЧПУ, для особенных частей cut-файла, например: начало программы, линейные движения, циклы сверления и конец программы (см. список «Имена блоков»)

Каждую часть программы ЧПУ задает **define a block**, состоящий из нескольких слов с нормальными значениями (например, " **X** " –м нормальное значение координаты X, " **F** " –скорость подачи и т.д.). Это может быть задано фиксированным значением (например, **F 9999**) или с использованием переменной (например **X OldX** вывод **X** с **последним значением**).

Define block список возможных слов, для вывода их в одной строке программы ЧПУ, слова разделяются «;».

Если между словами находящимися в одной строке стоит «точка-с-запятой», то в программу ЧПУ они будут выводиться в одной строке. Строки или части строк могут быть также определены используя запись в двойных кавычках (например, "G90G70")

Все блоки имеют следующую форму:

define block

**.....
end define**

Пример:

define block tape start

"% "

N ; ID PartID ; ")"

end define

define block tool change

N ; T ; M1 6

N ; S ToolSpeed ; M1 3

N ; G5 90 ; G6 54 ;

end define

define block move rapid

N ; G1 ; X ; Y ; Z ; F 5000

end define

define block move linear

N ; G1 ; X ; Y ; Z ; F

end define

Что даст вывод программы ЧПУ в форме:

%

N1 (PART NAME- Example Output)

N2 T1 M6

N3 S2500 M3

N3 G0 G90 G54 X... Y... Z... F5000

N4 G1 X.... Y.... Z..... F250

Блоки могут быть заданы в любом порядке, но рекомендуется сохранять логическую цепь, т.к. это упрощает дальнейшую отладку и корректировку.

Слова:

Большинство используемых слов предопределены исходным файлом и выводятся определенными символом и форматом.

Для их просмотра необходимо в командной строке Window, или в текстовом файле напечатать **ductpost - w [control name]** в командной строке, или **ductpost - w [control name] > [control name] . dmp** в текстовом файле. (используется любой текстовый редактор)
(Для более полной информации см. «Список Слов» и «Список Форматов»)

Слова типа " **X Y Z I J K F** " и т.д. будут иметь индивидуальную функцию вывода (например, **X1.234 Y5.678 Z9.0 F250**), тогда как **G** и **M** –группы слов имеющие один и тот же символ (например, **G1 G2 G3** все имеют "G", **M1 M2** - "M", за исключением Heidenhain)

Вышеупомянутый пример поясняет, что **G5 90 ; G6 54** выводятся **G90 G54** и **G1** может выводиться **G1 G0 G2 G3** в зависимости от характера выполняемого перемещения. Подобно, **M1** может выводиться **M0 M1 M3 M4 M54** и т.д. в зависимости от выводимой функции и списка **define codes**.

Значения:

Если необходимо выводить фиксированные значения, то в части блока должно быть :-

F 9999 будет выводиться как **F9999**

Примечание: В значениях нельзя использовать десятичные точки, так если требуется получить **X1.234**, то ввод для метрической системы - **X 1234**, и для дюймовой **X 12340**

Если значение требуется выводить в следующих строках, даже если значение остается неизменным, то слово необходимо писать как: **F =C**

Переменные:

Слова также могут быть выведены как переменные. Это может понадобиться, если необходимо вывести слово не с его обычным значением, например в блоке смены инструмента, обычная форма может выглядеть как:

define block tool change

T ; M1 6

end define

что

дает вывод: **T2 M6** для второго инструмента.

Однако некоторые станки могут требовать, предварительного выбора инструмента, тогда:

define block tool change

T ; M1 6

T NextTool

end define

что

дает вывод **T2 M6** для второго загруженного инструмента
следующего загружаемого инструмента.

и **T3** для

(Это часто используется для загрузки следующего инструмента в автоматическое устройство смены, во время того как обработка продолжается текущим инструментом).

Некоторые примеры:

Эти блоки используются для заголовка второго и последующего файла в разбитой на отдельные файлы программе ЧПУ:

```
define block tape split start  
  ID ProgID  
  N ; " G90 G70"  
end define
```

```
define block tape split move  
  N ; G1 0 ; X OldX ; Y OldY  
  N ; G1 =C ; Z StartZ  
  N ; G1 1 ; Z OldZ  
end define
```

ProgID для задания оригинального номера программы (например, MPF0001)
OldX, OldY и **OldZ** предыдущие позиции **X, Y** и **Z**.
StartZ проверка позиции плоскости.

(Другие переменные см. раздел «Переменные Блоки»)

Загрузка управляющей конфигурации DUCTpost

Иногда необходимо исследовать полную конфигурацию управления, используемую DUCTpost. Загрузка производится с помощью переключателя " **-w** ". (Загрузка выводится в окно MS-DOS, но может быть и переадресована в файл, который сохраняется в рабочей директории для дальнейшего использования).

Например, для получения загрузки внутренней конфигурации системы управления " **heid400** ", наберите в командной строке MS Command Window:

```
ductpost^-w^heid400^>^heid400.dmp (^ = пробел)
```

Это поместит файл в Ваш рабочий каталог.

Загрузка может производиться из *option-файла*

```
ductpost^-w^DMU60PT-EMetV2.opt^>^DMU60PT-EMetV2.dmp ( ^ = пробел )
```

(будет загружена полная конфигурация включая измененные опции), только для любой встроенной конфигурации. (Это необходимо для просмотра списка слов или сравнения изменой конфигурации с исходной)

Для получения списка систем управления поддерживаемый DUCTpost, используйте переключатель " **-l** " (строчная L), т.е. напечатайте:

```
ductpost -l ( строчная L )
```

Раздел 3. Функции постпроцессора

Номера блоков

Номера блоков используются для определения номера строки программы, они могут начинаться с 1 и прирастать по 1, как показано ниже:

```
% O1234
N1 G99 M6 T1
N2 G0 X0 Y0 S1850 M3 PA10
N3 Z10.
N4 G30
N5 G0 X10.118 Y-36.377
```

Однако, это не всегда требуется, различные варианты нумерации приведены ниже:

Пример 1

Если номера блоков не требуются, то номера переопределяются ключом " **N** ".

```
define format ( N )
  not permanent
end define
```

удалит номера блоков из программы:

```
% O1234
G99 M6 T1
G0 X0 Y0 S1850 M3 PA10
Z10.
G30
G0 X10.118 Y-36.377
```

Это

Если встроенный постпроцессор не выводит номера строк, то используйте формат " **N** " в " **permanent.** "

Пример 2

Если необходимо изменить последовательность нумерации, то вставьте в option-файл следующее:

```
block start      = 10
block increment  = 5
```

дает первый блок 10, с последующим приростом на 5.

Это

Пример 3

Если максимальное количество строк ограничено системой ЧПУ, то вставьте в option-файл:

```
maximum block number = 5999
```

тогда после номера 5999 произойдет сброс на начальный номер.

Пример 4

Если требуется задать специальный номер, как указано ниже,

```
N1000 BEGIN PGM SPECIAL MM
N1000 TOOL CALL 0 Z S3000
N1000 M55
N1000 M3
N1000 CYCL DEF 19.0 BEARBEITUNGSEBENE
N1000 CYCL DEF 19.1 A0 B0 C0
N10 L X0.0 Y0.0 Z150.0 B0 C0 FMAX
N12 L X254.345 Y146.780 B90.0 C35.250 FMAX
N14 L Z-55.70 FMAX
N16 L .....
```

Тогда выполните следующее:

define word NF

address letter = "N1000"

address width = 5

field width = 0

permanent

end define

word order = (+ NF)

define block tape start

NF ; " BEGIN PGM" ; ID PartID ; metric data

end define

define block tool change first

NF ; T2 0 ; " Z " ; S 3000

NF ; M1 55

NF ; M1 3

NF ; G4 190 ; " BEARBEITUNGSEBENE "

NF ; G4 191 ; A 0 ; B 0 ; C 0

N ; G1 ; X FromX ; Y FromY ; Z FromZ ; B =C ; C =C ; FMAX

end define

См. Также «Форматы» и «Целочисленные флажки»

Линейные перемещения

Обычно минимальный формат ввода для линейных перемещений, следующий»

G01 X... Y... Z... F...

где **G01** (или **G1**) - код линейных перемещений.

X Y и **Z** – абсолютные или относительные позиции.

F – скорость подачи

Скорость вращения шпинделя и коррекция длины инструмента может также задаваться для линейных перемещений, это наиболее вероятно для мульти-осевых перемещений.

Также может быть задана коррекция режущего инструмента, это применимо 2 и 3D обработке PowerMill.

define block move linear - очень редко определяется в исходных файлах.

Однако это рекомендуется сделать, если блок не задан в исходном файле управления постпроцессора, например, если его было бы лучше определить опциями. (В редких случаях обнаруживаются ошибки)

В option-файле "**define block move linear**" может выглядеть так:

```
define block move linear  
N ; G1 ; G2 ; X ; Y ; Z ; D ; F ; M1 ; M2  
end define
```

(**G1** –код линейных перемещений (обычно выводится **G1**), **G2** – код компенсации радиуса инструмента (обычно **G41** или **G42**), **D** – отступ радиуса инструмента, **F** – скорость подачи, и **M1, M2** - код M-функций. [Для мульти-осевых перемещений правильным будет добавить G6, S и H]

- Мы РЕКОМЕНДУЕМ использовать вместо вышеупомянутого следующий формат:

```
define block move linear  
N ; linear ; G2 ; x coord ; y coord ; z coord ; tool radius ; feedrate ; M1 ; M2  
end define
```

- Компенсация радиуса должна быть добавлена к большинству опций, т.к. это пропускается в исходных файлах и требуется для 2D-обработке в PowerMill 2D. (Многие системы управления ЧПУ не поддерживают радиальную компенсацию в дугах, а только на прямых участках)
Скорость подачи обычно выводится для линейных перемещений.

Первое перемещение после смены инструмента всегда рассматривается как особый случай (обычно это функция ускоренного перемещения, но может overlap (наложиться, внахлест, перекрытие), особенно в мульти-осевых опциях. В этом случае блок линейных перемещений нуждается в особом внимании (см. раздел «Смена Инструментов»).

Может быть установлен предел линейных перемещений, при этом появляется предупреждение о превышении этих пределов по X, Y или Z. (По умолчанию, эти пределы в DUCTpost заданы в пределе от -999999. до 999999. для всех осей. См. Раздел «Пределы»)

Выполнение проверки угловых изменений между перемещениями, возможно в выводе кода **постоянной контурной скорости** (например, **M90** в Heidenhain) для перемещений где угловые изменения очень малы.

- **Примечание:** Все мульти-осевые перемещения рассматриваются как линейные, таким образом необходимо добавить угловые оси: **azimuth axis ; elevation axis** к вышеупомянутым.

Ускоренные перемещения

Типичный формат вывода ускоренных перемещений, может выглядеть так:

G00 G6 X... Y... Z... S... H.... M3

Где, G00 (или G0) – код линейных перемещений.

X Y и Z – абсолютные или относительные позиции.

Может быть задана скорость вращения шпинделя и компенсация длины инструмента для ускоренных перемещений, S2500 M3 и G43 H1..

Код компенсации радиуса инструмента наиболее вероятна при выводе линейных перемещений.

Блок ускоренных перемещений обычно определен в исходном файле.

" **define block move rapid** " может выглядеть так:

define block move rapid

N ; G1 ; G2 ; G3 ; G6 ; X ; Y ; Z ; H ; S ; M1 ; M2

end define

(**G1** – код ускоренных перемещений (обычно **G0**)),

(**G2** – код компенсации радиуса инструмента (обычно **G41** или **G42**))

(**G3** – код рабочей плоскости (**G17**, **G18** или **G19**))

(**G6** – код компенсации длины инструмента (возможно **G43**))

(**S** - скорость вращения шпинделя)

(**F** - ускоренная скорость подачи (обычно не требуется, но можно затем задать опцию "**rapid feed code = 1** "))

(**H** – компенсация длины инструмента. (Это требует задания "**tlc output = true** " и "**tool reset coordinates = 3**)

(**M1**, **M2** - может быть M-функцией для ВКЛ шпинделя (**M3**) и/или ВКЛ охлаждения (**M8**))

Мы РЕКОМЕНДУЕМ следующий формат, вместо вышеупомянутого:

define block move rapid

N ; rapid ; G3 ; tool length offset ; x coord ; y coord ; z coord ; tool length ; spindle ;

M1 ; M2

end define

Одновременное перемещение по всем трем координатам X Y Z может быть разбито на двойные перемещения, поскольку многие станки не поддерживают трехмерную интерполяцию при ускоренных перемещениях. Что задается так:

split move = 1

В этом случае перемещения будут разбиты так: при движении вниз – сперва выполняются по XY затем по Z, движение вверх – сперва по Z затем по XY.

Это не установлено по умолчанию, т.к. обычно ускоренные 3D-перемещения выполняются в безопасных зонах.

Для Мульти-осевых постпроцессорах **НЕОБХОДИМО** задать как " **0** "

Первое перемещение после смены инструмента может быть задано как особый случай, для чего используется “ ***define block move from*** “, тогда перемещения производятся либо с ускоренной либо с линейной скоростью подачи.

Может быть установлен предел линейных перемещений, при этом появляется предупреждение о превышении этих пределов по X, Y или Z. (По умолчанию, эти пределы в DUCTpost заданы в пределе от -999999. до 999999. для всех осей. См. Раздел «Пределы»)

Круговые перемещения

Обычная форма для круговых перемещений:

```
G02 X... Y... I... J... F...)  
или } xy Plane ( G17 )  
G03 X... Y... I... J... F...)
```

```
G02 X... Z... I... K... F...)  
или } zx Plane ( G18 )  
G03 X... Z... I... K... F...)
```

```
G02 Y... Z... J... K... F...)  
или } yz Plane ( G19 )  
G03 Y... Z... J... K... F...)
```

где G02/G03 (или G2,G3) – коды для выполнения кругов по ЧС и против ЧС;

X, Y или Z – конечная точка дуги

I, J или K – определяют центр круга.

G-коды для окружности и плоскостей задаются блоком кодов:

```
define codes  
  circle cw  = G1 2  
  circle ccw = G1 3  
  xy plane   = G3 17  
  xz plane   = G3 18  
  zy plane   = G3 19  
end define
```

где, G1 – первая модальная группа G-кодов, и G3 – третья группа. Обычно они не меняются.

Координаты I, J и K могут определять фактический абсолютный центр круга, в этом случае:

```
incremental centre = false
```

или определять относительную дистанцию между начальной точкой и центром, в этом случае:

```
incremental centre = true
```

Если координаты I, J или K имеют неправильный знак (это важно проверить если тип центра был изменен), его можно изменить используя:

```
define format ( I J K )  
  scale factor = -1 или 1  
end define
```

Позиция центра круга I J, J K или I K появляются с соответствующей выводимой плоскостью. Если код плоскости также требуется (например, **G17**, **G18** и **G19**), тогда G3 определяется - **define block move circle**.

Если код плоскости необходимо вывести на строке перед кругом, то поместите **G3** в отдельной строке - "**define block move circle**".

Для некоторых станков необходимо разбивать окружности на отдельные дуги по квадрантам. Это достигается следующим:

single quadrant = true

Опция файла "**define block move circle**" должна выглядеть так:

define block move circle

N ; G1 ; G3 ; x coord ; y coord ; z coord ; key i ; key j ; key k ; feedrate ; M1 ; M2
end define

G1 – код круга (обычно G2/G3), **G3** – код плоскости (G17, G18, G19).

Вывод дуги круга предотвращен:

Дуги окружности могут быть подавлены и введены в виде коротких прямых отрезков, если определен следующий флажок:

integer 26 = 0 - предпочтительное использование или **circle output = (0 1 1 1)**
(см. «Состав данных»)

Предотвращение вывода дуг кругов в индивидуальных главных плоскостях:

Теперь дуга круга может быть подавлена и введена короткими прямыми отрезками в индивидуальной плоскости, используя определение следующими флажками:

suppress xy arc = true default = **false**
suppress zx arc = true default = **false**
suppress yz arc = true default = **false**

Вывод дуги круга с максимальным станочным пределом:

Если сгенерированный радиус дуги больше максимально доступного предела станка, выводится станочная ошибка.

Значение "**arc radius limit**" – это задание максимального радиуса, и по умолчанию является **10000.0 mm**.

Если нужно ввести другой макс. радиус, то в option-файл вводится следующее:

arc radius limit = 5450.0 (пример)

Вывод дуги круга с минимальным станочным пределом:

Если сгенерированный радиус дуги меньше минимально доступного предела станка, выводится станочная ошибка.

Значение "**arc minimum radius**" – это задание минимального радиуса, и по умолчанию является **0.0 mm**.

Если нужно ввести другой мин. радиус, то в option-файл вводится следующее:

arc minimum radius = 0.016 (пример)

The circle is not in a major plane :

DUCTpost prior to DP1335 will not support circular G2/G3 arc leads output for angular toolpaths offset in the **XY**, **YZ** or **ZX planes greater than 0.1 degrees**. Arcs are output as straight line moves, anything **less** than this, and the circle was **snapped to the plane** and a G02 / G03 output.

From **DP1335** this problem has been rectified to produce sensible output - however G02 / G03 is still not possible.

Реверс G2/G3 для дуг в G18/G19 плоскостях:

В некоторых случаях требуется произвести реверс **G2/G3** в одной или обеих плоскостях **G18** и/или **G19**.

Нет ни какого четкого объяснения, почему это требуется, разве что определено конфигурацией станка.

Крайне редко меняется плоскость **XY (G17)**

Тогда необходимо добавить следующую опцию **define block move circle**:

```
define block move circle
  if ( Word{G3} = 18 )
    N ; G1 ( 5 - Word{G1} ) ; G3 ; x coord ; y coord ; z coord ;
      key i ; key j ; key k ; feedrate ; M1 ; M2
  еще
    N ; G1 ; G3 ; x coord ; y coord ; z coord ;
      key i ; key j ; key k ; feedrate ; M1 ; M2
  end if
end define
```

(Word{G3} = 18 or Word{G3} = 19) если задействованы обе плоскости.

Вывод сообщений

Сообщения (команды APT PPRINT из DUCT и CLdata 1044 PPrint из PowerMill) могут быть выведены в программе используя логическую опцию " **message output = true** " в option-файле. (Обычно настроено по умолчанию в исходном файле, но не во всех случаях)

Как вариант, если сообщения не нужны: **message output = false**

Однако может появиться необходимость переопределить вводимые слова для **start** и **end** сообщений. Пример этого показан ниже:

Вывод требуемого формата: N100 (**MSG**, **ET**) станка

Вывод требуемого начала сообщения, в этом случае (**MSG**,

```
define format ( MS )
  address letter = "(MSG, " ## start address
  address width = 7          ## включая пробелы
  field width   = 0          ## не выводится (это автоматическое управление)
end define
```

Вывод требуемого формата для конца сообщения - **ET**)

```
define format ( EM )
  address letter = "ET )"
  address width = 4
  field width   = 0
end define
```

Нет необходимости снабжать Новыми словами, поскольку MS и EM – обычно определены в исходном файле, это может быть использовано только для переформирования существующего слова. Тоже относится и к вложенному списку порядка слов и ключам, определение показано ниже:

```
#
word order = ( OP N G1 G2 G3 G4 G5 )
word order = ( + G6 G7 X Y Z B C )
word order = ( + I J K R D S T )
word order = ( + H M1 M2 MS msg EM Q )
word order = ( + Q1 Z2 R2 ID F )
#
define keys
  message start = MS
  message end   = EM
end define
#
# Если сообщение не выводится, тогда в option-файл включается следующее:
#
  message output = true
#
end
```

Это произведет указанный ниже вывод:

```
%  
:0001  
N10G91G28X0Y0Z0  
N20G40G17G80G49  
N30G0G90Z10.  
N40T10M6  
N50G54G90M3  
N60( MSG, Toolpath Name: ET)  
N70( MSG, xyzxyz_cut_1 ET)  
N80( MSG, Output: ET)  
N90( MSG, UNITS: MILLIMETRES ET)  
N100( MSG, TOOL COORDINATES: TIP ET)  
N110( MSG, LOAD TOOL ET)
```

Примечание: Всегда в первую очередь проверяйте, действительно ли эти слова определены в источнике и внесены в список порядка слов.

Не обновляйте их, если только не требуется их **переформатирование**, как указано выше.

Пределы

Линейные пределы по осям могут быть установлены строкой:

linear axis limits = (-99999. 99999. -99999. 99999. -99999. 99999.)

где, первая пара – Максимальный/Минимальный предел по оси **X**, вторая пара – по оси **Y**, третья пара – по оси **Z**.

(**Примечание:** " Точка" после значения может быть опущена, однако рекомендуется ее ставить как указано по умолчанию)

Практический пример:

linear axis limits = (-1235.0 1235.0 -540.0 540.0 -250.0 250.0)

Если эти пределы будут превышены, то при постпроцессировании будет напечатано предупреждающее сообщение: например, "**x too small (or large) in block nn**", но вывод фактических значений продолжится.

Круговые пределы по осям могут быть установлены строкой:

rotary axis limits = (-99999 999999 -99999 999999 999 1) ## Настройки по умолчанию

где, первая пара – Максимальное/Минимальное пределы для **Азимутного** вращения, вторая пара – для **Вертикального** вращения, последняя пара – для значения **углового допуска** и требуемое **количество перемещений** для выполнения угловых изменений.

Практический пример:

4-осевой m/c

Азимутный Вертикальный

rotary axis limits = (0.0 0.0 -360.0 360.0 0.1 4)

Если эти пределы будут превышены, то инструмент будет возвращен к предыдущей высоте, и угол вращения отводится назад на 360 градусов, это будет повторяться несколько раз пока нужное значение не будет достигнуто.

(например, если вращательный путь инструмента задан как 1050 градусов, тогда последовательность будет такой: 0 - 360 [360] ; 0 - 360 [720] ; 0 - 330 [1050])

Вероятно, что для таких случаев, система ЧПУ, может иметь специальный код, который нужно вывести, для гарантии одностороннего вращения без возврата в нулевую позицию.

5-осевой m/c

Азимутный Вертикальный

rotary axis limits = (-20.0 110.0 -3600.0 3600.0 0.1 4)

Случай указанный выше, возможен при некоторых обстоятельствах, если переменная комбинация Азимутные/Вертикальные углы не возможны, появится предупреждающее сообщение и постпроцессирование прервется.

Рекомендуется, чтобы **Линейные** пределы по осям были точно определены, поскольку они имеют влияние на вращательные пределы.

Пояснение для Word[x] и возможное их использование

Во вложенных в Ductpost исходных файлах систем ЧПУ, имеется начальный список заданных слов, например:

Станок «tiger»

```
define word /  
  address letter = "/"  
end define
```

```
define word N  
  address letter = "N"  
end define
```

```
define word G1  
  address letter = "G"  
end define
```

```
define word G2  
  address letter = "G"  
end define
```

и т.д.

Каждое из перечисленных выше слов имеет значение, т.е.: / = **word[1]**; N = **word[2]**; G1 = **word[3]**; и т.д. вниз по списку.

Иногда они могут использоваться для достижения определенной функции не заложенной в Ductpost.

Пример для пояснения:

Требуется особенный вывод - в '**xz plane**', необходимо выполнить реверс между **G2** и **G3** только для этой плоскости. Для данного выполнения – нет никакой определенной переменной для '**xz plane**'. Как это можно сделать?

Мы знаем что плоскости задаются кодами **G3 17** (*xy плоскость*), **G3 18** (*xz плоскость*), и **G3 19** (*yz плоскость*). К счастью, **G3** – групповой функциональный код, его значение может использоваться для плоскости используя уловку **word[x]** как показано ниже:

```
define block move circle  
  if ( word[5] = 18 )  
    N ; G1 ( 5 - word[3] ) ; G3 ; G4 ; x coordinate ; y coordinate ;  
    z coordinate ; B ; C ; R =C ; feedrate  
  else  
    N ; G1 ; G3 ; G4 ; x coordinate ; y coordinate ;  
    z coordinate ; B ; C ; R =C ; feedrate  
  end if  
end define
```

В этом примере показана небольшая хитрость для реверса **G2 / G3**

G1 2 (круг по ЧС), **G1 3** (круг против ЧС), 2 и 3 – значения для **G1** (**word[3]**), так если мы используем выражение **G1 (5 - word[3])** мы эффективно можем реверсировать вывод значения **G1**. (т.е. $5 - 2 = 3$ или $5 - 3 = 2$).

Это может понадобиться для решения некоторых проблем, но имеет ограничения. Вам придется экспериментировать и проверять – работает ли это в Вашем случае.

Примечание: Если вы задаете дополнительные слова в option-файле, и желаете использовать их как в вышеупомянутом методе, тогда их значения **добавляются** в конце встроенного списка слов в порядке их определения.

Для проверки порядка сделайте следующее:

напечатайте: **ductpost -w [option file name] > [option file name].dmp** для получения списка в котором определено новое значение **word[x]**.

Список значений слов: см. раздел «Список Word[x]»

Вывод десятичных

Формат вывода координат определяется следующей целочисленной установкой:

В option-файл добавьте строку: ***integer 51 = n***

 **integer 51** = (n) *формат вывода десятичных*

- = 1 число меньше 1 пишется так: **.xxxx** (например: **.871**)
 целое число пишется так: **xxxx.** (например: **34.**)
- = 2 число меньше 1 пишется так: **0.xxxx** (например: **0.871**)
 целое число пишется так: **xxxx.** (например: **34.**)
- = 3 число меньше 1 пишется так: **.xxxx** (например: **.871**)
 целое число пишется так: **xxxx.0** (например: **34.0**)
- = 4 число меньше 1 пишется так: **0.xxxx** (например: **0.871**)
 целое число пишется так: **xxxx.0** (например: **34.0**)
- = 5 также как 3
- = 6 число меньше 1 пишется так: **.xxxx** (например: **.871**)
 целое число пишется так: **xxxx** (например: **34**)

Примечание: Это значение воздействуют на формат вывода любых десятичных.

Пояснение Ускоренной Skim-Подачи (Снятие тонкого слоя)

В PowerMill, Skim-перемещения выводятся как ускоренные **Линейные перемещения** со скоростью подачи установленной в Поле Rapid, а в Ductpos - **Feed Rates Form**. По умолчанию значение - **3000 units/min**

PowerMill выводит скорости подач в Форме Скоростей Подач - "**ppfun fedrat statement**".

Пример вывода вырезки Cldata-файла приведен ниже:

```
-----  
3 ( 25)   2000   1079 PPFUN  
          99  
          FEDRAT 6000 950 1100  
-----
```

т.е. **Ускоренная Skim = 6000 Плунжерная = 950 Резание = 1100**

Если Skim-перемещения будут использоваться при выводе пути инструмента, то время обработки будет зависеть от указанной выше скорости резания, и вывод скорости подачи появится в файле программы как показано ниже:

```
1538 G0 Z84.                ## Обычное ускоренное перемещение (Без скорости подачи)  
1539 G1 X15.653 Y-37.059 F6000 ## Skim-перемещения  
1540 G0 Z69.667             ## Обычное ускоренное перемещение  
1541 G1 Z66.667 F950        ## Плунжерные перемещения  
1542 X14.684 Y-37.456 F1100 ## Перемещения Резания
```

Обычно **NON Linear rapid G0** скорость подачи не действует с ускоренными значениями установленными по умолчанию в "**rapid feedrate = 9999.0**". (Или какое-либо другое значение, заданное в option-файле).

Максимальная **Линейная** скорость подачи по умолчанию контролируется - "**maximum feedrate = 9999.0**" (Или какое-либо другое значение, заданное в option-файле).

Если эта подача задана слишком малым значением, то необходимо включить в option-файл необходимые максимальные значения. (И наоборот, для дюймовой скорости подачи они могут быть слишком большими)

Для вывода скорости подачи в блоке с G0 необходимо добавить/изменить опцию "**define block move rapid**" и добавить "**; F**", и также добавить опцию "**rapid feed code = 1**"

Начиная с версии **DUCTpost1300** добавлена переменная для скорости подачи, "**Srat**", что добавило новых возможностей для манипулирования этой функцией.

Например, skim-перемещения должны быть классифицированы как **Rapid Non Linear G0-перемещения** вместо перемещений **G1**.

Ниже приведен пример этого случая:

```
define block move linear  
if ( feedrate => srat )  
  N ; rapid ; x coord ; y coord ; z coord ; M1 ; M2 ## Rapid Skim как RAPID  
else  
  N ; linear ; x coord ; y coord ; z coord ; feedrate ; M1 ; M2  
end if  
end define
```

Таким образом мы получим следующий вывод программы:

1538 G0 Z84.	## Обычные ускоренные перемещения
1539 X15.653 Y-37.059	## Skim-перемещения (Rapid Non Linear подача)
1540 Z69.667	## Обычные ускоренные перемещения
1541 G1 Z66.667 F950	## Плунжерное перемещение
1542 X14.684 Y-37.456 F1100	## Перемещение резания

Внимание: Рекомендуется осторожно использовать этот метод, т.к. существует опасность получения резких скачков с подачей G0 в перемещениях зигзагом.

Примечание: Если система ЧПУ поддерживает **feed rate parameters**, то для получения корректного вывода скорости подачи для Skim-перемещений см. раздел «Параметры Скорости подачи».

Пояснение по Угловой скорости подачи

Возможно, что некоторые машины требуют назначения вращательным перемещениям индивидуальной скорости подачи, основанной на заданной номинальной скорости подачи. (Обычно задается как *Inverse Time Feedrate*)

Пример:

```
%  
O0100 (0100-Feedrate)  
N2G0G40G17G80G90G49  
N3T04M6  
N4G0G90X0.Y0.A0.B0.S9500M3  
N5G0X0.Y0.A0.B0.  
N6G43Z15.H04  
N7G1Z12.F1.33  
N8X-7.F0.56  
N9Z5.F0.56  
N10Z.9519F0.24  
N11Z.9275A-1.0169F133.88  
N12Z.9033A-2.0339F137.47  
N13Z.8795A-3.0508F138.57  
N14Z.8567A-4.0678F143.12  
N15Z.835A-5.0847F130.66  
N16Z.8145A-6.1017F156.34
```

Для получения этого, в option-файл необходимо включить следующее:

```
integer 71 = 1  
integer 72 = 1
```

Вывод изменения скорости подачи "F" обеспечивается следующим:

```
define format ( F )      [Это пример и выше- для дюймовых опций]  
  not modal  
  imperial formats  
  decimal places = 2  
end define
```

Примечание: Поскольку эта функция очень редко используется, для полной уверенности в эффективности вывода, необходимо делать тщательную проверку, дабы избежать нежелательных результатов в механической обработке.

Разбиение и Сигментация программы

Программы ЧПУ могут быть **Разбиты** на отдельные файлы, по причине ограниченности объема станочной памяти, или начиная с версии Ductpost DP1321 можно использовать вывод подпрограмм.

Сегментация программы ЧПУ – это **специальная вставка** в программе на предварительно определенных интервалах в пределах одного файла.

Разбиение программы

С помощью задания реального флажка "**maximum tape length**", который разбивает программу на небольшие участки, размер которых определяется заданными значениями (см. ниже)

Реальный "**maximum tape length**" - длина задается в **фитах** и должна присутствовать **десятичная точка**.

Длина **100. (фит)** = около 12000 битов.

Длина **0. (ноль)** = без разбиения программы. (Обычно используется по умолчанию для большинства систем ЧПУ)

или,

Целочисленный флажок "**maximum tape blocks = 60000**", выполняющий ту же функцию, но после определенного количества блоков (60000, или любое другое заданное количество).

Рекомендуется использовать только один из двух указанных методов, **а не оба вместе**.

Разбиение программы на подпрограммы см. Раздел «Подпрограммы»

Перемещение инструмента

Следующий важный метод разбиения программы или сегментации – это дистанция пройденная инструментом.

Это задается целочисленным флажком "**tape split retract distance**". (Который заменяет более ранний "**retract distance**" введенный с версии DP1203 и если встречается в option-файле – постпроцессирование прерывается с **сообщением об ошибке**).

Путь отвода инструмента от детали не контролируется опцией "**split move**".

Следующие значения указывают на ожидаемые действия с **NO** " *define block tape split move* " или " *define block move linear* ":

DUCTPost 1100 и ранее	Действие подъема до разбиения программы	Действие возврата после разбиения программы
<i>retract distance = 0</i>	Отвод инструмента в позицию FromZ с подачей резания	Возврат в Safe Z (безопаснаяZ) с ускоренной подачей, затем в позицию Lift Off с подачей резания
<i>retract distance = -999</i>	Без отвода	Возврат в StartZ на ускоренной подаче, затем в позицию Lift Off с подачей резания
<i>retract distance = -998</i> (или любое отрицательное значение, кроме -999)	*** АВАРИЯ PLUNGE ***	Как выше
<i>retract distance = 100</i>	Отвод инструмента на 100 mm выше позиции Lift Off с подачей резания	Как выше
DUCTPost 1205	Действие подъема до разбиения программы	Действие возврата после разбиения программы
<i>tape split retract distance = 0</i>	Без отвода	Возврат в Safe Z (безопаснаяZ) с ускоренной подачей, перемещение в X0,Y0,Z0 с подачей резания (***АВАРИЯ***)
<i>tape split retract distance = -999</i> (или любое другое отрицательное значение)	Отвод инструмента в позицию FromZ с ускоренной подачей	Возврат в Safe Z (безопаснаяZ) с ускоренной подачей, затем в позицию Lift Off с подачей резания
<i>tape split retract distance = 100</i>	Отвод инструмента на 100 mm выше позиции Lift Off с ускоренной подачей	Как выше

Определение следующих блоков используется для разбиения программы:

define block tape split start ; define block tape split move ; define block tape split end

Пример настройки Системы ЧПУ

Ductpost1100 или ранее

maximum tape length = 70.

или

maximum tape blocks = 500

minimum tape blocks = 0 (По умолчанию, обычно не требуется)

retract distance = 100

Ductpost1203 onwards

tape split retract distance = 100

Пример определений блоков:

```
define block tape split start
  N ; "% ( Tape Split Start )"
  N ; ID ProgID
end define
```

```
define block tape split move
  N ; G1 0 =C ; X OldX ; Y OldY ; Z SafZ ; S ; M1
  N ; G1 1 ; Z OldZ ; F Prat
end define
```

```
define block tape split end
  N ; M1 30 ; " ( Tape Split end )"
end define
```

Также необходимо проверить задана или нет в исходном файле "*define block move linear*" и если не задана, то необходимо включить следующее минимальное определение:

```
define block move linear
  N ; G1 ; X ; Y ; Z ; F ; M1 ; M2
end define
```

Сегментация программы

Обеспечивается специальным блоком, который вставляется в программу ЧПУ в определенных местах и контролируется флажком "*segment type*", вместе с: "*maximum segment*", и "*max tape blocks*".

Данное определение вызывает вставку:

```
segment type = 0  после заданной длины программы ЧПУ в фитах
segment type = 1  после заданной дистанции пути инструмента
segment type = 2  после заданного количества блоков
```

Данные вставки управляются используя "*define block tape segment*", и направление движение при сегментации контролируется настройкой "*tape split retract distance*". (" *retract distance* " до DP1203)

1/Пример вставки по количеству блоков

```
segment type           = 2
max block number      = 11500
tape split retract distance = 100    ## Lift Off 100 mm.
```

```
define block tape segment
  N ; " ( Tape segment comment )"
  N ; M1 05 ; M2 09
  N ; M1 01
  N ; G1 00 =C ; X OldX ; X =C ; Y OldY ; Y =C ; Z SafeZ ; M1 03 ; M2 50
  N ; G1 01 ; Z OldZ ; F Prat
end define
```

2/ Пример вставки по длине программы или расстоянию перемещения инструмента

```

segment type          = 0   or   = 1
max segment           = 1100.  ## Вставка в 1100 feet.
tape split retract distance = 100  ## Lift Off 100 mm
split move            = 0      ## Без задания перемещений XYZ

```

Resultant moves to be expected following segmentation insert

Минимальное определение "*block tape segment* " и без "*define block move linear*" :-

```

define block tape segment
  N ; " ( Tape segment comment )"
end define

```

DUCTPost 1100 и ранее	Действие подъемы до сегментации	Действие возврата после сегментации
<i>retract distance = 0</i>	Без отвода	Перемещение в следующую точку после вставки блока сегментации на подаче резания
<i>retract distance = -999</i> (Или любое другое отрицательное число)	Отвод на (заданное значения) mm выше точки Lift Off на ускоренной подаче	Перемещение в следующую точку после вставки блока сегментации но без возврата в точку Lift Off (***Potential Disaster***)
<i>retract distance = 100</i>	plunges100 mm от точки Lift Off на подаче резания (***Disaster***)	Как выше (***Already a Disaster***)
DUCTPost 1205	Действие подъемы до сегментации	Действие возврата после сегментации
<i>tape split retract distance = 0</i>	Без отвода	Перемещение в следующую точку после вставки блока сегментации на ускоренной подаче
<i>tape split retract distance = -999</i> (Или любое другое отрицательное число)	Отвод инструмента в позицию FromZ на ускоренной подаче	Возврат в точку Lift Off point после вставки блока сегментации на подаче резания
<i>tape split retract distance = 100</i>	Отвод инструмента на 100 mm от точки Lift Off на ускоренной подаче	Как выше

Примечание: В вышеупомянутых действиях **FromZ** и **SafZ** могут быть рассмотрены как одно и тоже, даже если у них разные значения в DP1203-1205, но не для DP1100.

Рекомендуется:

Желательно иметь следующий формат для "*define block move segment*", и гарантированным определением помещения "*define block move linear*".

define block tape segment

```
N ; " ( Tape segment )" ## Данные вставки  
N ; G1 00 =C ; X OldX ; X =C ; Y OldY ; Y =C ; Z SafeZ ; M1 ; M2  
N ; G1 01 ; abs data ; Z OldZ ; F Prat  
end define
```

DUCTpost 1206 Пересмотренный

Разбиение программы

Настройки такие же как у предыдущих.

DUCTPost1206	Lift Action at Tape Split	Return Action after Tape Split
<i>tape split retract distance = 0</i>	Без отвода	Возврат в StartZ на ускоренной подаче, затем в точку Lift Off с подачей резания
<i>tape split retract distance = -999</i> (Или любое другое отрицательное число)	Отвод инструмента в позицию FromZ на ускоренной подаче	Как выше
<i>tape split retract distance = 100</i>	Отвод инструмента на 100 mm от точки Lift Off на ускоренной подаче	Как выше

Сегментация программы

DUCTPost1206	Lift Action at Segmentation	Return Action after Tape Segment
<i>tape split retract distance = 0</i>	Без отвода	Перемещение в следующую точку после вставки блока сегментации на подаче резания
<i>tape split retract distance = -999</i> (Или любое другое отрицательное число)	Отвод инструмента в позицию FromZ на ускоренной подаче	Возврат в точку Lift Off point после вставки блока сегментации на подаче резания
<i>tape split retract distance = 100</i>	Отвод инструмента на 100 mm от точки Lift Off на ускоренной подаче	Как выше

Постоянная контурная скорость

При каждом ускоренном перемещении и перемещении со скоростью подачи, угловые отклонения между предыдущим и последующим перемещениями проверяется замедление в конце небольшого перемещения и выводится код

Пример:

```
move safe angles = ( 0 5 -999 -999 )
```

```
define codes
```

```
constant contour speed = M2 90
```

```
end define
```

В этом случае в конце каждой строки где угловое изменение меньше 5 градусов, вставляется M90.

Имеется возможность (хотя редко используется) вставить два отдельных кода для разных диапазонов углов.

Пример:

```
move safe angles = ( 0 20 180 360 )
```

```
define codes
```

```
constant contour speed = G6 11
```

```
constant contour speed 2 = G6 12
```

```
end define
```

В этом случае в конце каждой строки где угловое изменение менее 20 градусов, вставляется G11, а где изменение более 180градусов - G12.

Важно:

По умолчанию проверка угла между считываемой и предыдущей строками размещает M90 на правильной строке для системы Heidenhain.

Для других станков наберите:

```
integer 77 = 2
```

что выводит код на строке *перед* малыми угловыми изменениями.

Компенсация Радиуса инструмента

Начиная с версии 1331 Ductpost и PowerMill 3.0 была пересмотрена функция компенсации радиуса инструмента, а именно 2 настройки теперь не используются.

Старые настройки использовали целочисленные значения 34 , 35 , 36, 37 (см. [Array Data](#)), из них сейчас устарели 34 и 36. И тем не менее необходимо определить вывод ключей и кодов компенсации и сгенерировать определение *блока линейных перемещений*:

Пример:

```
integer      34 35 36 37
comp output  = ( 0 1 0 1 ) ## Если используется этот формат, тогда должны быть
определены все 4 элемента.

или

integer 34   = 0  ## Устаревший
integer 35   = 1  ## Вывод Радиуса компенсации D (Обычно D ToolNum ) [По умолчанию]
integer 36   = 0  ## Obsolete
integer 37   = 1  ## Вывод Радиуса компенсации кодами G41 G42 G40 [По умолчанию]

define codes
  comp off      = G2 40
  comp on left  = G2 41
  comp on right = G2 42
end define

define keys
  tool radius = D
end define

define block move linear
  N ; linear ; G2 ; tool radius ; x coordinate ; y coordinate ; z coordinate ; feedrate ; M1
end define
```

Радиус инструмента может быть размещен как показано выше или перед "**feedrate**", но **НЕ ПОСЛЕ**, для большинства систем управления ЧПУ.

Особый случай - Маho

Маho в некоторых случаях требует вывода кода **G43** с компенсацией G41 или G42, что может быть получено следующим образом.

```
define block move rapid
  N ; G1 ; x coord ; y coord ; z coord
  set swa
end define

define block move linear
  if ( ToolComp = 41 or ToolComp = 42 and swa )
    N ; linear ; z coord ; feedrate
    N ; G2 43 ; x coord ; y coord ; z coord ; feedrate ; M1 ; M2
    unset swa
  else
```

```
    N ; linear ; G2 ; x coord ; y coord ; z coord ; tool radius ; feedrate ; M1 ; M2  
end if  
end define
```

Что обеспечит следующий вывод:

```
N26 G01 Z12. F750  
N28 G43  
N30 G01 G41 X56.779 D12 F1000
```

Drilling -Tapping Cycles

Ductpost первоначально предназначался для настройки Групповых Циклов для **DUCT**, к сожалению **PowerMill** работает по несколько другим правилам, что привело к путанице в некоторых случаях. Но все же, желательно потратить немного времени для настройки Ductpost, для дальнейшей комфортной работы.

Базовая структура Групповых циклов описана ниже, и в первую очередь имеет отношение к выводу для 3-осевых операций.

Имеется четыре блока для вывода групповых циклов:

define block cycle start

.....

end define

define block move cycle

.....

end define

define block move tap

.....

end define

define block cycle end

.....

end define

В большинстве случаев во встроенных исходных кодах **cycle start** не используется, т.к. параметры цикла выводятся вместе с первым перемещением. Вывод **cycle end** обычно **G80** (**конец сверления**)

Стандартные встроенные циклы

Коды для различных типов Групповых циклов обычно задаются групповым именем **G4** как показано ниже. Однако исключением является Heidenhain, классический пример чего приведен во втором столбце:

Система ЧПУ Fagor		Система ЧПУ Heid400	
define codes		define codes	
dwell	= G1 4	dwell	= G1 4
drill	= G4 81	drill	= G4 1
break chip	= G4 82	break chip	= G4 1
deep drill	= G4 83	deep drill	= G4 1
tap	= G4 84	tap	= G4 2
bore 1	= G4 85	bore 1	= G4 1
bore 2	= G4 86	bore 2	= G4 1
bore 3	= G4 87	bore 3	= G4 1
bore 4	= G4 88	bore 4	= G4 1
bore 5	= G4 89	bore 5	= G4 1
end of drill	= G4 80	end of drill	not used
cycle retract	not used	cycle retract	not used
end define		end define	

Различные требуемые *параметры* цикла настраиваются ключами или с помощью *переменных имен* блоков.

Используя ключи

Система ЧПУ Fagor	Система ЧПУ Heid400
<pre>define keys clearplane = R2 drill peck depth not used drill hole depth = Z2 cycle dwell not used ++ dwell = X ** end define</pre>	<pre>define keys clearplane = R2 drill peck depth = Z3 drill hole depth = Z2 cycle dwell not used ++ dwell = DW ** end define</pre>
** для DUCT и PM2.5	** для DUCT и PM2.5
++ для PM3.0 > (см. Примечания ниже)	++ для PM3.0 > (см. Примечания ниже)

Пример 2-х типичных встроенных настроек групповых циклов показан ниже:

Система ЧПУ Fagor	Система ЧПУ Heid400
<pre>define block cycle start end define</pre>	<pre>define block cycle start if (cycle != 4.) N ; " CYCL DEF 1.0 PECKING" N ; G4 =C ; clearplane =C N ; G4 =C ; drill hole depth =C N ; G4 =C ; drill peck depth =C N ; ".4 DWELL 0,000" ; G4 =C N ; G4 =C ; ".5" ; FF =C else N ; " CYCL DEF 2.0 TAPPING" N ; G4 =C ; clearplane =C N ; G4 =C ; drill hole depth =C N ; ".3 DWELL 0,000" ; G4 =C N ; G4 =C ; ".4" ; FF =C end if end define</pre>
<pre>define block move cycle N ; X ; Y ; G4 ; R2 ; Z2 end define</pre>	<pre>define block move cycle N ; G1 =C ; x coordinate =C ; y coordinate =C ; RR ; FF 9999 N ; " CYCL CALL M" N ; G1 =C ; z coordinate =C ; RR =C ; FMAX =C end define</pre>
<pre>define block move tap N ; X ; Y ; G4 ; R2 ; Z2 end define</pre>	<pre>define block move tap end define</pre>
<pre>define block cycle end N ; G4 80 end define</pre>	<pre>define block cycle end end define</pre>

Используя переменные:

Доступные **переменные** используемые для определения блока:

ClearPlane	ClearPlaneInc	Cycledwell
HoleDepth	HoleDepthInc	HoleTop
HoleDiameter	PeckDepth	PeckDepthInc
Cycfed		

Пример:

```
define block cycle start
  N ; G4 ; G6 ; Z2 HoleDepth ; R2 ClearPlane ; Q PeckDepth ; feedrate
end define
```

Примечание: Заглавные и Строчные буквы для *переменных* – используется для дифференцирования *ключа* или *кода*, смотри ниже.

```
define block move cycle
  N ; G4 ; x coord ; y coord ; z coord ; feedrate
end define
```

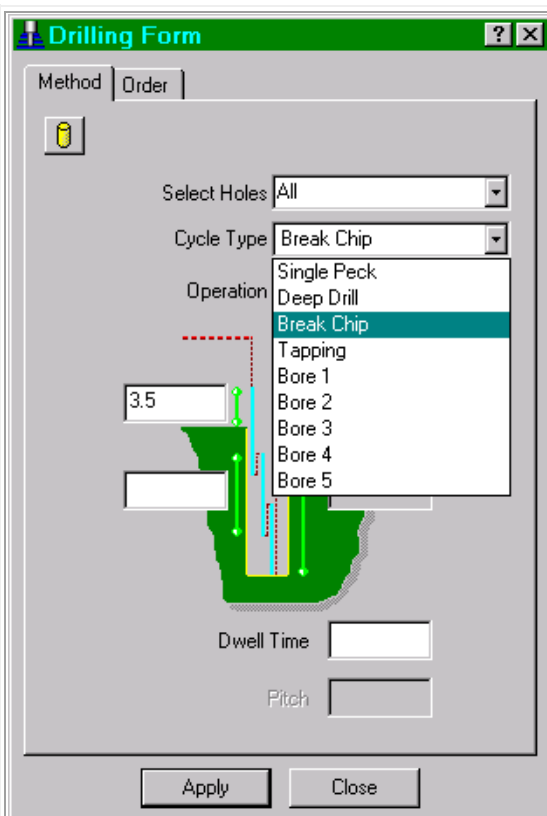
Однако, мы рекомендуем следующий формат ключа:

```
define block cycle start
  N ; G4 ; G6 ; drill hole depth ; clearplane ; drill peck depth ; cycle dwell ; feedrate Prat
end define
```

Примечание: Используйте только **строчные** буквы для *ключа* или *кода*.

Типы циклов

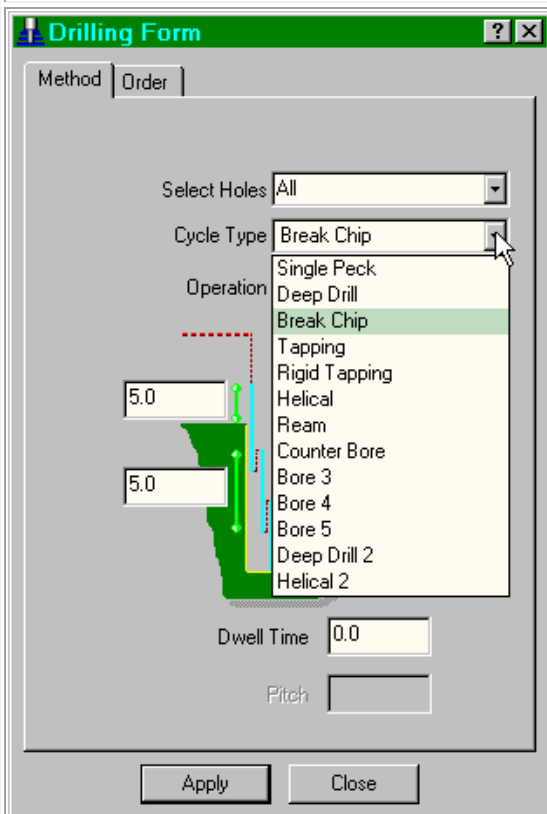
Форма имеющихся в PowerMill для вывода циклов сверления и понятные Ductpost указана ниже:



До PM4.0

Тип Цикла [DP Cycle Ref.](#)

Single Peck (простое)	= cycle 1
Break Chip (с ломкой стружки)	= cycle 2
Deep Drill (глубокое)	= cycle 3
Tapping (нарезание резьбы)	= cycle 4
Bore 1 (расточка)	= cycle 5
Bore 2	= cycle 6
Bore 3	= cycle 7
Bore 4	= cycle 8
Bore 5	= cycle 9



После PM4.0

Cycle Type [DP Cycle Ref.](#)

Single Peck (простое)	= cycle 1
Break Chip (с ломкой стружки)	= cycle 2
Deep Drill (глубокое)	= cycle 3
Tapping (нарезание резьбы)	= cycle 4
Rigid Tapping (жесткое нарезание резьбы)	= cycle 10
Helical (спираль)	= cycle 11
Ream (развертка)	= cycle 5
Counter Bore (обратная расточка)	= cycle 6
Bore 3 (расточка)	= cycle 7
Bore 4	= cycle 8
Bore 5	= cycle 9
Deep Drill 2	= cycle 13
Helical 2	= cycle 12

Рекомендуется использовать в циклах формулировки "if", как показано выше для Heid400, где "if (cycle != 4)" обеспечивает действие первого блока кода.

Подпрограммы цикла в Ductpost слишком просты, и не полностью соответствуют более универсальному выводу из PowerMill, и требованиям современных систем ЧПУ. С помощью формулировки "*if*" можно гарантированно задать формат ввода для большинства случаев.

Формат вывода Цикла (Предупреждение)

Целочисленный флажок «*cycle output*» использовался для определения вывода цикла из DUCT в виде G-кода, или линейными перемещениями.

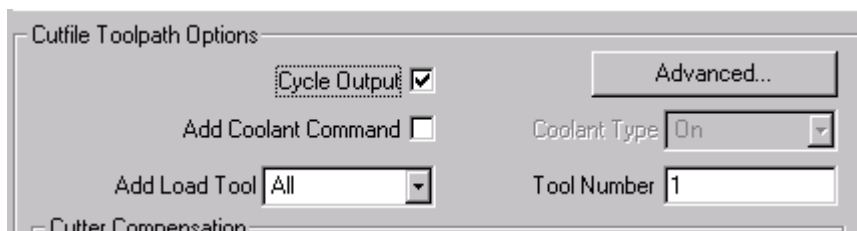
По умолчанию: *cycle output* = 1 – формат вывода в G-кодах.

Настройка: *cycle output* = 0 так же вывод в G-кодах, но с изменением последовательности цикла (см ниже)

<i>cycle output</i> = 1	<i>cycle output</i> = 0
N50X0Y0	N50X0Y0
N60X103.Y17.33	N60X103.Y17.33
N70G81Z-17.456R-10.456	N70G81 F100 Z-17.456R-10.456
N80G80	N80G80
N90G81 X179 .Z-17.448R-10.448	N90G81F100Z-17.448R-10.448
N100G80	N100 X179 .
	N110G80

Т.к. это противоречит тому что нужно сделать, рекомендуется для игнорирования этого метода управления и если требуется вывод линейных перемещений для вывода циклов тогда используют: PowerMill's - Tool Path Output Form - Cycle Output предпочтительно выбрать:

До PM4.0



NC Program : 25mmDeepHoles

Name: 25mmDeepHoles [Options...]

Output File: E:/users/am/am-user/25mmDeepHoles.cut

Machine Option File: standard

Output Workplane: 1 [v]

Part Name: to_-25

3D Machine Compensation: Off [v]

Tool Value: Tip [v]

Automatic Tool Alignment: On [v]

Connection Moves: Move, Rotate [v]

Toolpath	Number	Diameter	Tip	Length	Overhang	Tool ID	Type	Tolerance
to_-25	1	7.9		39.5		1	DRILL	0.01

Reset Tool Change: On Change [v] Tool Numbering: As Specified [v]

Toolpath to_-25

Tool

Tool Number: 1 Tool Length: 39.5

Tool ID: 1

Cutter Compensation

Length: Off [v] 0 Radius: None [v]

Arc Fit: Off [v] Coolant: Standard [v]

Drilling Cycle Output: On [v] [On] [Off]

Output File: [v]

Comments...

Write Apply Accept Close

Вывод Цикла не проверяется, если для цикла требуется формат линейных перемещений.

ВСЕГДА не контролируется для 3 + 2 углового сверления, если система может работать с такими групповыми циклами, например Heidenhain 430 через функцию CYCL DEF 19.0.

(Однако это не возможно осуществить с версиями PowerMill 3.0 и Ductpost1331 и более ранними)

Сверление, Нарезание резьбы и т.д. в DP1335 & PM4.0 и позже.

Два примера приведенные ниже поясняют требуемые изменения встроенных настроек цикла.

Код ISO (Fanuc11m) Встроенный исходный файл	Код ISO (Fanuc11m) Option-файл [изменения]
<pre> define keys cycle dwell not used dwell = X ## Старый DUCT drill peck depth = Q1 drill hole depth = Z2 clearplane = R2 end define # define codes drill = G4 81 break chip = G4 82 deep drill = G4 83 tap = G4 84 bore 1 = G4 85 bore 2 = G4 86 bore 3 = G4 87 bore 4 = G4 88 bore 5 = G4 89 end of drill = G4 80 cycle retract = G6 99 end define # cycle output = 1 integer 69 = 0 ## По умолчанию подача при нарезании резьбы обычно 85% плунжерной подачи (не показано) # define block cycle start N ; S ; M1 N ; G3 ; G5 ; M2 F =C ; G4 =C ; Z2 ; R2 ; Q ; Q1 ; G6 ; end define # </pre>	<pre> define format (P) field width = 3 tape position = 1 modal metric formats decimal point = true decimal places = 2 trailing zeros = false imperial formats = metric formats end define # word order = (+ P) # define keys cycle dwell = P dwell not used end define # define codes cycle retract = G6 98 ##Может быть 98 или 99 end define # block order = true integer 69 = 2 ## Сброс подачи нарезания резьбы в обычную плунжерную подачу.(cysfed или Prat) # define block cycle start if (cycle <= 2 or cycle => 5) N ; G4 ; G6 ; x coord =C ; y coord =C ; drill hole depth ; clearplane ; cycle dwell ; feedrate Prat end if if (cycle == 3) N ; G4 ; G6 ; x coord =C ; y coord =C ; drill hole depth ; clearplane ; drill peck depth ; cycled well ; feedrate Prat end if if (cycle == 4) N ; G4 ; G6 ; x coord =C ; y coord =C ; drill hole depth ; clearplane ; cycle dwell ; feedrate cysfed ; end if end define # </pre>

define block move cycle N ; G4 ; G6 ; X ; Y ; Z2 ; R2 ; Q ; Q1 ; F ; M2 end define #	define block move cycle N ; x coord ; y coord ; M1 ; M2 end define #
define block move tap N ; G6 ; G4 ; X ; Y ; Z2 ; R2 ; F ; M1 ; M2 end define #	define block move tap N ; x coord ; y coord ; M1 ; M2 end define #
define block cycle end G4 80 end define	define block cycle end N ; end of drill end define

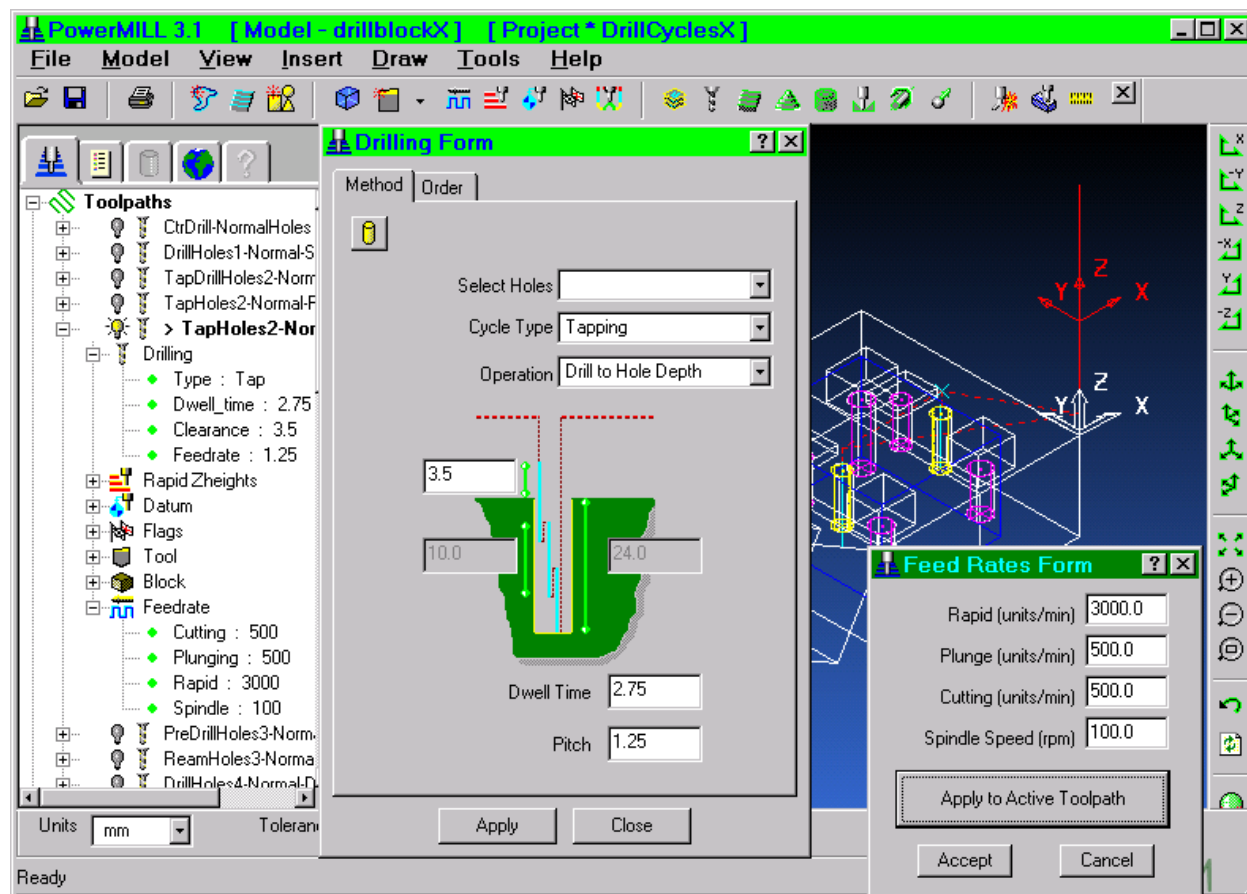
Следующий вывод программ ЧПУ в двух примерах, использует один и тот же cut-файл и поясняет форматы для выполнения операций для двух отверстий в каждом случае: Простое сверление **T2 G81**; Сверление с ломкой стружки **T3 G82**; Нарезание резьбы **T4 G84**; Расточка 1 (Развертка) **T6 G85** и Глубокое сверление **T7 G83**:

NC-Вывод для исходного файла	NC-Вывод для Option-файла
T2 M6 X0 Y0 S1500 M3 Z60. Z-40. X-24.998 Y24.998 G81 G99 Z-84. R-60.5 F100 X-74.998 G80 T3 M6 G43 Z60. H3 G0 X-25.001 Y44.999 S1500 M3 Z-40. G82 G99 Z-74. R-46.5 F100 G80 G82 G99 X-75.001 Z-84. R-46.5 F100 G80 T4 M6 G43 Z60. H4 G0 X-25.001 Y44.999 S100 M3 Z-40. G84 G99 Z-74. R-46.5 F118 G80 G84 G99 X-75.001 Z-84. R-46.5 F118 G80 T6 M6 G43 Z60. H6 G0 X-24.999 Y64.998 S800 M3 Z-40. G85 G99 Q5. Z-74. R-50.5 F110 G80 G85 G99 X-74.999 Q5. Z-84. R-46.5 F110 G80 T7 M6 G43 Z60. H7	T2 M06 (10.000mm Drill) G0 G90 X0 Y0 B0 S1500 M03 G43 Z60. H2 X-24.998 Y24.998 G81 G98 X-24.998 Y24.998 Z-84. R-60.5 P0 F100 X-74.998 G80 T3 M06 (6.800mm TapDrill) G0 G90 X-25.001 Y44.999 B0 S1500 M03 G43 Z60. H3 G82 G98 X-25.001 Y44.999 Z-74. R-46.5 P0 F100 G80 G82 G98 X-75.001 Y44.999 Z-84. R-46.5 P0 F100 G80 T4 M06 (8.000mm Tap7) G0 G90 X-25.001 Y44.999 B0 S100 M03 G43 Z60. H4 G84 G98 X-25.001 Y44.999 Z-74. R-46.5 P2.75 F125 G80 G84 G98 X-75.001 Y44.999 Z-84. R-46.5 P2.75 F125 G80 T6 M06 (8.000mm Reamer) G0 G90 X-24.999 Y64.998 B0 S800 M03 G43 Z60. H6 G85 G98 X-24.999 Y64.998 Z-74. R-50.5 P1.5 F110 G80 G85 G98 X-74.999 Y64.998 Z-84. R-46.5 P1.5 F110 G80 T7 M06 (10.000mm Drill)

G0 X-24.999 Y84.997 S1250 M3 Z-40. G83 G99 Q4. Z-94. R-60.5 F120 X-74.999 G80	G0 G90 X-24.999 Y84.997 B0 S1250 M03 G43 Z60. H7 G83 G98 X-24.999 Y84.997 Z-94. R-60.5 P2.25 Q4. F120 X-74.999 G80
Точки Примечания	Точки Примечания
Отсутствие в приведенном выше выводе вызова паузы " G4 Xn.nn ", т.к. PowerMill не использует это в новых функциях сверления, а использует вместо этого " cycle dwell ". # Особое внимание нужно уделить выводу позиций отверстий, т.к. настройки по умолчанию, в некоторых случаях могут привести к сверлению первого отверстия дважды или к пропуску обоих. # Обратите внимание на скорость подачи Цикла нарезания резьбы G84 и сравните с выводом напротив. Это вызвано " integer 69 "	Пауза была определена новым словом P и назначена ключом " cycle dwell " с символом адреса P. # В приведенном выше примере, позиция первого отверстия всегда выводится со строкой данных G8X, даже в процессе перемещения в позицию отверстия. Корректность этого зависит от системы ЧПУ, некоторые системы ЧПУ нуждаются в определении позиции сверления после строки G8X # Обратите внимание на разные скорости подачи нарезания резьбы G84 по сравнению с предыдущим, это настраивается " integer 69 ".

Вывод Шага резьбы из PowerMill PM3.0

Некоторые системы ЧПУ требуют вывод подачи для нарезания резьбы как **F Pitch (ua2)** и получения настройки шага из PowerMill Drill Form (см. ниже)



В вышеупомянутом примере определено следующее: Шаг = 1.25 (M8) – Подача погружения = 500 units/min – Скорость вращения шпинделя = 100 rpm, две последние настройки скорости для размеров Резьбы и Шага должны быть определены очень внимательно, поскольку **Скорость подачи (cycfed)** вычисляются PowerMill по формуле **Подача = Шаг x Скорость шпинделя**.

Для правой резьбы пример изменений option-файла отображен ниже:

<pre># define block cycle start :.....: if (cycle == 4) N ; G4 ; G6 ; X =C ; Y =C ; R2 ; Z2 ; P cycldwell ; F cycfed ; end if end define #</pre>	<pre>define word PF address letter = "F" address width = 1 field width = 3 tape position = 1 modal metric formats decimal point = true decimal places = 2 trailing zeros = false imperial formats = metric formats end define # word order = (+ P PF) # define block cycle start :.....: if (cycle == 4) N ; G4 ; G6 ; x coord =C ; y coord =C ; drill hole depth ; clearplane ; cycle dwell ; PF (cycfed / ToolSpeed) end if end define # define block move tap N ; x coord ; y coord ; M1 ; M2 end define #</pre>
<pre>T4 M06 (8.000mm Tap) G0 G90 X0 Y0 B0 S100 M03 G43 Z60. H4 X-25.001 Y44.999 G84 G98 X-25.001 Y44.999 Z-74. R-46.5 P2.75 F125 G80 G84 G98 X-75.001 Y44.999 Z-84. R-46.5 P2.75 F125 G80</pre>	<pre>T4 M06 (8.000mm Tap) G0 G90 X-25.001 Y44.999 B0 S100 M03 G43 Z60. H4 G84 G98 X-25.001 Y44.999 Z-74. R- 46.5 P2.75 F1.25 G80 G84 G98 X-75.001 Y44.999 Z-84. R- 46.5 P2.75 F1.25 G80</pre>

Примечание: Для скорости подачи Нарезания резьбы, мы не можем использовать обычное слово " F ", т.к. для Шага требуется другой формат, следовательно для этого необходимо создать новое слово.

Циклы Сверления на сложные современные системы ЧПУ

Heidenhain (Система ЧПУ TNC430 и моложе) & Ductpost1400 и моложе

Heidenhain, and other machine tool control manufactures, have started to introduce more complex cycles, mainly as Manual programing functions, but customers expect CAM systems to output the format so that machine operators can modify the program at the machine without recourse to going back to the programmer.

To achieve these required formats is proving taxing, and it is becoming essential that all option file construction should adhere to certain fixed rules to ensure some conformity is maintained no matter, who is building them.

We have included the **OLD**, simpler type of **PECKING** cycle with the NEW which adds additional complexity that we consider maybe unjustifiable. (However, specific **exceptions** are made to this with **CYL DEF 1.0 PECKING** and the **CYCL DEF 5.0 CIRCULAR POCKET** format) (See Word document [cycle-records.rtf](#))

Heidenhain CYCL DEF 2XX Cycles

New Words are required to be defined for these, and some of the old ones re-defined, however, it is possible that if the option includes Feedrate Parameter and Spline function then there is a likelihood of running out of available word capacity with Ductpost versions earlier than DP1414.

If this occurs it is suggested that the Spline function is dropped as Heidenhain do not recommend its use with their controls.

Ductpost1414 will have a **95** word capacity.

The Cycles used are the ones indicated in the PowerMill4> Drill Form (See [Cycle Types](#))

We recommend the following new words and formatting in order to have some standardisation of use :-
(## Comments **only** - relate to PM Drill Form)

Words, formats, keys, codes, settings	Cycle Block definitions
<pre># vvvvvvv CYCLE WORDS vvvvvvv # define word CQ address letter = "Q" address width = 1 field width = 3 not modal tape position = -3 metric formats leading zeros = false trailing zeros = true decimal point = false decimal places = 0 imperial formats = metric formats end define # define word CF address letter = "=" address width = 1 field width = 5 tape position = 0</pre>	<pre>define block cycle start if (swa) N ; " ;" N ; TPN ToolPathName N ; " ; TOOLPATH" ; WPN ToolPathWorkPlaneName N ; " ;" unset swa end if # N ; CYC Cycle ; ")" if (cycle == 1) N ; " CYCL DEF 1.0 PECKING" set swj else if (cycle == 2) N ; G4 0 ; " DRILLING~" unset swj else if (cycle == 3) N ; G4 3 ; " UNIVERSAL DRILLING~" unset swj else if (cycle == 4) N ; G4 6 ; " TAPPING NEW~"</pre>

```

print position = 0
not modal
metric formats
leading zeros = false
trailing zeros = true
decimal point = false
decimal places = 0
imperial formats = metric formats
end define
#
define word C1
  address letter = "="
  sign = always
end define
#
define word C2
  address letter = "="
  sign = none
end define
#
define word C3
  address letter = "="
  address width = 1
  field width = 2
  tape position = 0
  sign = none
  decimal point = false
  decimal places = 0
  trailing zeros = false
  imperial formats = metric formats
end define
#
define word C4
  address letter = " "
  address width = 1
  sign = if negative
end define
#
define word CYC
  address letter = "; (Cycle= "
  address width = 10
  field width = 2
  tape position = 1
end define
#### This last word is not normally
required
but may be of help in debugging
options.
#
# vvvvvvv CYCLE WORDS Format
vvvvvvv
#
define format ( C1 C2 C4 )

```

```

unset swj
else if ( cycle == 5 )
  N ; G4 1 ; " REAMING~"
  unset swj
else if ( cycle == 6 )
  N ; G4 8 ; " BORE MILLING~"
  unset swj
else if ( cycle == 7 )
  N ; " CYCL DEF 5.0 CIRCULAR POCKET"
  set swj
else if ( cycle == 8 )
  N ; G4 14 ; " CIRCULAR POCKET FINISHING~"
  unset swj
else if ( cycle == 9 )
  N ; G4 2 ; " BORING~"
  unset swj
else if ( cycle == 10 )
  N ; G4 7 ; " RIGID TAPPING NEW~"
  set swi
  unset swj
else if ( cycle == 11 or cycle == 12 )
  error "THIS IS NOT A SUPPORTED CYCLE-TURN
CYCLES OFF IN POWERMILL"
else if ( cycle == 13 )
  N ; G4 5 ; " UNIVERSAL PECKING"
  unset swj
end if
##-0
if ( not swj )
  CQ 200 ; clearplane ; E ; " CLEARPLANE~"
  CQ 201 ; drill hole depth ; E ; " HOLE DEPTH~"
  if ( not swi )
    CQ 206 ; CF Prat ; E ; " PLUNGE
FEEDRATE~"
  end if
end if
#
##-1
if ( cycle == 1 )
  N ; " CYCL DEF 1.1 SET UP" ; C4 ( Clearplane -
HoleTop )
  N ; " CYCL DEF 1.2 DEPTH" ; C4 ( HoleDepth -
HoleTop )
  N ; " CYCL DEF 1.3 PECKG" ; C4 PeckDepth
  N ; " CYCL DEF 1.4 DWELL" ; TL CycleDwell
  N ; " CYCL DEF 1.5" ; FF Prat
  edit intd 1
##-2
else if ( cycle == 2 )
  CQ 202 ; Z3 PeckDepth ; E ; " PECKING
DEPTH~"
  CQ 210 ; DW 0 ; E ; " DWELL TIME
AT TOP~"

```

```

address width = 1
field width   = 8
tape position = 0
print position = 0
not modal
metric formats
leading zeros = false
trailing zeros = true
decimal point = true
decimal places = 3
imperial formats = metric formats
end define
#
define format ( G4 )
  address letter = "CYCL DEF 2"
  address width = 11
  field width   = 2
  not modal
  leading zeros = true
  imperial formats = metric formats
end define
#
define format ( G6 )
  field width   = 3
  decimal places = 1
  decimal point = true
  not modal
  imperial formats = metric formats
end define
#
define format ( TL )
  address letter = " "
  address width = 0
  field width   = 5
  sign          = none
  tape position = 1
  metric formats
  decimal point = true
  decimal places = 3
  trailing zeros = false
  imperial formats = metric formats
end define
#
define format ( Z2 )
  address letter = "=-"
  address width = 2
  sign          = none
  tape position = 0
  print position = 0
end define
#
define format ( R2 Z3 )
  address letter = "="
  CQ 203 ; C1 HoleTop          ; E ; " SURFACE
  COORDINATE~"
  CQ 204 ; C2 ( SafZ - Holetop ) ; E ; " 2ND SET-UP
  CLEARANCE~"
  CQ 211 ; DW CycleDwell =C    ; E ; " DWELL TIME
  AT BOTTOM "
  ##-3
  else if ( cycle == 3 )
    CQ 202 ; Z3 PeckDepth      ; E ; " PECKING
    DEPTH~"
    CQ 210 ; DW 0              ; E ; " DWELL TIME
    AT TOP~"
    CQ 203 ; C1 Holetop        ; E ; " SURFACE
    COORDINATE~"
    CQ 204 ; C2 ( SafZ - Holetop ) ; E ; " 2ND SET-UP
    CLEARANCE~"
    CQ 212 ; C2 ( HoleDepth / NumberPecks ) ; E ; "
    PECK DECREMENT~"
    CQ 213 ; C3 ( NumberPecks / 2 )          ; E ; "
    NUMBER OF PECKS~"
    CQ 205 ; C2 ( NumberPecks / Word{TD} ) ; E ; " MIN.
    PECK DEPTH~"
    CQ 211 ; DW CycleDwell =C    ; E ; " DWELL TIME
    AT BOTTOM~"
    CQ 208 ; CF Srat            ; E ; " RETRACT
    FEEDRATE~"
    CQ 256 ; C2 200            ; E ; " CHIPBREAK
    DIST. "
    ##-4
    else if ( cycle == 4 )
      CQ 211 ; DW CycleDwell      ; E ; " DWELL TIME
      AT BOTTOM~"
      CQ 203 ; C1 Holetop          ; E ; " SURFACE
      COORDINATE~"
      CQ 204 ; C2 ( SafZ - Holetop ) ; E ; " 2ND SET-UP
      CLEARANCE~"
      ##-5
      else if ( cycle == 5 )
        CQ 211 ; DW CycleDwell      ; E ; " DWELL TIME
        AT BOTTOM~"
        CQ 208 ; CF Srat            ; E ; " RETRACT
        FEEDRATE~"
        CQ 203 ; C1 Holetop          ; E ; " SURFACE
        COORDINATE~"
        CQ 204 ; C2 ( SafZ - Holetop ) ; E ; " 2ND SET-UP
        CLEARANCE~"
        ##-6
        else if ( cycle == 6 )
          CQ 334 ; C2 ( Word{TD} / 3 ) ; E ; "DWELL TIME
          AT BOTTOM~"
          CQ 203 ; C1 Holetop          ; E ; "SURFACE
          COORDINATE~"
          CQ 204 ; C2 ( SafZ - Holetop ) ; E ; "2ND SET-UP

```

<pre> address width = 1 sign = none tape position = 0 print position = 0 end define # define format (R2) scale factor = 1 end define # define format (DW) address letter = "=" address width = 1 field width = 3 sign = none decimal places = 2 print position = 0 end define # define format (E) address letter = ";" field width = 0 permanent not modal tape position = -18 end define # word order = (+ CQ CF C1 C2 C3 C4 Q E DW) # define keys cycle dwell = DW dwell not used end define # block order = true use hole top in cycles = true integer 69 = 2 </pre>	<pre> CLEARANCE~" CQ 335 ; CF HoleDiameter ; E ; "NOMINAL DIAMETER" CQ 342 ; CF 0 ; E ; "ROUGHING DIAMETER" ##-7 else if (cycle == 7) N ; " CYCL DEF 5.1 SET UP" ; C4 (Clearplane - HoleTop) N ; " CYCL DEF 5.2 DEPTH" ; C4 (HoleDepth - HoleTop) N ; " CYCL DEF 5.3 PLUNGING" ; C4 (Prat / ToolSpeed) ; FF Prat N ; " CYCL DEF 5.4 RADIUS" ; C4 (HoleDiameter / 2) N ; " CYCL DEF 5.5" ; FF Frat ; " DR+" edit intd 1 ##-8 else if (cycle == 8) CQ 202 ; Z3 (Word{TD} / 3) ; E ; "PLUNGING DEPTH~" CQ 207 ; CF Frat ; E ; "FEED RATE FOR MILLING~" CQ 203 ; C1 Holetop ; E ; "SURFACE COORDINATE~" CQ 204 ; C2 (SafZ - Holetop) ; E ; "2ND SET-UP CLEARANCE~" CQ 216 ; C2 xcoord =C ; E ; "CENTER IN 1st AXIS~" CQ 216 ; C2 ycoord =C ; E ; "CENTER IN 2nd AXIS~" CQ 222 ; C2 Word{TD} ; E ; "WORKPIECE BLANK DIA.~" CQ 223 ; C2 HoleDiameter ; E ; "FINISHED PART DIA." ##-9 else if (cycle == 9) CQ 211 ; DW cycledwell ; E ; "DWELL TIME AT BOTTOM~" CQ 208 ; CF Srat ; E ; "RETRACT FEEDRATE~" CQ 203 ; C1 Holetop ; E ; "SURFACE COORDINATE~" CQ 204 ; C2 (SafZ - Holetop) ; E ; "2ND SET-UP CLEARANCE~" CQ 214 ; C3 1 ; E ; "DISENGAGING DIRECTN~" CQ 336 ; C3 0 ; E ; "ANGLE OF SPINDLE" ##-10 else if (cycle == 10) CQ 239 ; C2 (Cycfed / ToolSpeed[ToolNum]) ; E ; " THREAD PITCH~" </pre>
--	--

```

CQ 203 ; C1 HOLETOP ; E ; " SURFACE
COORDINATE~"
CQ 204 ; C2 ( SafZ - HOLETOP ) ; E ; " 2ND SET-
UP CLEARANCE"
unset swi
##-13
else if ( cycle == 13 )
CQ 202 ; Z3 PeckDepth ; E ; " PECKING
DEPTH~"
CQ 203 ; C1 HOLETOP ; E ; " SURFACE
COORDINATE~"
CQ 204 ; C2 ( SafZ - HOLETOP ) ; E ; " 2ND SET-UP
CLEARANCE~"
CQ 212 ; C2 ( HoleDepth / NumberPecks ) ; E ; "
PECK DECREMENT~"
CQ 205 ; C2 ( NumberPecks / Word{TD} ) ; E ; " MIN.
PECK DEPTH~"
CQ 258 ; C2 500 ; E ; " UPPER ADV
STOP DIST~"
CQ 259 ; C2 1000 ; E ; " LOWER ADV
STOP DIST~"
CQ 257 ; C2 ( PeckDepth / 2 ) ; E ; " DEPTH FOR
CHIP BRKNG~"
CQ 256 ; C2 200 ; E ; " CHIPBREAK
DIST.~"
CQ 211 ; DW CycleDwell ; E ; " DWELL TIME
AT BOTTOM~"
end if
end define
#
define block move cycle
if ( Cycle == 1 or Cycle == 7 and intd == 1 )
N ; G1 ; x coord ; y coord ; FMAX
N ; G1 ; z coord Clearplane ; FMAX
N ; G1 ; x coord =C ; y coord =C ; RR 0 ; FMAX ; "
M99"
edit intd 0
else
N ; G1 ; x coord =C ; y coord =C ; RR ; FMAX ; "
M99"
end if
end define
#
define block move tap
N ; G1 ; x coord =C ; y coord =C ; RR ; FMAX ; " M99"
end define
#
define block cycle end
N ; G1 ; z coord Saf Z =C ; RR ; FMAX
end define
#
Note :- M99 must be a string, not M1 99 or M2 99

```

The above provides the following output :- (See illustrated [pictures](#) as a reference)

NOTE: This is a TIF file and will issue a warning - it is safe to use.

These are ChipBreak Drilling and Deep Drill 2 Cycles	These are the <i>EXCEPTIONS</i> - and use the Single Peck and Bore 3 Cycle
79 L X-3.188 Y-10.79 FMAX 80 ; CYCLE=(2) ## Indicator of cycle only, can be removed 81 CYCL DEF 200 DRILLING~ Q200=5.000 ; CLEARPLANE~ Q201=-36.949 ; HOLE DEPTH~ Q206=333 ; PLUNGE FEEDRATE~ Q202=36.950 ; PECKING DEPTH~ Q210=0.0 ; DWELL TIME AT TOP~ Q203=+0.0 ; SURFACE COORDINATE~ Q204=55.290 ; 2ND SET-UP CLEARANCE~ Q211=3.25 ; DWELL TIME AT BOTTOM 82 L X-3.188 Y-10.79 FMAX M99 83 L Z55.29 FMAX 98 L X10.0 Y16.5 FMAX M03 99 L Z20.0 FMAX M08 100 L Y12.5 FMAX 101 ; CYCLE=(13)## Indicator of cycle only, can be removed 102 CYCL DEF 205 UNIVERSAL PECKING Q200=4.999 ; CLEARPLANE~ Q201=-5.000 ; HOLE DEPTH~ Q206=750 ; PLUNGE FEEDRATE~ Q202=1.500 ; PECKING DEPTH~ Q203=+0.0 ; SURFACE COORDINATE~ Q204=20.000 ; 2ND SET-UP CLEARANCE~ Q212=0.132 ; PECK DECREMENT~ Q205=6.333 ; MIN. PECK DEPTH~ Q258=0.500 ; UPPER ADV STOP DIST~ Q259=1.000 ; LOWER ADV STOP DIST~ Q257=0.750 ; DEPTH FOR CHIP BRKNG~ Q256=0.200 ; CHIPBREAK	53 L X18.0 Y2.34 FMAX 54 ; CYCLE=(1)## Indicator of cycle only, can be removed 55 CYCL DEF 1.0 PECKING 56 CYCL DEF 1.1 SET UP 5.000 57 CYCL DEF 1.2 DEPTH -36.950 58 CYCL DEF 1.3 PECKG 36.950 59 CYCL DEF 1.4 DWELL 1.5 60 CYCL DEF 1.5 F333 61 L Z5.0 FMAX 62 L X18.0 Y2.34 R0 FMAX M99 63 L Z55.29 FMAX 68 L Y16.5 FMAX 69 L X65.0 Y12.5 FMAX M08 70 ; CYCLE=(7)## Indicator of cycle only, can be removed 71 CYCL DEF 5.0 CIRCULAR POCKET 72 CYCL DEF 5.1 SET UP 5.000 73 CYCL DEF 5.2 DEPTH -5.000 74 CYCL DEF 5.3 PLUNGING 1.500 F750 75 CYCL DEF 5.4 RADIUS 5.000 76 CYCL DEF 5.5 F2000 DR+ 77 L Z5.0 FMAX 78 L X65.0 Y12.5 R0 FMAX M99 79 L Z20.0 FMAX

DIST.~ Q211=0.0 ; DWELL TIME AT BOTTOM~ 105 L X10.0 Y12.5 FMAX M99 106 L Z20.0 FMAX	
---	--

NOTES :-

1/ **Helical** and **Helical 2** are not strictly cycles and are treated as Linear movement by PowerMill and Ductpost, therefore the **Drilling Cycle Output** check box in the NC edit form must be **OFF** for these two functions.

2/ **Word{TD}** = ToolDiameter and is extracted from the Tool Change block data.
It **MUST** be defined as **TD (ToolRadius[ToolNum] * 2)** and **NOT** as previously by the word **TD** scaled by 2.

3/ It is probable that additional cycle functions will be added later and will follow on from **cycle 13**

4/ The defined cycles are not cast in stone and can be changed as felt necessary, and will be refined as required from feedback from users. However, we would advise that words given above are used wherever possible if and when a change is made to try and avoid confusion.

Циклы Сверления Siemens

Siemens drilling cycles are nearly as complex as Hiedenhain though not as informative friendly in format output.

The following is a breakdown of the codes applicable to the various cycle functions and they must be placed in the order indicated for each particular cycle for the cycle to work correctly. Siemens do not use a "Q" values as Hiedenhain, but a " **comma** " to separate each function, e.g. , **clearplane** , which can be a problem if a function is a zero value as then , , is acceptable and **must** still be included in the sequence. The following is an example output which has been artificially spaced to correspond to the individual sector. :-

(It is suggested that the MSG line is output to the NC program as a guide for the operator, it can always be suppressed if not wanted.)

Example of NC output :-

```

N12          MSG(RTP, RFP, SDIS, DP , DPR , FDEP, FDPR, DAM , DTB , DTS, FRF, VARI)
N13 MCALL CYCLE83(10. , -20. , 5. , -81.322, -61.322, -30.5, 10.5, 1.75, 3.55, , 1 , 1 )
N141          MSG(RTP, RFP, SDIS, DP , DPR, DTB, DTS, , PIT, SST , SST1")
N142 MCALL CYCLE84(10. , 0. , 3. , -45. , 45. , 1.5, , , 1.5, 3000, 4500 )

```

Note : CYCLE84 is for Rigid Tapping (Cycle10) : CYCLE840 is for Floating Tap holder (Cycle 4)

Reference Table to codes used :-

Code = Reference

RTP = Absolute SafeZ Height

DP = Drilled Hole Point Depth

FDPR = Incremental Peck Depth

DTS = Cycle Dwell at Hole Top

Code = Reference

RFP = Absolute HOLETOP Surface Coordinate

DPR = Full Hole Depth

DAM = Degression distance

FRF = 1st.Peck Feedrate Variation Factor

Code = Reference

SDIS = Incremental Clearplane Height

FDEP = 1st.Abs.Peck Coordinate

DTB = Cycle Dwell at Hole Bottom

VARI = 0 = ChipBreak. 1 = Swarf Out

Tapping Codes :-

PIT = Thread Pitch

SST = Tapping Tool Speed (RPM)

SDAC = Spindle Rotation Direction after Cycle

SST1 = Retract Tool Speed (RPM)

Words, formats, keys, codes, settings	Cycle Block definitions
<pre># vvvvvvvv CYCLE Words vvvvvvvv # define word RFP address letter = "," address width = 1 field width = 8 sign = if negative metric formats decimal point = true decimal places = 3 leading zeros = false trailing zeros = false end define # define word DP address letter = "," address width = 1 sign = if negative end define # define word Q1 address letter = "," address width = 1 end define # define word PT address letter = ", " address width = 3 field width = 5 sign = none</pre>	<pre># define block cycle start if (swa) N ; TPN ToolPathName ; " \")" unset swa end if N ; G3 ; feedrate Cycfed ; CYC Cycle ; ")" if (cycle == 1 or cycle == 5) N ; " MSG (RTP , RFP , SDIS , DP , DPR)" ** N ; G4 ; R11 SafeZ ; RFP HoleTop ; clearplane ; DP HoleDepth ; drill hole depth ; ")" else if (cycle == 2) N ; MSG (RTP ,RFP , SDIS , DP , DPR , DTB)" ** N ; G4 ; R11 SafeZ ; RFP HoleTop ; clearplane ; DP HoleDepth ; drill hole depth ; cycle dwell ; ")" else if (cycle == 3) N ; " MSG (RTP ,RFP ,SDIS ,DP ,DPR ,FDEP ,FDPR ,DAM ,DTB ,DTS ,FRF ,VAR)" ** N ; G4 ; R11 SafeZ ; RFP HoleTop ; clearplane ; DP HoleDepth ; drill hole depth ; Q (HoleTop - PeckDepth) ; Q1 PeckDepth ; " , 1" ; cycle dwell ; " , , 1, 0)" else if (cycle == 4) N ; " MSG (RTP ,RFP ,SDIS ,DP ,DPR ,DTB ,DTS ,)" ** N ; G4 840 ; R11 SafeZ ; RFP HoleTop ; clearplane ; DP HoleDepth ; drill hole depth ;</pre>

<pre> metric formats decimal point = true decimal places = 3 leading zeros = false trailing zeros = false end define # define word ST address letter = ", " address width = 2 field width = 5 sign = none metric formats decimal point = false decimal places = 0 leading zeros = false trailing zeros = true end define # define word ST1 address letter = ", " address width = 2 scale factor = 3 scale divisor = 2 sign = none end define # define word PF address letter = "F=R10" address width = 5 field width = 1 modal end define # define word CYC address letter = "MSG (Cycle= " address width = 11 field width = 2 tape position = 1 end define #### This last word is not normally required but may be of help in debugging options. # # vvvvv CYCLE WORD Formats vvvv # define format (G4) address letter = "MCALL CYCLE" address width = 11 end define # define format (ST1) field width = 5 </pre>	<pre> cycle dwell ; ", , " ; PT (Cycfed / ToolSpeed) ; ST ToolSpeed[ToolNum] ; ST1 ToolSpeed[ToolNum] ; ")" else if (cycle = 10) N ; " MSG (RTP ,RFP ,SDIS ,DP ,DPR ,DTB ,SDAC ,PIT, , SST , SST1)" ** N ; G4 84 ; R11 SafeZ ; RFP HoleTop ; clearplane ; DP HoleDepth ; drill hole depth ; cycle dwell ; " 3, , " ; PT (Cycfed / ToolSpeed) ; ST ToolSpeed[ToolNum] ; ST1 ToolSpeed[ToolNum] ; ")" else if (cycle = 13) N ; " MSG (RTP ,RFP ,SDIS ,DP ,DPR ,FDEP ,FDPR ,DAM ,DTB ,DTS ,FRF ,VAR)" ** N ; G4 73 ; R11 SafeZ ; RFP HoleTop ; clearplane ; DP HoleDepth ; drill hole depth ; Q (HoleTop - PeckDepth) ; Q1 PeckDepth ; ", 1" ; cycle dwell ; ", , 1, 0)" end if end define # #### These lines can be omitted ** # define block move cycle N ; x coord =C ; y coord =C end define # define block move tap N ; x coord =C ; y coord =C end define # define block cycle end N ; " MCALL" N ; rapid =C ; z coord SafZ ; coolant off end define # </pre>
--	--

```

metric formats
decimal point = false
decimal places = 0
leading zeros = false
trailing zeros = true
end define
#
define format ( R11 )
address letter = "("
address width = 1
sign = if negative
end define
#
define format ( Z2 R2 Q P )
address letter = ", "
address width = 1
end define
#
define format ( R11 DP P Q Q1 TD )
field width = 5
metric formats
decimal point = true
decimal places = 3
trailing zeros = false
imperial formats = metric formats
end define
#
word order = ( + RFP DP Q1 CYC PT ST
ST1 )
#
define keys
cycle dwell = P
end define
#
use hole top in cycles = true
integer 69 = 2
#

```

The above Siemens 840D cycles are still in proving trials and may well require adjustment for the particular machine control so be careful!

Циклы системы ЧПУ Cincinnatti Acramatic

These Cycles have been developed in conjunction with Cincinnatti Milacron in the UK but again we have had little feed back from them to say they work so use them as a basis for development.

Words, formats, keys, codes, settings	Cycle Block definitions
<pre> define format (P) address letter = "\$CYCLE_PARAMS(2)G8" address width = 19 field width = 1 end define #82 define format (null) field width = 3 metric formats decimal point = true decimal places = 1 trailing zeros = true end define #90 define format (L) address letter = "W" not modal end define #95 define format (Q) address letter = "K" end define #99 define format (O) address letter = "J" field width = 2 end define #104 define format (L Q) field width = 8 tape position = 1 metric formats decimal point = true decimal places = 3 trailing zeros = false end define #113 word order = (+ P L O null) # define keys cycle dwell = P dwell not used end define # </pre>	<pre> define block cycle start if (swa) N ; TPN ToolPathName ; ")" unset swa end if if (ToolSpeed[ToolNum] > 2000) N ; S 2000 end if if (cycle == 1) N ; G4 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; feedrate Prat ; M1 ; M2 else if (cycle == 2) N ; P cycle ; "_DWELL]=" ; null cycledwell N ; G4 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; feedrate Prat ; M1 ; M2 else if (cycle == 3) N ; P cycle ; "_DWELL]=" ; null cycledwell N ; G4 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; drill peck depth ; O 13 ; feedrate Prat ; M1 ; M2 else if (cycle == 4) N ; P cycle ; "_DWELL]=" ; null cycledwell N ; G4 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; O 2 ; feedrate Prat ; M1 ; M2 else if (cycle == 5) N ; G4 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; O 2 ; feedrate Prat ; M1 ; M2 else if (cycle == 6) N ; P cycle ; "_BOT_RET]=" ; null 0 N ; G4 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; "U-0.5V-0.5" ; O 0 ; feedrate Prat ; M1 ; M2 else if (cycle == 7 or cycle == 8 or cycle == 9) error "THESE CYCLES G87 G88 G89 NOT SUPPORTED" else if (cycle == 10) </pre>

<pre> use hole top in cycles = true integer 69 = 2 # </pre>	<pre> N ; P 4 ; ".1_DWELL]=" ; null cycledwell N ; G4 84 ; ".1" ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; O 2 ; drill peck depth ; "P1" ; feedrate Prat ; M1 ; M2 else if (cycle == 11 or cycle == 12) error "HELICAL MILLING CYCLES NOT SUPPORTED" else if (cycle == 13) N ; P 3 ; "_SHRT_RET]=" ; null (PeckDepth / 2) N ; P =C ; "_DWELL]=" ; null cycledwell N ; G4 83 ; x coord =C ; y coord =C ; drill hole depth HoleDepth ; clearplane HoleTop ; L SafeZ ; drill peck depth ; O 2 ; feedrate Prat ; M1 ; M2 end if end define #317 define block move cycle if (xcoord == OldX and ycoord == OldY) N ; else N ; x coord ; y coord ; L SafeZ end if end define #325 define block move tap if (xcoord == OldX and ycoord == OldY) N ; else N ; x coord ; y coord ; L SafeZ end if end define #333 define block cycle end N ; end of drill ; z coord end define #337 </pre>
--	---

Смещение длины инструмента

Код смещения длины инструмента выводится командой APT LOADTL в DUCT или POWERMILL. И обычно вводится в программу ЧПУ как **H**, (или иногда D) при первом перемещении после смены инструмента. Чаще всего большинством систем ЧПУ вводится с перемещением по Z с соответствующим G-кодом (например, G43 Z12. H1) (Это означает, что в конце первого перемещения по Z станок вынуждается корректно позиционироваться с компенсацией длины инструмента)

Эта настройка определяет соответствующие форматы слов и ключей (Чаще всего это уже задано в исходном файле, но лучше проверить)

```
define keys
  tool length = H
end define
```

Слово **tool length (H)**, также должно быть включено в определение *define block move rapid* как G6.

```
define block move rapid
  N ; rapid ; G6 ; x coord ; y coord ; z coord ; tool length ; spindle ; M1
end define
```

Также для получения желаемого вывода необходимы следующие флажки:

```
tool reset coordinates = 3
split move             = 0
tlo output              = true
```

при вышеупомянутых настройках произойдет следующий вывод: %

```
:0001
G91 G28 X0 Y0 Z0
G40 G17 G80 G49
G0 G90 Z49.5
T1 M6
X81. Y48.5 S1500 M3
G43 Z49.5 H1
X12.446 Y10.931
Z44.5
G1 Z44.4 F2500
```

Если изменить значение **tool reset coordinates = 3** на **= 4**, мы получим следующее:

```
T1 M6
G43 Z49.5 S1500 H1 M3
X81. Y48.5
X12.446 Y10.931
Z44.5
G1 Z44.4 F2500
```

Таким образом, регулируя значения могут быть получены разные форматы вывода (Это не всегда гарантируется, особенно в мульти-осевых опциях, здесь для получения нужного формата придется применять другие настройки)

Для вывода кода G43 требуется дополнительная настройка (хотя, это может быть определено в исходном файле)

```
define codes
  tool len off = G6 43
end define
```

и должен быть определен **G6** в *define block move rapid* как указано выше.

Вариации:

Если слово **H** требуется в строке смены инструмента, это должно быть определено в блоке инструмента с соответствующей переменной:

1/

```
define block tool change
  N ; tool number ; tool length ToolNum ; change tool
end define
```

tlo output = false ## это препятствует выводу обычной функции

получим следующее:

```
%
:0001
G91 G28 X0 Y0 Z0
G40 G17 G80 G49
G0 G90 Z49.5
T1 H1 M6
Z49.5 S1500 M3
X81. Y48.5
```

Иногда слово T используется как номер инструмента и номер смещения (например, T03.50 – обработка инструментом 3 со смещением длины 50мм). Это может быть получено используя задание смены инструмента с следующими настройками формата **H**:

2/

```
define format ( H )
  address letter = "."
  tape position = 0
end define

define block tool change
  N ; tool number ; tool length ToolLength ; change tool
end define
```

что произведет:

```
%
:0001
G91 G28 X0 Y0 Z0
G40 G17 G80 G49
G0 G90 Z49.5
T1.50 M6
```

Z49.5 S1500 M3
X81. Y48.5

Если необходимо поместить смещение инструмента перед его сменой, надо сделать следующее определения смены инструмента:

```
define block tool change  
  N ; G6 49 ; tool length 0  
  N ; tool number ; change tool  
  N ; tool length offset ; z coord FromZ ; tool length ToolNum  
end define
```

что дает:

```
G0 Z49.5  
G49 H0  
T3 M6  
G43 Z49.5 H3  
Z49.5 S1500 M3  
X12.446 Y10.931
```

Есть много других вариантов использования определения *define block move linear*, например для *мульти-осевых опций*. Приведенные примеры, по крайней мере, дают идею – как это можно сделать.

Раздел 4. Мульти-осевые опции

Настройка option-файла для 4 осей

Some 4 axes examples to illustrate the various possibilities are given here : ([4th Rotary Axis](#))

Requirements for 4th Rotary axis

The following inclusions are required to be defined in an option file for Rotary Multi-axes working.

(The three Main rotary axis definitions are being defined in this example, A , B , ? C , but only **one** will actually be used for a single rotary axis.)

```
define format ( A B C )  ## May well be defined already in the inbuilt source
  metric formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 3
  imperial formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 4
end define

word order = ( + A B C )  ## Only needed if not in inbuilt word order list

block order = true        ## Override for inbuilt word order listing, uses "define block xxx "
order

define keys
  azimuth axis    not used  ## Not required for single rotary axis
  elevation axis  = A      ## 4th rotary axis normally an elevation ( e.g. A rotates about X )
end define

## " elevation axis ", will cover A or B, or C " requires to be inserted in both Rapid, and
Linear, blocks
define block move rapid
  N ; rapid ; G2 ; G3 ; G6 ; x coord ; y coord ; z coord ; elevation axis ; spindle ; tool length ; M1 ;
M2
end define

define block move linear
  N ; linear ; G2 ; x coord ; y coord ; z coord ; elevation axis ; feedrate ; M1 ; M2
end define
```

Rotary axes parameter settings

The following parameters will need to be included in any rotary axes option.

azimuth axis parameters = (0. 0. 0. 1. 0. 0.)
elevation axis parameters = (0. 0. 0. 0. 1. 0.)

The **first three figures** in each case give the *centre of rotation* (usually 0. 0. 0. for single rotary tables).

The **second set of three figures** define the *axis of rotation* about either X, Y, or Z axis.

(It is essential that both axes are not aligned to the same axis. The **elevation** is the **main** alignment for the **4th** axis)

Thus the example shown above represents a machine where the **azimuth** rotation is set to be round the X axis (vector 1. 0. 0.), normally for 4 axes these can remain at **zero**.

The **main elevation** rotation is round the Y axis (vector 0. 1. 0.).

4th Rotary Axis is a swivel tool head unit.

The following will need to be included for this type of rotary axis.

spindle elevation rotation = **true** ## **true** = **Spindle rotation** - **false** = **Table rotary unit** (**default**)
elevation centre = (0. 0. 180.5) ## Only used for Spindle offset from *centre of rotation*, usually in the Z axis

(NOTE :- These figures must be written with points. (0. not 0)

The distance entered should be the difference between the the *centre of rotation* and the face of the tool holder. (In the example above the distance is 180.5 mm in the Z axis)

The distance between the top of the tool holder and the tool tip can then be entered using the tool length setting in DUCT, or PowerMILL.

NOTE :- Many machines with tilting tools have some form of code on the machine which will allow the tool to be run as though the *centre of rotation* were the centre of the tool tip. In this case set the centre offset to zero.

Additional parameters

elevation axis units = degrees ## Usual units (Default could be none)
elevation axis direction = positive ## Normal rotational direction required. (Default could be none)
pcs origin = (0 0 0 0 0 0) ## (Default, rarely used, mainly for 5 axes)

Azimuth Min Max Elev'n Min Max Tol. Moves
rotary axis limits = (-99999 99999 -99999 99999 0.1 1) ## Default values, virtually unlimited rotation

EXAMPLE :-

rotary axis limits = (0 0 -720.0 720.0 0.1 1) ## Elevation limited to two table revolutions

Tolerance value (Tol.) is the tolerance to which the angular displacement is maintained by Ductpost in the number of Moves set. Keep tolerance and moves to the highest and lowest values respectively ,conducive to good machine performance. (FROM DP1331 the move fuction (last digit) has been superceeded by " *linearise multiaxis moves = true* " which automatically adjusts the number of moves needed to achive the tolerance setting)

See also [4-axes examples](#)

Настройка option-файла для Стола 5-осевых опций

An example to illustrate a type of 5 axes Table is given here :- ([Rotary Axes Table](#))

Requirements for 4th and 5th Rotary axes

The following inclusions are probably required to be defined in an option file for Rotary Multi-axes working :-

(The three Main rotary axis definitions are being defined in this example, **A** , **B** , and **C** , but **only two** will actually be used for any multi-rotary axes machine.system)

```
define format ( A B C ) ## May well be defined already in the inbuilt source
  metric formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 3
  imperial formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 4
end define

#
  word order = ( + A B C ) ## Only needed if not in inbuilt word order list
#
  block order = true ## Override for word order listing, uses "define block .... " order
#
define keys
  azimuth axis  = A ## 4th rotary axis normally an azimuth ( Table tilt axis )
  elevation axis = C ## 5th rotary axis normally an elevation ( Rotary table on tilt table )
end define

## " azimuth axis and elevation axis ( A, and/or B, and/or C " requires to be inserted in both
Rapid, and Linear, blocks depending upon axis alignments.
( Using the pictorial example No.1, A tilts about X, C rotates about Z )
#
  define block move rapid
    N ; rapid ; G2 ; G3 ; G6 ; x coord ; y coord ; z coord ; azimuth axis ; elevation axis ; spindle ;
    tool length ; M1 ; M2
  end define
#
  define block move linear
    N ; linear ; G2 ; x coord ; y coord ; z coord ; azimuth axis ; elevation axis ; tool radius ;
    feedrate ; M1 ; M2
  end define
```

Rotary axes parameter settings

The following parameters will need to be included in the rotary axes option.

azimuth axis parameters = (0. 0. 0. **1.** 0. 0.)
elevation axis parameters = (0. 0. 0. 0. 0. **1.**)

The **first three** figures in each case position the **centre of rotation**, the **azimuth tilt table** usually has no offset from the centre line of the elevation axis, (therefore these will be set at **0. 0. 0.**) but the **elevation table surface centre** could be either **on**, **above**, or **below**, the **centre line of the tilt table** axis.

In the pictorial example No.1 the rotary **elevation table surface** is **below** the **azimuth tilt table axis centre line**, and for the post processor to account for this **offset**, the difference is set in the appropriate **elevation axis parameter**. The sign of this value will depend upon **where** the **Machine Centre of Rotation** has been defined by the manufacturer, and is usually on the **Centre Line of the Tilt Table Rotary axis**.

The settings for an **offset** of say, 80.56 mm. would be as follows :-

elevation axis parameter = (**0. 0. 80.56** [**positive** value for **below**, **negative** value for **above**, **zero** if surface **on** the Centre Line of azimuth rotation])

If the **Machine Centre of Rotation** is defined at the **Surface** of the **elevation table** then the **signs are reversed**.

The **second set of three** figures define the **axis of rotation** about which the table rotates, either X, Y, or Z axis.

(It is essential that both axes are not aligned to the same axis. The **azimuth** is nominally the **main** alignment)

Thus the example shown represents a machine where the **azimuth** rotation is rotating around the **X** axis (vector **1. 0. 0.**), and the **elevation** rotation is rotating around the **Z** axis. (vector **0. 0. 1.**)

(**NOTE 1 :-** These figures can be written with, or without points. (**0.**) or (**0**)

(**NOTE 2 :-** Some machine controls may have some form of **code** that will make the machine control work out it's own Rotational Transformations which allows the post processor to ignore any offset. In this case set the centre offsets to **zero**.

Additional parameters

spindle azimuth rotation = **false** ## **Default** for table unit
azimuth axis units = degrees ## Usual units (**Default** could be none)
azimuth axis direction = positive ## Normal rotational direction required. (**Default** could be none)

spindle elevation rotation = **false** ## **Default** for table unit
elevation axis units = degrees ## Usual units (**Default** could be none)
elevation axis direction = positive ## Normal rotational direction required. (**Default** could be none)

pcs origin = (0 0 0 0 0 0) ## (**Default**, rarely used, sometimes in 5 axes spindle alignments)
#

Azimuth Min Max. Elev'n Min Max. Tol. Moves
 rotary axis limits = (-99999 99999 -99999 99999 0.1 1) ## Default values, virtually unlimited rotation

#

(Example settings) :-

rotary axis limits = (-95.0 10.0 -3600.0 3600.0 0.1 4) ## Azimuth Tilt Table limited swing values, and Elevation rotary table +/- 10 revolutions)

#

linearise multiaxis moves = true ## Default setting (From DP1321 see [What's New 5-axes improvements](#))

multiaxis coordinate transform = true ## Default setting

#

Tolerance value (Tol.) is the tolerance to which the angular displacement is maintained by Ductpost within the number of Moves set, to achieve a good linearisation. Keep tolerance and moves to the **highest** and **lowest** values respectively conducive to good machine performance.

(**FROM DP1321** the move fuction (last digit) has been superceeded by " **linearise multiaxis moves = true** " which automatically adjusts the number of moves needed to achive the tolerance setting)

See also [5-axes Table examples](#)

Настройки option-файла для Головы/Стола 5-осевых опций

An example to illustrate a type of 5 axes Head/Table is given here :- ([Rotary Axes Head/Table](#))

Requirements for 4th and 5th Rotary axes

The following inclusions are probably required to be defined in an option file for Rotary Multi-axes working.

(The three Main rotary axis definitions are being defined in this example, **A** , **B** , **C** , but **only two** will actually be used for multi-rotary axes systems.)

```
define format ( A B C ) ## May well be defined already in the inbuilt source
  metric formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 3
  imperial formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 4
end define

word order = ( + A B C ) ## Only needed if not in inbuilt word order list
block order = true       ## Override for word order listing, uses "define block xxx " order

define keys
  azimuth axis    = A      ## 4th rotary axis normally an azimuth ( Spindle Rotary )
  elevation axis  = C      ## 5th rotary axis normally an elevation ( Rotary table )
end define

## " A , and/or B, and/or C " requires to be inserted in both Rapid, and Linear, blocks depending
upon axis alignments. ( Using the pictorial example, A rotates about X, C rotates about Z )

define block move rapid
  N ; G1 ; G2 ; G3 ; G6 ; X ; Y ; Z ; A ; C ; S ; H ; M1 ; M2
end define

define block move linear
  N ; G1 ; G2 ; X ; Y ; Z ; A ; C ; F ; M1 ; M2
end define
```

Rotary axes parameter settings

The following parameters will need to be included in the rotary axes option.

spindle azimuth rotation = **true** *## Default false for rotary table*

This defines that the spindle is a rotary axis head.

azimuth axis parameters = (0. 0. 0. **1.** 0. 0.)
elevation axis parameters = (0. 0. 0. 0. 0. **1.**)

The **first three figures** in this case will be set at **0. 0. 0.** for both axis, there being no likelihood of any offset in the **elevation**, and the **spindle azimuth** offset is accounted for by the **azimuth centre** parameter.

azimuth centre = (0. 0. **180.5**) **## Offset of 180.5mm in Z**

The **second set of three figures** give the axis of rotation about either X, Y, or Z axis.
(It is essential that both axes are not aligned to the same axis. The **azimuth** is the **main** alignment)

Thus the example shown represents a machine where the **azimuth** rotation is set to be about the **X** axis (vector **1. 0. 0.**), and the **elevation** rotation is about the **Z** axis. (vector 0. 0. **1.**)

(**NOTE 1 :-** These figures must be written with points. (**0.** not **0**)

(**NOTE 2 :-** Some machine controls may have a form of code that will allow the tool to be run as though the **centre of rotation** were the centre of the tool tip. In this case set the **azimuth centre offset** to **zero**.

Additional parameters

azimuth axis units = degrees **## Usual units (Default could be none)**
azimuth axis direction = positive **## Normal rotational direction required. (Default could be none)**

elevation axis units = degrees **## Usual units (Default could be none)**
elevation axis direction = positive **## Normal rotational direction required. (Default could be none)**

pcs origin = (0 0 0 0 0 0) **## (Default, used only in some 5 axes horizontal spindle alignments)**

Azimuth Min Max. Elev'n Min Max. Tol. Moves
rotary axis limits = (**-99999 99999 -99999 99999 0.1 1**) **## Default values, virtually unlimited rotation**

(**Example Settings**) :-

rotary axis limits = (**-190.0 10.0 -360.0 360.0 0.1 4**) **## Spindle Tilt Azimuth limited swing values, and Elevation Rotary Table limited to one revolution either way)**

Tolerance value (Tol.) is the tolerance to which the angular displacement is maintained by Ductpost in the number of Moves set. Keep tolerance and moves to the **highest** and **lowest** values respectively, conducive to good machine performance.

(**FROM DP1331** the move fuction (last digit) has been superceeded by " **linearise multiaxis moves = true** " which automatically adjusts the number of moves needed to achive the tolerance setting)

(**NOTE 3 :-** It is assumed that the Tool points vertically down at **A 0** for these settings.

If A 0 is either horizontal, or vertical, then a special case applies and should be refered to Delcam Support.

See also [5-axes Table examples](#)

Настройки option-файла для Шпиндельной головы 5-осевых опций

An example to illustrate a type of 5 axes Spindle Head is given here :- ([Rotary Axes Head](#))

Requirements for 4th / 5th Rotary axes

The following inclusions are probably required to be defined in an option file for Rotary Multi-axes working.

(The three Main rotary axis definitions are being defined in this example, **A** , **B**, **C**, but **only two** will actually be used for any multi-rotary axes system.)

```
define format ( A B C )  ## May well be defined already in the inbuilt source
  metric formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 3
  imperial formats
    leading zeros      = false
    trailing zeros     = true
    decimal point      = true
    decimal places     = 4
end define

word order = ( + A B C )  ## Only needed if not in inbuilt word order list
block order = true        ## Override for word order listing, uses "define block xxx " order

define keys
  azimuth axis    = C      ## 4th rotary axis normally an azimuth ( Column Rotary )
  elevation axis  = B      ## 5th rotary axis normally an elevation ( Spindle Rotary )
end define

## " A , and/or B, and/or C " requires to be inserted in both Rapid, and Linear, blocks
depending upon axis alignments. ( Using the pictorial example, B rotates about Y, C rotates about Z )

define block move rapid
  N ; G1 ; G2 ; G3 ; G6 ; X ; Y ; Z ; B ; C ; S ; H ; M1 ; M2
end define

define block move linear
  N ; G1 ; G2 ; X ; Y ; Z ; B ; C ; F ; M1 ; M2
end define
```

Rotary axes parameter settings

The following parameters will need to be included in the rotary axes option.

```
spindle azimuth rotation    = true  ## Default false for rotary table
spindle elevation rotation = true  ## Default false for rotary table
```

These define that the column / spindle is a rotary axis.

azimuth axis parameters = (0 0 0 0 0 **1**)
elevation axis parameters = (0 0 0 0 **1** 0)

The first three figures in this case will be set at **0 0 0** for **both** axis, and are normally reserved for table units.

Whereas **spindle azimuth** and **elevation offsets** are accounted for by the **azimuth** and **elevation centre** parameters.

azimuth centre = (0. 0. 0.) *## Unusual to have an Offset in the azimuth)*
elevation centre = (0. 0. **180.5**) *## Offset of 180.5mm in Z*

The second set of three figures give the **axis of rotation** about either X, Y, or Z axis.
 (It is essential that both axes are not aligned to the same axis. The **azimuth** is the **main** alignment)

Thus the example shown represents a machine where the **azimuth** rotation is set to be round the **Z** axis (vector 0. 0. **1.**), and the **elevation** rotation is round the **Y** axis. (vector 0. **1.** 0.)

(**NOTE 1 :-** These figures can be written with, or without, points. (**0.**) or (**0**)

(**NOTE 2 :-** Some machine controls may have a form of code that will allow the tool to be run as though the **centre of rotation** were the centre of the tool tip. In this case set the **elevation centre** offset to **zero**.

Additional parameters

azimuth axis units = degrees *## Usual units (Default could be none)*
azimuth axis direction = positive *## Normal rotational direction required. (Default could be none)*
elevation axis units = degrees *## Usual units (Default could be none)*
elevation axis direction = positive *## Normal rotational direction required. (Default could be none)*

pcs origin = (0 0 0 0 0 0) *## (Default, used only in some 5 axes horizontal spindle alignments)*

Azimuth Min Max. Elev'n Min Max. Tol. Moves
 rotary axis limits = (-99999 99999 -99999 99999 0.1 1) *## Default values, virtually unlimited rotation*

(EXAMPLE SETTINGS)

rotary axis limits = (**-360.0 360.0 -110.0 110.0** 0.1 4) *## Spindle Azimuth Column rotation limited to one revolution either way, and Spindle Tilt Elevation rotation limited to +/- 110 degs. either side of zero [Vertical])*

Tolerance value (Tol.) is the tolerance to which the angular displacement is maintained by Ductpost within the number of Moves set. Keep tolerance and moves to the **highest** and **lowest** values respectively, conducive to good machine performance.

(**FROM DP1331** the move fuction (last digit) has been superceeded by " **linearise multiaxis moves = true** " which automatically adjusts the number of moves needed to achive the tolerance setting)

(**NOTE 3 :-** It is assumed that the Column is vertical and the Tool points **vertically** down at **C 0 , B 0** , or the Column is horizontal and the Tool points **horizontally** at **C 0 , B 0** for these settings.

If the Column is horizontal and B 0 is vertically UP or DOWN, then a special case applies and should be refered to Delcam Support.) See also [5-axes Spindle examples](#)

Раздел 5. Механическая обработка Сплайна

Механическая обработка Сплайна

Many of the latest machine tool control units have the capability of spline interpolation. This means that rather than taking as input a series of small straight line moves, the control unit will accept a mathematical definition for a curve to control the movement of the machine tool.

However, most 3D CAM systems are only capable of producing straight line vectors, even when the shapes they are machining are designed with a type of **spline** curve.

What is spline machining?

Spline machining is using spline curves to define the path a machine tool should take.

This has two major advantages.

- 1/ The amount of data that is needed to be passed to the controller can be dramatically reduced.
- 2/ On some controllers with higher feed rates being used, the machine can traverse the spline tool path faster than a conventional one.

When people talk about spline tool paths, there are three main forms of them.

NURBS tool paths, (**N**on **U**niform **R**ational **B**Splines)
POLYNOMIAL tool paths, (Two types :- **Polynomial** parametric curves
and :- **Hermite** polynomials.)

All three curve types have their differing advantages, and disadvantages, so there is no one standard. All can be made to work in similar ways.

With the splining method developed by Delcam, we offer tangent continuity between the curve spans when it is available.

What you will need :-

SplineMILL:- This is the program that takes the normal CLdata cut file, and splines the result. These splines are then Post Processed using a special (or modified) Option file.

DuctPost ****:- This is a Post Processor that can interpret the Splined 5000 cut file record and change it into the format to produce the tape file that is needed by the controller.

The **** is the Post Processor version that you will need and will change with time but should be **1203, or later**.

DUCTPost and SplineMILL will accept cut files produced by both DUCT, and PowerMILL.

However, cut files produced on a PC in native NT format, are not portable to a UNIX system and will therefore have to be processed on an NT system.

Which controllers are supported:-

Siemens

Controls	Sinumerik 810D and 840D
Output type	Parametric curves (Version 1) Nurbs (Version 2)
Tested	Bridgeport - with Siemens approval
Basic Option File	spl_s840.opt
Shipped	PowerMILL 2475
Post Processor	DUCTPost 1203 (or later)

Fanuc

Controls	15m and 16m controllers. Needs the 64 bit Risc option
Output type	Nurbs curves
Tested	Fanuc 16m Sidel of France.
Option File	spl_f16m.opt
Shipped	PowerMILL 2475
Post Processor	DUCTPost 1205 (or later)

Heidenhain can be supported but we have not found any material advantages over the conventional Heidenhain output.

Okuma spline can be provided but has not had extensive testing and therefore cannot be guaranteed to be effective.

There are no plans to support any other controls at the moment.

For Licence and operational details refer to :- [Spline Licence ? Post Processor details](#)

[Splining Tolerances](#)

As more customers become interested in what splining has to offer, it is becoming apparent that the use of tolerances needs explaining.

When Delcam did its original development work it was realised that the tolerances set during tool path generation could have a drastic effect on surface finish.

With the release of **SplineMill 1101** it was recognized that this could cause problems and the following was added to the documentation :-

[A note about tolerances](#)

In order for the splining process to be effective, it must usually make some approximations to the input data. The splining process is controlled by a tolerance value which is input to the program, and the fitted spline is guaranteed to be accurate to within the given tolerance at the co-ordinate positions given in the input file.

A second tolerance is used internally within SplineMILL to control the deviation of the fitted spline from the straight lines which join together the co-ordinate positions in the input file. This tolerance is initially set to the same tolerance which is used for the positions themselves, but may be increased automatically if it cannot be achieved. A warning will be printed if this is done, but this is only likely to occur where the tolerance used to generate the CLdata in the CAM system, is larger than the fitting tolerance input to SplineMill.

Since the generation of the original tool path is itself an approximation to the underlying geometry of the surface being produced, great care should be taken to avoid the build up of error which could affect the accuracy of the finished component. In order to achieve a given tolerance it is recommended that half the required value is used in the CLdata generation process within the CAM system, and the same value is also used in the splining process, in this way the overall tolerance should be as required.

Should the program warn that it can not maintain the surface tolerance and automatically increase the value, the new value should be taken into consideration when accessing the overall accuracy of the Splined tool path. If the result is unacceptable the input file should be recalculated using a finer tolerance value and the splining process repeated.

Whenever this program is used it is advisable to check the resulting CLdata file to make sure it is acceptable and has not produced splines which deviate significantly from the original tool path.

A Rule of Thumb for Tolerances

Delcam development have made various tests, and have come up with a rule of thumb for the relationship between the various tolerances. This is simply an empirical rule, but has been found to work well in a wide variety of circumstances.

TOL = SPL	SPL = Splining tolerance
M/C = 1/2 TOL	M/C = Machining tol. in PowerMill
TRI = 1/4 TOL	TRI = Triangulation tolerance

Why are these recommended?

These guidelines were developed during the machining trials of [SplineMill 3001](#), and the reason for the factor being set to 2 is as follows.

In PowerMill you are generating a tool path over an approximation of the data, (i.e. A triangle set). If the value of the machining tol. is close to the triangular tolerance, a faceted surface will result. (This will show up as surface effects of the triangle boundaries)

If the machining tolerance is too high, (i.e. close to the splining tolerance) the splines have no room to move, and the amount of data reduction is restricted . This gives a longer tape and lessens the benefits that splining can provide.

The factor of 2 gave the best results and was the simplest description method. This does not mean that the values can not be altered, but you must be aware of the adverse effects that can result.

For example, if you wanted to achieve a tolerance of 0.01

$$\text{TOL} = 0.01$$

$$\text{SPL} = 0.01$$

$$\text{M/C} = 0.005$$

$$\text{TRI} = 0.0025$$

This allows the spline maximum movement and to remain inside the tolerance. As a result the data is reduced.

The reasoning behind the last statement should probably be explained further ... This assumes that the linear tool path will under cut a convex shape, and leave additional material on a concave shape. In both cases the effect of splining will be to reverse the linearisation effect.

In many examples we have reduced the data to a sixth, or better, than its original linear ISO tape size.

Лицензия SplineMill Licence и Useage Details

Splinepost has now been integrated into Ductpost from version Ductpost1203.

Ductpost1203 was supplied with PowerMill2475, and has recently been superseded by **Ductpost1205** which is now the recommended version (and any later versions) to use with **SplineMill1101**. As development proceeds further, up-dates will be inevitable and will usually be incorporated with the latest release of PowerMILL CDs.

SplineMill is a separate utility that is provided on the PowerMill CD.

To run spline machining, you need the following options in your license file:

To run **SplineMill** : SPLINEMILL--SPLINEMILL

To run **Ductpost** with spline machining, you need:

DUCTPOST--SPLINE DUCTPOST--DUCTPOST

You will also need a machine licence, and the only **two proven** available option files for spline machining at the present moment are :-

spl_f16m for which you need the machine licence : DUCTPOST--FANUC11M

spl_s840 for which you need the machine licence : DUCTPOST--SIEM850

These are the base configuration files and will run as stand alone, but normally will be incorporated within a Customer's Option file.

An **Okuma** Spline option has been developed but at the present moment we do not have sound machine testing to be assured that it is viable.

Useage

To generate a spline machining tape file, it is first necessary to output a cutfile as normal from PowerMILL.

e.g. :- **normal.cut**

Then use the product " **splinemill** ".

You will be prompted for the input cut file name :- **normal.cut**

Then for an output cut file name e.g. :- **normal-spl.cut**

Followed by a request for a tolerance for splining :- e.g **0.01 (say) { This will not be asked for in later versions }**

The output cutfile will be the **splined** version of the original cutfile.

A sensible approach to naming cut / tape files that have been **Spline** processed is to use a naming convention similar to the above **[filename-spl]** (as used by Delcam) to signify files which have been splined.

The next step is to use the appropriate spline option with ductpost to process the splined cutfile,

e.g. :- `ductpost spl_f16m.opt * normal-spl.cut`

which will produce a spline tape file :- `normal-spl.tap`

The tape file produced will after the first few lines start to `spline`. The normal " **X,Y,Z** " output will not be evident.

** **Note** :This is the stand alone option, if it has been incorporated in a customised option then use that instead.*

For further information on **Splinemilling** refer to :- [Spline Machining](#)

Option-файл Сплайна

This document is an overview of the necessary steps to writing an option file that allows spline machining. For a more detailed explanation of spline machining itself see the introduction to [Spline Machining](#), and also [Using Spline Machining](#) guide.

Spline types

DUCTpost supports two types of spline representations : **polynomial** representation, and **B-spline** representation.

The choice is determined by what is supported for a given machine tool, and the representation effects the way the option file is written, since each makes use of different data provided by DUCTpost.

To choose **polynomial** representation for the spline data, use the flag:

spline type = polynomial

To use **B-spline** representation for the spline data, use the flag:

spline type = bspline

Code "spline"

The code " *spline* " is used in the same ways as other codes, (for example the code " *linear* "). It is usually used at the start of a spline move block, to signify that spline mode has been entered.

```
define codes
  spline = G1 "6.2" ( For B-spline )
end define
```

Polynomial representation

This type of representation presents the spline data as polynomials in each of X, Y and Z co-ordinates in terms of a parameter form. The machine tool then requires the additional coefficients of the polynomials and these are accessed via certain numbered keys :-

```
define keys
  key 58 = K0X
  key 59 = K1X
  key 60 = K2X
  key 61 = K3X
#
  key 62 = K0Y
  key 63 = K1Y
  key 64 = K2Y
  key 65 = K3Y
#
  key 66 = K0Z
  key 67 = K1Z
  key 68 = K2Z
  key 69 = K3Z
end define
```

K0X will refer to the **constant** term in the **X polynomial**
K1X will refer to the **linear** term in the **X polynomial**
K2X will refer to the **quadratic** term in the **X polynomial**
K3X will refer to the **cubic** term in the **X polynomial**
Similarly for the **Y and Z polynomials**.

It is necessary to refer to the machine tool documentation to see precisely how these coefficients are to be presented to the machine. (For example, some machine tools may not require the **linear** coefficient, since this can be determined by other means).

Sample " **define block move spline** " for " **spline type = polynomial** "

```
define block move spline  
  N ; G5 ; G7 ; G1 "POLY " ;  
  if ( swx )  
    K0X xcoord ; K2X ; K3X ; " ) " ;  
  end if  
  if ( swy )  
    K0Y ycoord ; K2Y ; K3Y ; " ) " ;  
  end if  
  if ( swz )  
    K0Z zcoord ; K2Z ; K3Z ; " ) " ;  
  end if  
  F  
end define
```

The movement switches, " **swx** " etc., are used to ensure that the **polynomial** for a particular coordinate is only output if necessary.

B-spline representation

A full discussion of the mathematics of the **B-spline** representation is beyond the scope of this document. For our purposes it suffices to say that a **B-spline** is defined by a sequence of control points, and also by a **knot vector**.

The **knot vector** is accessed through a particular numbered key :-

```
define keys  
  key 70 = KNOT  
end define
```

The control points are simply considered as any other points, and are accessed through X, Y and Z.

Sample " **define block move spline** " for " **spline type = bspline** "

```
define block move spline  
  N ; G1 ; KNOT ; X ; Y ; Z  
end define
```

The above example will output each control point of the **B-spline** on a separate line, together with an associated **KNOT** value.

Раздел 6. Список данных

Список Систем ЧПУ

Ductpost	Производитель (Базовые системы)
ab84	Allen Bradley
acra8	Acramatic 850 (Cincinnati Milacron)
agie	Agie wire spark
anilam	Anilam Crusader
boss	Bridgeport (Series 1, Series 2)
bosch	Bosch 200
bostom	Bostomatic
charm	Charmille Robofil wire spark
deckel3	Deckel Dialogue 3
deckel4	Deckel Dialogue 4
deckel11	Deckel Dialogue 11
dyna	Dynapath
dm	Dynamite
eberle	Eberle
elexa	Elexa
fadal	Fadal
fanuc	Fanuc (General)
fanucom	Fanuc OM
fanuc6m	Fanuc 6M
<i>fanuc10m</i>	Fanuc 10M (теперь не поддерживается)
fanuc11m	Fanuc 11M
fanuc15m	Fanuc 15M
fanucw	Fanuc wire spark
<i>fanuc6t</i>	Fanuc6t lathe (теперь не поддерживается)
<i>fanuc10t</i>	Fanuc10t lathe (теперь не поддерживается)
fagor	Fagor
fidia	Fidia
ge2000m	GE2000 General Electrics
<i>hauser</i>	Hauser 311,314 (теперь не поддерживается)
heid	Heidenhain 150,355
h33	Heidenhain H33 (Только для России)
h155	Heidenhain 155
heid400	Heidenhain 400 series

heidiso	Heidenhain Iso
hurco	Hurco
<i>incon</i>	Incon (теперь не поддерживается)
<i>japtw</i>	Japt wire spark (теперь не поддерживается)
kryle	Kryle
maho	Maho(Philips 432, 532, and DMG MillPlus controls)
mazak	Yamazaki Machine Tools (Mazak)
matsura	Matsura
mitsu	Mitsubishi
mitsuw	Mitsubishi wire spark
num	Num760F
okuma	Okuma
<i>p28m</i>	Posidata 2800 (теперь не поддерживается)
ph432	Philips 432
r2e3	Bridgeport R2E3
roland	Roland
siem850	Siemens 800,810,850
selca	Selca
<i>servo</i>	Servo (теперь не поддерживается)
sodick	Sodick wire spark
standard	<i>Base Post Processor</i>
<i>tcomp</i>	Tcomp (теперь не поддерживается)
tiger	Tiger

Жирным шрифтом обозначены системы ЧПУ, которые в настоящее время включены в Ductpost, и поддерживаются им.

Постпроцессоры для более поздних версий и других систем ЧПУ могут быть разработаны на основе имеющихся ***option-файлов***.

Определение блоков

A block definition consists of one or more lines which can appear in the NC program output in response to a particular part of the CLDATA input.

These take the format :-

define block

end define

(e.g. **move linear**, or **tool change**).

The blocks which may be defined are as follows, and are listed in their logical order of use :-

define block	Function
tape start	Output standard data at the beginning of each nc program.
tool change first	Этот блок определяет правила вывода первой смены инструмента.
tool change	This block is output on the second and subsequent tool changes if the tool change first block is defined. It is output for all tool changes otherwise.
tool change clear	If this block is defined the stored values for all modal output words are cleared, and then the block is output with the current values.
move from	Output for each of the XYZ co-ordinate triplets in a record corresponding to the from (datum) point.
move rapid	Output for each of the XYZ co-ordinate triplets in a record corresponding to a rapid move.
datum shift	This handles any 3+2 datumshift/workplane changes and data related to them.
move linear	Output for each of the XYZ co-ordinate triplets in a record corresponding to a linear move.
multiaxis transition	This will handle the required output data for Intermediate Toolpath moves as specified in the block.
move spline	Output for a spline move. The output depends on the " spline type " and is either a sequence of piecewise-continuous polynomials, given in coefficient form, or a B-spline representation. The " move spline " block is used to output either each individual polynomial (in terms of its coefficients), or alternatively, one line of the standard B-spline representation.
move circle	Output for each of the XYZ co-ordinate triplets , and either I J K or R radius co-ordinates in a record corresponding to a circle output.
cycle start	Output at the beginning of each canned cycle.
move cycle	Output for each of the XYZ co-ordinate triplets in a record during a canned drill / tapping or boring cycle.
move tap	Output for each of the XYZ co-ordinate triplets in a record during a canned tapping cycle.
cycle end	Output at the end of each canned cycle.

tape split start	Output at the beginning of each file when the nc program is split into several files.
tape split move	Define First moves on consecutive split tape(s). (e.g. Old coordinates followed by Z move, then XY move)
tape split end	Output at the end of each split program file.
segment	Insert into tape file included data when prescribed limit of distance traveled by tool, or number of line blocks reached. (set by " segment type "
go home preamble	Output at the start of the go home sequence. This block is always output, even if the go home output flag is false .
go home z move	Output during a go home move if the Z coordinate value has changed .
go home xy move	Output during a go home move if the X and Y coordinate values have changed .
tape end	Output standard data at the end of each nc program.

Other blocks that can be defined for specific requirements :-

define block *****	Function
cldat (xxxx)	Output included data, or set action, when cldata record called by CLDATA. (xxxx is the appropriate cldat number – brackets not required)
ppfun (abcd)	Output included data, or set action, when ppfun record called by CLDATA. (abcd is the appropriate ppfun name relating to the one in the PM Command Setting – brackets not required)
user (efgh)	A user defined block can be set up to encompass standard data that is used several times in the option file and can prevent mistakes in missed edits. “ define block user efgh ” Note:- “ efgh ” is case sensitive “ eFgH ” is not the same To call the block use “ call block efgh ” in a defined block where “ efgh ” or “ eFgH ” is the defined name of the user block.

If the output of a default defined block is not required, then in the option re-define as follows :-

```

define block .....
    N ;
end define
or as
define block .....
end define

```

Переменные используемые в определении блоков

An NC tape block/line definition consists of one, or more, lines of data that appear in response to a particular part of a Postprocessed Cut file, or NC program. (e.g. a linear move or a tool change, etc., such as “ **N0 G71** ”).

The words, and values, which appear in these block/lines can be defined in the option file by utelising defined Words and associated Variables within the various block forms :-

```
define block tape start
  N BlockNumber ; G4 TapeUnits
  .....
end define
```

The **variables** that may be used in block definitions with Words are given below.

(e.g. **T NextTool** gives the Word **T** the value of the **next load tool number**. They may be written in upper or lower case, or **Preferably As Indicated**)

Variable Name	Use and Function
Angle	The defined output angle
Argument	The argument from an PPFUN record in PowerMILL NCPrograms/Preferences/Commands Dialog box Use :- FEEDRATE 3000 <pre>define block ppfun feedrate N ; "FN 0 Q3= " ; TN Prat (150) N ; "FN 0 Q4= " ; TN Frat (1200) N ; "FN 0 Q5= " ; TN Argument (3000) end define</pre> N20 FN 0 Q3=150 N25 FN 0 Q4=1200 N30 FN 0 Q5=3000 (The argument can be used in other blocks with care but only ONE argument is valid in the ppfun block. To use elsewhere in the option use Word{TN}, relating to TN in the above example can be utelised)
Azmove	The difference between the current and previous azimuth angle.
Azimuth	The Azimuth angle
Azpos	The current value of the azimuth angle.
BlockMaxX BlockMaxY BlockMaxZ	The Maximum values of X, Y and Z of the Block Size , usually used in comments at tape start.
BlockMinX BlockMinY BlockMinZ	The Minimum values of X, Y and Z of the Block Size , usually used in comments at tape start.
BlockNumber	The actual block number at that point of output

Variable List Continued (2)

Variable Name	Use and Function
Character(n)	Depending on the value , outputs an ASCII character. (e.g. character 8 = ";" will place this string at the end of every line)
CheckCounter	This will return the current value of the counter without incrementing its value.
ClearPlane	Drilling/Tapping clear plane above the work surface (Absolute position)
ClearPlaneInc	Drilling/Tapping clear plane above the work surface (Incremental position)
CompensationToolLength	The Cutter Compensation Length as defined in the PowerMill NC Preferences Toolpath dialog box
CoolantCode	M code value for coolant setting (e.g. 8 if coolant on)
ContactNormalX ContactNormalY ContactNormalZ	The current tool path Contact Normal Vector
Counter	This will be incremented each time it is used. The start value and increment can be set, using counter start and counter increment
Cycfed	Canned cycle Tapping feed rate
Cycle	Number of current drilling/tapping cycles (e.g. drill cycle = 1)
CycleDwellTime	The current cycle dwell time
CycleSafeZ	Returns the previous behavioral SafeZ in drill cycles.
Day	Day of the week (numerical e.g. 01 for first day of the Month)
Distazi	The Azimuth angle feedrate code for MillPlus multiaxis controls
DisteLe	The Elevation angle feed rate code for MillPlus multiaxis controls
DPversion	Provides the version of Ductpost used to produce the NC tape output. (Use in conjunction with OptionFileName)
Elmove	The difference between the current and previous elevation angle.
Elpos	The current elevation angle position
FeedRate	The current feed rate
Frat	The current cutting feed rate
FromX ; FromY ; FromZ	The X , Y , Z co-ordinates of the Datum (FROM) point
GotContactNormals	The variable to check if the tool path has Contact Normal Vectors enabled
HoleDepth	Drilling/Tapping hole depth (Absolute bottom position from clear plane)
HoleDepthIncremental	Drilling/Tapping hole depth (Incremental distance from clear plane to bottom of hole)
HoleDiameter	This provides the actual Hole Diameter - not the Drill diameter
HoleTop	Provides the absolute position of the top of the hole .
Hours	Time of Day Hour
inta, intb, intc -> intf	Integers which can be set with an integer value (e.g. edit inta 25)

JobName	Cut or NC Tape file name
---------	---------------------------------

Variable List Continued (3)

Variable Name	Use and Function
LastTool	Last Load Tool number (Last tool in the series to be used)
LoopCounter	Returns the number of passes in the " repeat " loop function
MaxX ; MaxY ; MaxZ	The overall Maximum limit values of travel for X, Y and Z (usually used in comments at tape start, or for DUCT block size)
MinX : MinY : MinZ	The overall Minimum limit values of travel for X, Y and Z (usually used in comments at tape start, or for DUCT block size)
Minutes	Time of Day Minutes
MinorWord	Values from CLdata associated with a particular function (Rarely used now)
ModelX : ModelY : ModelZ	Give access to the coordinates in the model system in all circumstances. These were introduced for a specific operational debug and as such will not normally be of general use
Month	The current month (numerical e.g. 04 for April)
MoveComponentType	<i>This is still under development</i>
MoveMinorType	<i>This is still under development</i>
MoveType	This checks the type of tool path movement (MoveType = 11 connection moves ; MoveType = 10 5axis moves) For full list see What's New
NextAngle	The next angle defined
NextTool	The tool number of the next load tool to be used (Used for preselection)
NextToolPath	The next tool path to the current one.
NumberPecks	Number of pecks in a drilling cycle
NumberToolPaths	Returns the number of toolpaths in the CLdata file (used in " repeat " function)
NumberTools	Returns the number of tools in the CLdata file (used in " repeat " function)
OldX ; OldY ; OldZ ; OldAz ; OldEl	The last positional axis move
OptionFileName	This should be used in conjunction with DPversion as a verification that the correct Option has been used to produce the NC output.
OutputWorkPlaneName	The PowerMill NC program Output WorkPlane
PartID	A part identification string. This may be obtained from the DUCT PARTNO cldata record, PowerMILL Part Name , the configuration file, or prompted for at run time (<i>Except prompts are ignored if posted via PowerMILL, or will cause PowerMill to hang</i>)
PartID[5]	This is similar to above, but has the added flexibility to select the starting point in the Part Identification string via the [value]. In the example the 5th character in the Part Name will be the start point, and the number of characters output will be limited by the field width of the carrier word. (ID PartID[5])
PeckDepth	Drilling cycle peck depth (Incremental)

Pitch	Returns the Pitch setting in PowerMill Tap Cycles
Plunging	Check for Plunge move.

Variable List Continued (4)

Variable Name	Use and Function
PMCodeBase	Provides the first 5 digits of the PowerMill Codebase number
PMCodeBaseRevision	Provides the 2 digit PowerMill Codebase <u>Revision</u> number
Prat	The current plunge feed rate
PrevTool	The tool number of the previous tool used
PrevToolPath	The previous tool path to the current one.
ProgID	A program ID number . (This defaults to " 1 ", or can be pre-set in the option by " program id start ", or prompted for at run time. (<i>Except prompts are ignored if posted via PowerMILL-see PartID</i>)
Programmer	The current owner of the programme
Reconfiguring	Checks to see if a rotary axis reconfiguration is happening.
RepeatedLoadTL	Check to verify if same tool is being used in current tool path as previous tool path.
RFrat	The maximum rapid feed rate setting (rapid feedrate = 30000.0)
rna, rnb -> rnf	Defined Real Numbers (edit rna = 1.234)
SafeZ (SafZ)	A safe Z positional hieght above the job
Seconds	Time of Day Seconds
SpecialRecordCode	Returns the Integer Code used to identify the particular special record in question. (see DP1460 <u>A new “ define block special record ” has been added</u>)
SpecialRecordCode[n]	Returns the “ Real ” value defined in the particular special record.
SpecialRecordString[n]	Returns the “ String ” value defined in the particular special record.
SpindleCode	The current M code for spindle state (e.g. 3 if spindle ON)
SpindleSpeed	The current spindle speed
StartZ (StaZ)	The Z position at which Plunge feed rate starts
Srat	The rapid skim feed rate as defined in PowerMill Feed Rate Form " Rapid "(Not applicable to DUCT)
swa, swb , swc -> swl	Switches which may be set and tested for in logic " if (statements) " (<i>swx, swy, swz are used solely for Spline applications</i>)
TapeCoords	The function code for the current co-ordinate system (usually 90 for absolute or 91 for incremental)
TapeUnits	The function code for the current tape units (21 or 71 for metric or 20 or 70 for imperial except for Heidenhain)
Thickness	The thickness of material left on a defined cutter path
TipCoordinates	The Tool Path output coordinates as set for Tool Tip (true) or Tool Center (false)

TipRadius[ToolNum]	The Tip radius of current tool
Tolerance	The current tool path tolerance setting
ToolCompensation	The current compensation G code (e.g. 41, 42 or 40)
ToolCount	The current tool sequence number as aposed to given tool number
ToolFeed	The current feed rate for tool
ToolLength[ToolNum]	The current created tool length

Variable List Continued (5)

Variable Name	Use and Function
ToolLenOffset	The Tool length offset (Usually tool number)
ToolName	The Tool Name from PowerMill 2.5 Tool Form or Tool Identifier from PM3.0 (Usually a reference Number, or Name)
ToolNumber	The current tool number
ToolNumber[3]	The Tool number of the 3rd in the list of tools programmed
ToolPath3axis	Checks for 3 axes type of tool path
ToolPath3Plus2	Checks for 3+2 type of tool path
ToolPath5axis	Checks for 5 axes type of tool path
ToolPathFromTool	Maps toolnumbers to a toolpath number (used in “ repeat ” function)
ToolPathLength	The length of tool path cutting metal in Output Units .
ToolpathMaxX , Y , Z	The current tool path Maximum travel distance.
ToolpathMinX , Y , Z	The current tool path Minimum travel distance
ToolPathName	The current tool path name .
ToolPathTime	The time of tool path cutting metal in Seconds
ToolpathWorkplaneName	The WorkPlane Name used to generate the tool path
ToolPathWorkplaneA , B , C	Povide access to the Euler angles for a Toolpath Workplane
ToolPathWorkplaneX , Y , Z	Povide access to the Shift position for a Toolpath Workplane
ToolRadius[ToolNum]	The current tool radius
ToolRadOff	The Tool radius offset (Usually G43 coupled with H tool number)
ToolSpeed[ToolNum]	The current spindle speed for tool
ToolType	The defined PowerMill Tool definition (e.g. EndMill)
ToolVectorX ToolVectorY ToolVectorZ	The current linearised Tool Vector
UsingHoleTop	Check to see if Hole Top coordinates is defined for drilling cycles.
Word[n] Word{ N }	The "value" of the [n] word in the list of defined words. (e.g. Word[2] is the current value of the N word) This is new to DP1361 and serves the same function as above but uses the Word itself. (NOTE: curly brackets are used)
WordString[n] WordString{ TPN }	This is new to DP1361 and is similar to the above but will return the "STRING" content of the word.

WorkPlaneA, B, C	The current 3+2 tool path Work Plane Coordinates (Used with setting “ workplane angles = apparent ”)
WorkPlaneX, Y, Z	The current 3+2 tool path Datum Shift Work Plane Coordinates (Used with setting “ workplane angles = apparent ”)
WorkPlaneAz WorkPlaneEl	The current 3+2 tool path Work Plane Angles (Used with setting “ workplane angles = machine tool ”)
WorldFromX, Y, Z	The current 3+2 tool path Work Plane From World Coordinates

Variable List Continued (6)

Variable Name	Use and Function
Xcentre ; Ycentre ; Zcentre	The stored X, Y, Z co-ordinates
Xcoord ; Ycoord ; Zcoord	The current X, Y, Z co-ordinates
Xmove ; Ymove ; Zmove	The distance moved between current and last position of X ,Y, Z
Year	The current year (numerical e.g. 01 for 2001)

Пояснение Строя / Целочисленных переключателей

The following is the basic structure of the arrays, and the function of the switches.

Circles

● integer 26 20 27 28
Circle output = (n n n n)
! ! ! !
! ! ! 0 = Absolute Centre position
! ! ! 1 = Incremental Centre from start point
! ! !
! ! 0 = Single Quadrant
! ! 1 = Full Circular Interpolation
! !
! 0 = X,Y,Z coordinates not required
! 1 = X,Y,Z coordinates required
!
0 = NO Circular Output
1 = Circular Output

Spindle

● integer 10 11 12 13
Spindle output = (n n n n)
! ! ! !
! ! ! 0 = Don't link Coolant to spindle
! ! ! 1 = Link Coolant to spindle (i.e. M13)
! ! !
! ! 0 = No M code output
! ! 1 = M code output for spindle (i.e. M03)
! !
! 0 = No S output for spindle OFF
! 1 = Output S0 for spindle OFF
!
0 = Output spindle data when called by CLdata
1 = Output spindle data in next block
2 = Output spindle data with next move

Note :- It is possible that the " define keys ", spindle = S, or spindle not used, may have to be changed to get a particular output required. **Experiment !**

Coolant

● integer 6 7
Coolant output = (n n)
! !
! 0 = Output cool't OFF when called by CLdata
! 1 = Output cool't OFF in next block
! 2 = Output cool't OFF with next move
!
0 = Output cool't ON when called by CLdata
1 = Output cool't ON in next block

2 = Output cool't ON with next move

Radial Compensation (Applicable for PowerMill 3.0 and Ductpost 1331 onwards see [Radial Compensation](#))

integer 34 35 36 37
Cutter compensation output = (n n n n)
! ! ! !
! ! ! 0 = No cutter compensation output
! ! ! 1 = Output G41/G42 and G40 codes
! ! !
! ! 0 = Obsolete, no longer needed
! !
! 0 = No radius offset (Usually D)
! 1 = Output radius offset (D ToolNumber)
!
0 = Obsolete, no longer needed

Note :- A minimum linear block must be defined as follows :-

define block move linear

N ; linear ; G2 ; tool radius ; x coord ; y coord ; z coord ; feedrate ; M1 ; M2

end define

Movement following Tool Change

integer 16
Tool reset coordinates = (n)
!
0 = No special output
1 = Force out X, Y and Z after tool change
2 = Force out X and Y after tool change
3 = Force out X and Y, then Z after tool change
4 = Force out Z then X and Y after tool change

Split Moves

integer 24
split move = (n)
!
0 = NO special output for X Y Z moves (Best for 5 axes)
1 = Output X Y move before Z for downward moves
Output Z move before X Y moves for upward moves

A note on splitting moves:- Be extremely careful of using split move for Multi-Axes options as it does not work in the same way as for Three Axes. It is a little too complex to explain in this brief note, so if you have problems refer back to Delcam Support for assistance.

Also ensure that a linear block has moves defined, some inbuilt controls haven't.

Canned Cycles (With the latest drilling in PowerMill ignore these, leave as the default except for integer 69)

integer 19
Cycle output = (n)
!
0 = Output canned cycles (G81 etc.)
1 = NO Output of canned cycles

(Peck depth output)

integer 4 = (n)

!

0 = Output as required in cycle

2 = Output even if only ONE peck in all cycles

(Tapping Feedrate reduction)

integer 69 = (n)

!

0 = Reduce normal feedrate by approx 8%

2 = Apply normal plunge feedrate (with PM3 and later Drilling always use

this)

(Clearance Plane)

integer 54 = (n)

!

0 = Absolute position value output

1 = Incremental move value output

(Drill Depth)

integer 55 = (n)

!

0 = Absolute value output

1 = Incremental output

See also [Ductpost Integer list](#)

Список целочисленных DuctPost

Always use the name format indicated in blue = n unless none available.

Symbols :- *preferred use format*: (**absolute**) indicates usual default : (**false**) indicates usual alternative

Those marked + see [Array Data](#) Also [Integers, Reals / Characters](#)

integer 1 not used

message output = true or false

integer 2 = 0 NO messages output
= 1 Messages output

integer 3 = 0 Multi-axes dissabled
= 1 Multi-axes enabled

integer 4 = 0 Output **peck depth** as required
= 2 Output **peck depth** even if only for one peck

integer 5 not used

coolant output = (n n) +

coolant ON

integer 6 = 0 Output when encountered in cldata
= 1 Output in next block
= 2 Output with next move

coolant OFF

integer 7 = 0 Output when encountered in cldata
= 1 Output in next block
= 2 Output with next move

integer 8 not used

block order = false or true (DP1203 onwards)

integer 9 = 0 Define block output format as **word order** sequence
= 5 Define block output format as **block order** sequence

spindle output = (n n n n) +

integer 10 = 0 Output when encountered in cldata
= 1 Output in next block
= 2 Output with next move

integer 11 = 0 NO S output for spindle off
= 1 Output S0 for spindle off

integer 12 = 0 NO M code output
= 1 M code output for spindle

integer 13 = 0 NO link with spindle M code to coolant
= 1 Output coolant with spindle M code

integer 14 not used

tlc output = true or false

integer 15 = 0 NO special code output
= 1 Output special code with offset

tool reset coordinate = (n) +

integer 16 = 0 NO special output after tool change
= 1 Force out X, Y and Z after tool change
= 2 Force out X and Y after tool change (Z may be present)
= 3 Force out X, Y then Z on next line after tool change
= 4 Force out Z then X, Y on next line after tool change

integer 17 not used

integer 18 not used

cycle output = (n)

integer 19 = 0 NO canned cycle output move codes
= 1 Output canned cycles (G81 etc)

circle output = (n n n n)

integer 26 = 0 NO circular output
= 1 Circular output

integer 20 = 0 NO circle X, Y coordinates output
= 1 Output circle X, Y coordinates

full circle = true or false

integer 27 = 0 Single quadrant interpolation
= 1 Full circular interpolation

incremental centre = true or false

integer 28 = 0 Absolute position of Circle Centre
= 1 Incremental position of Circle Centre from Start point

integer 21 not used

block start = (n)

integer 22 = n Block start value (i.e. 10)

block increment = (n)

integer 23 = n Block increment value (i.e. 2)

split move = (n) (Linear XYZ moves)

integer 24 = 0 **NO** special X, Y, Z splitting moves (Best for 5 axes)
 = 1 Output X, Y move before Z for downward moves and
 Output Z move before X, Y move for upward move (RAPID only)
 = 2 Output as above for Rapid and Linear moves

tape split on tool change = false or true (New for DP1310)

integer 25 = 0 Tape split on tool change not enabled
 = 1 Tape split on tool change enabled (Useful for Sub-routine work)

integer 29 not used

go home output = true or false

integer 30 = 0 Suppress go home move
 = 1 Output go home move

use partid = true or false

integer 31 = 0 Request PART NAME at run time (i.e. Prompt)
 = 1 Use PARTNO from Duct, PART NAME from PowerMILL

use progid = true or false

integer 32 = 0 Request PROGRAM NUMBER at run time
 = 1 Use **default** program number (1) or as defined by integer 33

program id start = n (default program number = 1)

integer 33 = n Reset program number to "n" value

cutter compensation = (n n n n)+

integer 34 = 0 **NO LONGER REQUIRED**

integer 35 = 0 **NO** radius compensation offset (Usually D)
 = 1 Output radius compensation offset (Usually D ToolNum)

integer 36 = 0 **NO LONGER REQUIRED**

integer 37 = 0 **NO** compensation codes output
 = 1 Output compensation codes (G41 G42 & G40)

rapid feed code = n

integer 38 = 0 **NO** feedrate output on rapid moves
 = 1 Output feedrate on rapid moves.
 (Needs F defined in rapid block)

units = input or metric or imperial

 **integer 39** = 30 Imperial output
 = 40 Metric output
 = 50 Input = Output

coordinates = absolute or incremental

 **integer 40** = 10 All output will be in absolute values (default)
 = 20 All output will be in incremental values
 (**Note :-** These can be redefined in the option move blocks)

rotary axis parameters

azimuth axis units = none or degrees

 **integer 41** = 0 Azimuth axis units linear
 = 1 Azimuth axis units degrees

azimuth axis direction = positive or negative

 **integer 42** = 0 Azimuth axis direction *negative*
 = 1 Azimuth axis direction *positive*

elevation axis units = none or degrees

 **integer 43** = 0 Elevation axis units *linear*
 = 1 Elevation axis units *degrees*

elevation axis direction = positive or negative

 **integer 44** = 0 Elevation axis direction *negative*
 = 1 Elevation axis direction *positive*

spindle x motion = false or true

 **integer 45** = 0 Spindle movement **NOT** in x axis
 = 1 Spindle movement in x axis

spindle y motion = false or true

 **integer 46** = 0 Spindle movement NOT in y axis
 = 1 Spindle movement in y axis

spindle z motion = true or false

 **integer 47** = 0 Spindle movement **NOT** in z axis
 = 1 Spindle movement in z axis

spindle w motion = false or true

 **integer 48** = 0 Spindle movement **NOT** in w axis
 = 1 Spindle movement in w axis

spindle azimuth rotation = false or true

● ●

●

●

- 114 -

integer 60	= 0	Segment tape using feet of tape
	= 1	Segment tape using distance tool travelled
	= 2	Segment tape using number of blocks

<i>counter start = (n)</i>		
integer 61	= 0	Counter start value (for use with variable counter)

<i>counter increment = (n)</i>		
integer 62	= 0	Counter increment value

<i>tape split retract distance = (n)</i>		
integer 63	= n	Distance for the tool to retract from surface when the tape splits.
	= 0	NO retract
	= 100	Retract 100 units from Lift Point
	= -999	Any NEGATIVE number, retract to Datum Z above surface

integer 64	not used	

integer 65	= 0	Not used
	= n	Used only for robot control ??

integer 66	= 0	Print file width (0 = 132 character width)

integer 67	= ?	Final tool number for preselection (-1 last tool) (Extremely rare use - Use ToolNum[x] , NextTool , etc.,)

integer 68	not used	

integer 69	= 0	Reduce plunge feedrate for tapping (0.8 Normal Feedrate)
	= 2	Normal plunge feedrate for tapping

integer 70	= 0	Normal feedrate output
	= 2	Calculate separate feedrates for X, Y, Z axis moves

integer 71	= 0	NO special output
	= 1	Special multiaxis feedrate (<i>set with Int 72</i>)

integer 72	= 0	NO special output
	= 1	Special multiaxis feedrate (<i>Inverse time</i>)

integer 73	= (n)	Special character at tape end (default 0 : 19 = □)

integer 74	= 0	Normal multiaxis output
	= 2	Multiaxes used for tool vector output

integer 75	= 0	Normal feedrate output
	= 1	Special feedrate output for Heidenhain h33 control

integer 76	not used	

([Constant Contour speed](#) code output)

● **integer 77** = 0 Ramp down code calculated on next angle (Heid default)
= 2 Ramp down code calculated on previous angle (All other controls)

● **integer 78** = 1 Split Tape increment (e.g. t01 , t02 etc.)

● **integer 79** = 0 Normal standard output
= 1 Output start and end in agie wirespark format

● This list refers to the most common of Ductpost used integers and named aliase, the information is as accurate as we can make it but changes in the development of Ductpost may effect an output, so use with care and check thoroughly if used in an option.

Список Кодов CLdata

This list provides an indication of the British Standard (BS5110-Pt.2 : 1978) CLdata used by DUCT and PowerMill.

NOTE :- PowerMill has extended the use of some of these functions that will not be recorded here.

MAJOR WORD	INTEGER CODE NUMBER	MEANING
END	1	END
STOP	2	STOP
OPSTOP	3	OPTIONAL STOP
<i>RAPID</i>	<i>5</i>	<i>RAPID</i>
SWITCH	6	SWITCH
RETRCT	7	RETRACT
DRESS	8	DRESS
UNLOAD	10	UNLOAD
PENUP	11	PEN UP
PENDWN	12	PEN DOWN
BREAK	16	BREAK
GOHOME	17	GO HOME
HEAD	1002	HEAD
MODE	1003	MODE
CLEARP	1004	CLEARANCE PLANE
TMARK	1005	TAPE MARK
REWIND	1006	REWIND
CUTCOM	1007	CUTTER COMPENSATION
FEDRAT	1009	FEEDRATE
DELAY	1010	DELAY
AIR	1011	AIR
OPSKIP	1012	OPTIONAL SKIP
LEADER	1013	LEADER
PPLOT	1014	POST-PROCESSOR PLOT
MACHIN	1015	MACHINE
MCHTOL	1016	MACHINING TOLERANCE
SEQNO	1019	SEQUENCE NUMBER
DISPLY	1021	DISPLAY
AUXFUN	1022	AUXILIARY FUNCTION
TOOLNO	1025	TOOL NUMBER
ROTABL	1026	ROTATE TABLE
ORIGIN	1027	ORIGIN

COOLNT	1030	COOLANT
<i>SPINDL</i>	<i>1031</i>	<i>SPINDLE</i>
TURRET	1033	TURRET
ROTHED	1035	ROTATE HEAD
THREAD	1036	THREAD
TRANS	1037	TRANSLATE
OVPLOT	1042	OVERPLOT
LETTER	1043	LETTER
PPRINT	1044	POST-PROCESSOR PRINT
<i>PATNO</i>	<i>1045</i>	<i>PART NUMBER</i>
INSERT	1046	INSERT
PREFUN	1048	PREPARATORY FUNCTION
COUPLE	1049	COUPLE
PITCH	1050	PITCH
<i>CYCLE</i>	<i>1054</i>	<i>CYCLE</i>
LOADTL	1055	LOAD TOOL
SELCT	1056	SELECT TOOL
CLRSRF	1057	CLEARANCE SURFACE
DRAFT	1059	DRAFT
LINTOL	1067	LINEARISATION TOLERANCE.
CLDIST	1071	CLEARANCE DISTANCE
CHUCK	1073	CHUCK
CLAMP	1074	CLAMP
<i>PPFUN</i>	<i>1079</i>	<i>POST-PROCESSOR FUNCTION</i>
STAN	1080	SETTING ANGLE
OFSTNO	1083	OFFSET SWITCH NUMBER
PIERCE	1090	PIERCE
SAFPOS	1094	SAFE POSITION

The selection below indicates use, taken from a PowerMill Cldata output example :-

1 2000 *1045* PARTNO
cachsendrehung

7 2000 *1031* SPINDL ON
SPINDLE SPEED = 10000.0 RPM CLW
RANGE Undefined (Select range)
TOOL TIP RADIUS = 0.0

5 2000 *1079* PPFUN
FEDRAT 10000 500 500

6 2000 *1079* PPFUN
CUTLEN 79.60

```

-----
 7  2000  1079  PPFUN
      CUTTIM 9.00
-----
17  2000   5    RAPID
-----
18  5000   5    RAPID TOOL MOTION RECORD
X-COORD  Y-COORD  Z-COORD    TOOL X  TOOL Y  TOOL Z
19.136200 0.000000 16.996048  -0.139173 0.000000 0.990268
-----
51  2000  1054  CYCLE (288)
      CLEAR PLANE = 5.000000
      FEEDRATE = 500.000000
      NO. OF PECKS = 15
      CYCLE Z HEIGHTS (PECK HEIGHTS) =
        7.000000

```

Homepa Word[x]

In the inbuilt machine source files of Ductpost there is an initial list of defined words, the lists given below are the values of those words, and the associated letter for the various controls :-

[No.]	Standard Word / Address Letter List	Other Controls	Variations
1	define word / address letter = "/"	.	.
2	define word N address letter = "N"	h155 heid heid400	address letter = " " address letter = " " address letter = " "
3	define word G1 address letter = "G"	dyna h155 heid heid400 siem850	address letter = "(" address letter = "L" address letter = "L" address letter = "L" address letter = "GD"
4	define word G2 address letter = "G"	dyna h155 heid heid400	address letter = ")" address letter = "C" address letter = "C" address letter = "C"
5	define word G3 address letter = "G"	.	.
6	define word G4 address letter = "G"	h155 heid heid400	address letter = "CYCL DEF" address letter = "CYCL DEF" address letter = "CYCL DEF"
7	define word G5 address letter = "G"	h155 heid heid400	address letter = " " address letter = " " address letter = " "
8	define word G6 address letter = "G"	h155 heid heid400	address letter = "CYCL DEF" address letter = "CYCL DEF" address letter = "CYCL DEF"
9	define word G7 address letter = "G7"	ab84 fanuc fanucom fanuc6m fanuc11m fanuc15m h33	address letter = "G" address letter = "G" address letter = "G" address letter = "G" address letter = "G" address letter = "G" address letter = "G"
10	define word X address letter = "X"	.	.
11	define word Y address letter = "Y"	.	.
12	define word Z address letter = "Z"	.	.
13	define word I address letter = "I"	h155 heid heid400	address letter = "X" address letter = "X" address letter = "X"
14	define word J address letter = "J"	h155 heid heid400	address letter = "Y" address letter = "Y" address letter = "Y"
15	define word K address letter = "K"	h155	address letter = "Z"

		heid heid400	address letter = "Z" address letter = "Z"
16	define word R address letter = "R"	bosch deckel4 deckel11 maho ph432 siem850	address letter = "," address letter = "SA" address letter = "SA" address letter = "B" address letter = "B" address letter = "R2="
17	define word R2 address letter = "R"	h155 heid heid400 heidiso num siem850	address letter = ".1 SET UP" address letter = ".1 SET UP" address letter = ".1 SET UP" address letter = "PO1" address letter = "ER" address letter = "R2="
18	define word Q address letter = "Q"	bosch deckel4 deckel11 fidia heidiso	address letter = "," address letter = "MI" address letter = "MI" address letter = "D" address letter = "PO3"
19	define word A address letter = "A"	.	.
20	define word B address letter = "B"	.	.
21	define word C address letter = "C"	.	.
22	define word U address letter = "U"	.	.
23	define word V address letter = "V"	.	.
24	define word W address letter = "W"	.	.
25	define word Z2 address letter = "Z"	bosch deckel4 deckel11 h155 heid heid400 heidiso fidia siem850	address letter = "[" address letter = "TA" address letter = "TA" address letter = ".2 DEPTH" address letter = ",2 DEPTH" address letter = ".2 DEPTH" address letter = "PO2" address letter = "E" address letter = "R3="
26	define word F address letter = "F"	.	.
27	define word S address letter = "S"	.	.
28	define word T address letter = "T"	.	.
29	define word M1 address letter = "M"	h155 heid	address letter = " " address letter = " "
30	define word M2 address letter = "M"	h155 heid	address letter = " " address letter = " "
31	define word L address letter = "L"	.	.
32	define word P address letter = "P"	.	.
33	define word D address letter = "D"	fagor	address letter = "."
34	define word E address letter = "E"	dyna heidiso	address letter = "\$" address letter = "*"
35	define word H address letter = "H"	h155	address letter = "D"

		heid heid400	address letter = "D" address letter = "D"
36	define word O address letter = "O"	.	.
37	define word MS address letter = "("	acra8 bosch bostom tiger	address letter = "(MSG," address letter = "(MSG," address letter = "%" address letter = "/"
38	define word EM address letter = ")"	.	.
39	define word EOB address letter = "*"	ab84 acra8 bosch bostom deckel4 deckel11 dyna eberle elexa fadal fagor fanuc fanucom fanuc6m fanuc11m fanuc15m fidia h33 h155 heid heid400 heidiso hurco maho matsura mazak mitsu num okuma ph432 selca siem850 tiger	define word OP address letter = "/" define word N2 address letter = ":" define word ID address letter = "(DFS, P" define word OP address letter = "/" define word N2 address letter = "N*" define word OP address letter = "/" define word OP address letter = "/" define word N2 address letter = "N*" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word null address letter = " " define word OP address letter = "/" define word FF address letter = "F" define word FF address letter = "F" define word FF address letter = "F" define word CF address letter = "PO5" define word OP address letter = "/" define word K2 address letter = "J0.2K" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/" define word K2 address letter = "J0.2K" define word OP address letter = "/" define word OP address letter = "/" define word OP address letter = "/"
40	.	ab84 acra8 bosch bostom deckel4 deckel11 dyna eberle elexa fadal	define word null address letter = " " define word OP address letter = "/" define word PA address letter = "," define word null address letter = " " define word OP address letter = "/" define word ID address letter = "%" define word null address letter = " " define word OP address letter = "/" define word ID address letter = ":" define word ID address letter = ":"

		fagor fanuc fanucom fanuc6m fanuc11m fanuc15m h33 h155 heid heid400 heidiso hurco maho matsura mazak mitsu num okuma ph432 selca siem850 tiger	define word ID define word ID define word ID define word ID define word ID define word null define word XXF define word XXF define word T1 DEF" define word ID define word Z3 define word OP define word ID define word ID define word ID define word ID define word OP define word ID define word ID "&MPF" define word ID	address letter = "%" address letter = ":" address letter = ":" address letter = ":" address letter = ":" address letter = " " address letter = " " address letter = " " address letter = "TOOL address letter = "%" address letter = "Z" address letter = "/" address letter = "O" address letter = "O" address letter = "O" address letter = "O" address letter = "/" address letter = ":" address letter = address letter = ":"
41	.	acra8 bosch deckel4 deckel11 eberle elexa fadal fagor fanuc fanucom fanuc6m fanuc11m fanuc15m h155 heid heid400 heidiso hurco maho matsura mazak mitsu num okuma ph432 selca siem850 tiger	define word ID "(MSG," define word EI E D)" define word ID define word P1 D11-000000--" define word ID define word QZ define word Q1 define word null define word Q1 define word Q1 define word Q1 define word Q1 define word Q1 define word Q1 define word MP define word MP define word T2 CALL" define word null define word null define word ID define word null define word null define word Q1 define word P1 define word Q1 define word ID	address letter = address letter = ", R W address letter = "(&%" address letter = " (M- address letter = "(&%" address letter = "QZ" address letter = "Q" address letter = " " address letter = "Q" address letter = "Q" address letter = "Q" address letter = "Q" address letter = "Q" address letter = "Q" address letter = "M" address letter = "M" address letter = "TOOL address letter = " " address letter = " " address letter = "N9" address letter = " " address letter = " " address letter = "Q" address letter = "P1=" address letter = "Q" address letter = "N9"

			define word null define word R11 define word null	address letter = " " address letter = "R11=" address letter = " "
42	.	acra8 bosch deckel4 deckel11 eberle elexa fadal fanuc fanucom fanuc6m fanuc11m fanuc15m h155 heid heid400 maho ph432 num siem850	define word P1 define word OP define word ED "/000000)" define word P2 define word ED "/000000)" define word PA define word null define word null define word null define word null define word null define word T1 DEF" define word T1 DEF" define word TL define word P1 define word P1 define word null define word null	address letter = "P1=" address letter = "?" address letter = " address letter = "." address letter = " address letter = "PA" address letter = " " address letter = " " address letter = " " address letter = " " address letter = " " address letter = " " address letter = "TOOL DEF" address letter = ";TOOL DEF" address letter = "L" address letter = "P1=" address letter = "P1=" address letter = " " address letter = " "
43	.	acra8 bosch deckel11 eberle elexa h155 heid heid400 maho ph432	define word null define word EC define word P3 define word null define word null define word T2 CALL" define word T2 CALL" define word RR define word null define word null	address letter = " " address letter = "]" address letter = "." address letter = " " address letter = " " address letter = "TOOL CALL" address letter = "TOOL CALL" address letter = "R" address letter = " " address letter = " "
44	.	bosch deckel11 h155 heid heid400	define word null define word P4 define word TL define word TL define word null	address letter = " " address letter = "-" address letter = "L" address letter = "L" address letter = " "
45	.	deckel11 h155 heid heid400	define word null define word RR define word RR define word Z3 PECKG"	address letter = " " address letter = "R" address letter = "R" address letter = ".3
46	.	h155 heid heid400	define word null define word null define word ID	address letter = " " address letter = " " address letter = " "

47	.	h155 heid heid400	define word Z3 address letter = ".3 PECKG" define word Z3 address letter = ".3 PECKG" define word Z4 address letter = "Z"
48	.	h155 heid heid400	define word ID address letter = " " define word ID address letter = " " define word OP address letter = "/"
49	.	h155 heid heid400	define word Z4 address letter = "Z" define word Z4 address letter = "Z" define word DR address letter = "DR"
50	.	h155 heid heid400	define word OP address letter = "/" define word OP address letter = "/" define word circ address letter = "CC"
51	.	h155 heid heid400	define word DR address letter = "DR" define word DR address letter = "DR" define word DW address letter = "DWELL"
52	.	h155 heid heid400	define word circ address letter = "CC" define word circ address letter = "CC" define word FMAX address letter = "F9999"
53	.	h155 heid	define word DW address letter = "DWELL" define word DW address letter = "DWELL"
54	.	h155 heid	define word FMAX address letter = "F9999" define word FMAX address letter = "FMAX"

For use of word values see :- [Word Values](#)

Раздел 7. Опции Структуры

Пример Структуры Option-файла

Урок 1

Введение

Здесь приведены не полные правила строения option-файла, а лишь руководство для начинающих, которое может сделать задачу немного проще.

Option-файл служит для изменения и дополнения вывода стандартного встроенного постпроцессора DUCTpost, и выполнения необходимого формата вывода программы. Для различных технико-технологических условий можно создавать различные option-файлы на одну и ту же Систему ЧПУ (например, Метрические или Дюймовые, Абсолютные или Относительные координаты). Разумеется эти option-файлы должны иметь разные имена.

Для создания option-файла, требуется следующее:

- 1/ Текстовый редактор
- 2/ PAF Лицензия для специфической Системы ЧПУ**
- 3/ PAF Лицензия для **LONG** если опции префышают **50 рабочих строк программы****
(Excludes spaces, blank lines, or # commented lines)
(** - С версии **DP1331 не требуется**)

Имеется два начальных способа создания опций: 1 – копирование макета опций для систем ЧПУ в директорию [/dcam/config/ductpost/](#); 2 – начиная «с нуля» с бланка. Если Вы начинаете «с нуля», существуют 4 обязательных строки:

Line 1. [machine xxxxx](#) ## xxxxx имя системы ЧПУ, например [fanuc](#)
Line 2. пробел
Line 3. [end](#)
Line 4. пробел

Option-файле "[machine xxxxx](#)" – всегда первая строка, и "[end](#)" – всегда последняя, и не должны иметь никаких начальных пробелов.

Сохраните с соответствующим [\[именем файла\]](#) и расширением [.opt](#), обычно, в вашем [рабочем](#) каталоге.

Это будет основой для структуры option-файла.

Пример файла программы

Это типичный пример требуемого формата вывода для файла программы известной Системы ЧПУ FANUC

```
%
O5222
(THIS PROGRAM MACHINES THE)
(FOR PJ# )
(CNC MILL LOCATION= X, Y, Z)
(FILE NAME )
(DATE LAST RAN:)
(DATE LAST EDITED:)
(OPERATOR )
(PROGRAMMER:)
G17G20G90G54
T99
T1
M6
S2400M3(TOOL 1 IS A 1/2" HOG BALL)
G0X0Y0
G43Z3.H1
M8
G0X0Y.0475
G0Z.5
G1Z0F15.
.....
.....
G0Z.2
T2
M6
S2400M3(TOOL 2 IS A 1/4" CARB BALL)
G0X0Y0
G43Z3.H2
M8
G0Z.5
G1Z0F30.
.....
.....
T99
M6
M2
%
```

Предлагается следующий метод обеспечения в option-файле требуемого формата вывода программы:

Ступень1. Необходимая

Первое правило – установить, какие опции системы ЧПУ должны быть заданы для Fanuc и др. (например, Heid, Fadal и т.д.)

Второе правило – состоит в том, чтобы пробывать и получать хороший типичный файл программы, охватывающий различные существующие опции станка. Например: инициализация старта, требования вывода круговых и мульт-осевых перемещений и т.д. (в вышеупомянутом

примере информации немного, но он охватывает старт программы, смены инструмента, конец программы и дополнительные специфические требования).

Третье правило – определение базового встроенного в Ductpost файла опций, как самого близкого к вашей Системе ЧПУ. Это отпадает если нет близкой системы ЧПУ. В этом случае используйте встроенный файл " **standard** ".

Для получения полного описания встроенного файла Системы ЧПУ наберите в окне: **ductpost -l** (строчная L).

Неплохо будет иметь 2-3 коротких рабочих файла обработки охватывающие различные функции - для выполнения.

Основа Старта:

Для получения нужного файла программы, правильным будет начать со Старта и далее постепенно рассматривать другие опции.

В этом примере мы будем использовать Систему ЧПУ **Fanuc**.

Имея файл обработки сначала прогоните его через встроенный постпроцессор, для проверки подобности основного вывода. В рабочем окне: **ductpost fanuc [filename].cut ## .cut** можно опустить, если используется другое расширение.

Рекомендуется, чтобы постпроцессирование выполнялось в рабочем окне, поскольку так проще заметить сообщение об ошибках. Хотя в самом последнем релизе Ductpost постпроцессирование прерывается автоматически, если имеются погрешности (сообщение обычно дает полное описание ошибки, указывающей на то, что не верно в option-файле.

Для начала откройте дамп нужной встроенной Системы ЧПУ для использования ее в качестве источника. Сделать это можно в операционном окне:

ductpost -w [control name] > [control name].dmp

В первую очередь сравните Старт выполняемый встроенным файлом с требуемым Стартом.

Встроенный вывод

%
:0001
N10G91G28X0Y0Z0
N20G40G17G80G49
N30G0G90Z10.

Требуется

%
O5222
(Messages - 8 lines)
G17 G20 G90 G54

Далее чтобы избежать беспорядка: первоначальная опция будет выглядеть как- ' **X** ', а измененная как - ' **X** ' (в обоих случаях игнорируйте кавычки).

Программы отличаются ID (' : ' - ' **O** '), и выводом номеров блоков.

Итак, исправим первое.

В исходном дампе найдите ' **ID** ' в виде строки ' **define word ID** ', скопируйте три строки этого блока в новый option-файл, вставив его ниже первой пустой линии после **machine fanuc**, и измените следующее: ' **define word ID** ' на ' **define format (ID)** ' и ' **address letter = ":"** ' на ' **address letter = "O"** '

Сделайте тоже самое для ' **N** ', разместив его ниже блока ID, но на этот раз сотрите символьную строку адреса и замените на ' **not permanent** '

Сохраните изменения и сделайте пробный прогон вашего файла обработки, для просмотра вывода.

Теперь, Вы должны увидеть, что вместо ':' появилось 'O', и исчезли номера блоков. Но теперь мы имеем целый ряд сообщений (-----), который вероятно не будет требоваться. Пока подавите их вставив опцию после последнего блока и перед 'end', 'message output = false'

Затем, займитесь заголовком программы. Скопируйте 'define block tape start' в опцию после строки 'message output = false'.

Пока игнорируйте (требуемые сообщения) некоторые из них, тем более, что пока мы не можем подставить их. Сначала установим правильные коды. Первые две строки блока правильные, а последние три можно свести в одну добавив T99 как показано ниже:

Старый метод	Предпочтительный метод
N ; G3 17 ; G4 20 ; G5 90 ; G6 54	N ; xy plane ; imperial data ; absolute data ; G6 54
N ; T 99	N ; tool number 99

G-коды взяты из списка *дамп-файла* 'define codes', но так как 'xy plane' - G3 17 и код 'from' - G3 54, мы не можем иметь две группы G3 на одной и той же строке, поэтому устанавливаем G3 54 как G6 54 – что будет приемлемо. Держите их в последовательности, т.к. это упрощает отладку, и обычно станок не следит за порядком not particular about the order.

Затем из примера мы видим, что есть T99 при старте и конце, таким образом можем предположить что они установлены. Поэтому включите это в блок старта как показано выше. Вы можете заметить что **номер инструмента (T)** был использован для этого и задан как 99.

Ниже мы скажем о формате построения option-файла:

- а) **НИКОГДА** не используйте **Клавишу табуляции** для отступа (обзаца), используйте только клавишу **Пробел**.
- б) **ПРОБЕЛЫ** должны использоваться до и после разделителя ; (точка с запятой).
- с) **ПРОБЕЛ** требуется между **Словом** и **Значением**, т.е. **T пробел 99 пробел ; пробел G6 пробел 54**
- д) **#** используется как **РЕМ**-формулировка для комментария.
- с) Рекомендуются чтобы **ПРОБЕЛ** был введен между каждым заданным блоком, для большей наглядности, например:

```
end define
ПРОБЕЛ или #
define block .....
```

Сохраните и протестируйте полученный результат.

Далее – Смена инструмента, и если мы будем смотреть на *дамп-файл*, то Вы заметите, что есть 3 опции: 'define block tool change', 'define block tool change first', 'define block tool change clear'. 'first' и 'change' имеют заданные данные, а 'clear' - пустая.

'define block tool change first' – определяет есть ли какие-нибудь требования вывода на данном этапе, после чего за смену инструмента отвечает 'define block tool change'.

Из нашего примера мы видим, что все смены инструмента имеют одинаковый формат вывода, поэтому мы можем использовать 'define block tool change' для всех смен, и стереть данные в 'define block tool change first'. Тем не менее лучше использовать следующий формат:

```
define block tool change first
  N ;
end define
```

Игнорируйте ' **define block tool change clear** ' т.к. это очень редко используется, но всегда проверяйте, что она не задана.

Скопируйте оба блока из исходного файла и поместите их после ' **define block start**

'. Отредактируйте их следующим образом:

' **first** ' - сотрите все, оставив только строки ' **define block tool change first** ' и ' **end define** ' (или, предпочтительно, сделать так как описано выше).

' **change** ' – сотрите первые три строки, т.к. здесь не требуются предварительные установки перед сменой инструмента, также сотрите и последнюю строчку, оставив только:

```
  N ; T ToolNumber ; change tool  ( N ; tool number ToolNum ; change tool также может
использоваться)
```

Для получения основного вывода требуемого для смены инструмента, мы рекомендуем следующим образом (пока игнорируя Данные инструмента).

```
  N ; T ToolNumber
  N ; M1 6  ( or, change tool )
  N ; S ToolSpeed[ToolNum] ; M1 3  ( N ; spindle ToolSpeed[ToolNum] ; spindle on cw )
```

Сохраните и протестируйте. Вы должны обнаружить что вывод смены инструмента практически верен до команды M8, за исключением дополнительного вывода **Шпинделя** и **M3** на первой строке перемещения.

Они выводятся из **блока ускоренных перемещений**.

Скопируйте из **дамн-файла** блок ускоренных перемещений и отредактируйте его следующим образом:

```
  N ; G1 ; G2 ; G3 ; G5 ; G6 ; X ; Y ; Z ; H ; M1 ; M2
или предпочтительно
  N ; rapid ; G2 ; G3 ; G5 ; G6 ; x coord ; y coord ; z coord ; tool length ; M1 ; M2
```

Строки ' **message output = false** ' и ' **integer 12 = 0** ' **подавят вывод M3** (См. «Информация Строения Данных»)

Сохраните и протестируйте. Теперь вывод должен напоминать основной формат старта программы, смены инструмента и без вывода СООБЩЕНИЙ (о них пойдет речь в **Уроке 2**.

Единственное изменение требуемое на данном этапе – это конец программы.

Однако, спервы мы должны исправить одно упущение в исходном тексте встроенного файла (**дамн-список**), в котором нет определения ' **define block move linear** '. Хотя возможно это не создаст проблем, но было несколько моментов в старых версиях Ductpost когда это упущение вызывало проблемы.

Теперь всегда рекомендуется добавлять это, при отсутствии во встроенном файле. Необходимый минимум – это добавить опцию ' **define block move rapid** ', см. ниже:

define block move linear

N ; linear ; G2 ; x coord ; y coord ; z coord ; tool radius ; feedrate ; M1 ; M2
end define

Теперь займемся концом программы. Скопируйте ' **define block tape end** ' в Выш новый option-файл и поместите его после только сто созданной ' **define block move linear** '.

Здесь требуется полный пересмотр, т.к. во встроенном файле много нетребуемой информации для вывода. Поэтому разумнее будет стереть все содержимое блока и начать заново, добавив новые данные:

N ; tool number 99
N ; change tool
N ; end of tape
"%"

Еще один фактор должен быть включен для используемых в США дюймовых единиц (**Дюймы**). Это может быть выполнено двумя методами. Вставьте одну из двух строк ниже строки ' **integer 12 = 0** '.

' units = input ' или **' units = imperial '**

По умолчанию Ductpost обычно отражает вывод в любых используемых единицах (первая опция). Тогда как вторая строка преобразует вывод **Метрических единиц** в **Дюймовые**. (И наоборот, ' **units = metric** ' преобразует вывод **Дюймовых единиц** в **Метрические**). Используйте второй метод.

ПРЕДУПРЕЖДЕНИЕ: НЕ БУДУТ ПРЕОБРАЗОВЫВАТЬСЯ заданные значения типа: **withdrawal amount = 100.0 ; tape split retract distance = 50.0 ; xcoord 53000 ; И другие** они должны быть определены в соответствующих единицах.

Сохраните и проверьте.

Теперь Вы должны увидеть, что большая часть программы соответствует примеру. Продолжим работу по добавлению СООБЩЕНИЙ в заголовке программы (См. «Добавление Данных Сообщений»).

Прежде чем мы это сделаем, стоит добавить в option-файл хронологический отчет для регистрации сделанных изменений. Пример этого показан ниже:

Конструкция Option-файла:

```
machine fanuc
#2
# History
# Option for North American Support - WizzyWig M/c. with Fanuc control
# am 15/12/98 Ref.No. 0280 Issue 1.1
#
#
#8          ## Line number to aid debugging.
define format ( N )
  not permanent
end define
#12
define format ( ID )
  address letter = "O"
```

```

    end define
#16
    message output = false
    integer 12      = 0
    units           = imperial
#20
    define block tape start
        "%"
        ID ProgID
        N ; xy plane ; imperial data ; absolute data ; G6 54
        N ; tool number 99
    end define
#27
    define block tool change first
        N ;
    end define
#31
    define block tool change
        N ; tool number ToolNumber
        N ; change tool
        N ; spindle ToolSpeed[ToolNum] ; spindle on cw
    end define
#37
    define block move rapid
        N ; rapid ; G2 ; G3 ; G5 ; G6 ; x coord ; y coord ; z coord ; tool length ; M1 ; M2
    end define
#41
    define block move linear
        N ; linear ; G2 ; x coord ; y coord ; z coord ; tool radius ; feedrate ; M1 ; M2
    end define
#45
    define block tape end
        N ; tool number 99
        N ; change tool
        N ; end of tape
        "%"
    end define
#52
end

```

Пример Конструкции Option-файла.

Lesson 2

Введение

Теперь посмотрим на представленные в начале программы сообщения примера части программы приведенной ниже:

```
%
O5222
(THIS PROGRAM MACHINES THE)
(FOR PJ# )
(CNC MILL LOCATION= X, Y, Z)
(FILE NAME )
(DATE LAST RAN:)
(DATE LAST EDITED:)
(OPERATOR )
(PROGRAMMER:)      ## ____ ^^ Tape Start Info ^^ ____
G17G20G90G54
T99
T1
M6
S2400M3(TOOL 1 IS A 1/2" HOG BALL)  ## Tool Info
G0X0Y0
G43Z3.H1
M8
```

До версии Ductpost DP1205 возможности вывода текста были чрезвычайно ограничены. Однако, теперь мы можем обеспечить вывод данных ' [Part Name](#) ' используя Powemill и с версии DP1331.

Информационные данные инструмента

Если мы начнем с требуемого описания инструмента, например (TOOL x IS A 1/2" HOG BALL).

Лучше всего будет выполнить это следующим образом: (TOOL x Dia. 0.5 Tip Rad. 0.25). Можно использовать как заглавные так и строчные буквы. Также можно добавить и длину инструмента. Но невозможно обеспечить формат дробей в дюймовой системе.

Сначала нам нужно создать некоторые новые слова, для фиксации там данных. Они будут для Диаметра Инструмента и Радиуса вершины инструмента.

Если мы добавим указанное ниже и разместим в начале после строки ' [machine xxxx](#) ' и до ' [define format \(N \)](#) ' (это сделано для удобства их поиска в option-файле)

```
define word TN
  address letter = "(TOOL "
  address width  = 5
  field width    = 2
end define
#
define word TD
  address letter = " Dia. "
  address width  = 6
  field width    = 5
```

```

    scale factor    = 2
end define
#
define word TR
    address letter = " Tip Rad. "  ## включая пробелы если требуется
    address width  = 10            ## считая пробелы
    field width    = 5             ## set to number of places + point
end define

```

тогда мы можем добавить дополнительные опции характеристики к вышеупомянутому, добавляя их после новых слов:

```

define format ( TD TR )      ## TN определено выше.
    metric formats
    decimal point = true
    decimal places = 3
    leading zeros = false
    trailing zeros = false
    imperial formats = metric formats
end define

```

и наконец мы должны добавить эти слова в порядковый список слов, иначе в выводе ни чего не появится.

Самый простой способ, состоит в том, чтобы поместить их в конце встроенного списка, добавляя строку опций: " **word order = (+ TN TD TR)** ", размещать их нужно в порядке их появления в программу. Помещая это перед строкой **message output = false** '.

Затем надо добавить данные в **' define block tool change '** как показано ниже:

```

N ; spindle ToolSpeed[ToolNum] ; spindle on cw ; TN ToolNumber ;
TD ToolRadius[ToolNum] ; TR ToolRadius[ToolNum] ; EM =C

```

Далее приведена поясняющая информация, хотя часто это может быть очевидным.

N Вывод номера блока (строки). В нашем случае не требуется, но лучше включить.
spindle (S) Вывод скорости вращения шпинделя, определяемая переменной **' ToolSpeed[ToolNum] '**.
spindle on cw (M1 3) Функция Auxiliary - в этом случае используется для вывода **M3** (вращение шпинделя по ЧС). Подробнее об этом позже.
TN Фиксируемый адрес в выводе **ToolNumber**
TD Фиксируемый адрес для диаметр инструмента., обеспеченный формулировкой **ToolRadius**, это *умножается на 2* в установке **TD масштабного фактора**.
 Рекомендуется всегда использовать **' ToolRadius[ToolNum] '** и только **HE ' ToolRadius '**, т.к. в некоторых случаях выводится неправильное значение.
TR Подобен TD и используется в выводе **Tip Radius** из инструмента.
EM Это слово **') '** вывода сообщений, здесь мы используем **=C для этого вывода**. One point while we are on this, it is possible that the EM word will output as **) 0** , which means the inbuilt definition field width has not been set to zero. If this is the case then it will mean that EM will need to be re-formatted. It would have been in order to use **"")"** on its own instead of EM.

Примечание:

spindle on cw (M1 3) = указывает предпочитаемый формат (здесь показан старый формат)

spindle on cw и т.д. должен быть написан **строчными буквами** и определен ключом или блоком кодов.

Если мы сохраним и протестируем эти добавления, то увидим следующий вывод:

S1850M3) (TOOL 1 Dia. 10. Tip Rad. 5.

Почти правильно, но ') ' не в конце как требуется. Почему?

Если мы посмотрим на список ' **word order** ' *дам-файла*, то заметим, что **ЕМ** не последнее слово в списке, именно это и портит содержание.

Для исправления этого мы имеем две опции:

- 1/ Скопируйте порядковый список слов из встроенного файла и перестройте его порядок.
- 2/ Вставьте строку ' **block order = true** ' ниже строки ' **integer 12 = 0** ' для гарантии определенного списка слов (используя **integer 9 = 5** в версиях **Ductpost1100** и **более ранних**, **БУДЬТЕ ОСТОРОЖНЫ**, могут быть уловки, особенно в Heidenhains. Более поздние версии (с DP1203 стабильнее и приемлемы для любого формата, предпочтительно **block order = true**)

Если мы сохраним и попробуем теперь, то обнаружим, что ") " располагается правильно.

S1850M3 (TOOL 1 Dia. 10. Tip Rad. 5.)

Примечания в Квадратных скобках [].

Формулировка способа вывода ToolRadius ' **ToolRadius[ToolNum]** ' - определяется для особой смены инструмента.

Формулировка способа вывода ToolRadius ' **ToolRadius[2]** ' - определяется для особой смены второго инструмента, но НЕ инструмента №2

Подобная формулировка вывода ' **S ToolSpeed[2]** ' – для задания скорости смены второго инструмента.

Пример: **Временно** добавьте эту строку в смену инструмента для просмотра вывода.

**N ; spindle ToolSpeed[2] ; spindle on cw ; TD =C ; TN ToolNumber[2] ;
TD ToolRadius[2] ; TR ToolRadius[2] ; EM =C**

В файле обработки, имеющем вторую смену инструмента, мы увидим такой вывод:

**S1850M3 (TOOL 1 Dia. 10. Tip Rad. 5.)
S2250M3 (TOOL 2 Dia. 6. Tip Rad. 3.)**

После проверки **удалите** добавленную строку.

Первую ступень по созданию нужных сообщений – мы закончили.

1/ Начальные Данные Программы (Ввод номера программы)

- 1 %
- 2 O5222
- 3 (THIS PROGRAM MACHINES THE)
- 4 (FOR PJ#)
- 5 (CNC MILL LOCATION= X, Y, Z)
- 6 (FILE NAME)
- 7 (DATE LAST RAN:)

```
8 (DATE LAST EDITED:)  
9 (OPERATOR )  
10 (PROGRAMMER:)  
11 G17G20G90G54  
12 T99
```

Единственные подробности которые мы бы могли иметь с версией DP1205 - это Строка 2 – номер программы, и Строка 3 – дата создания программы. В новых версиях затратив минимум времени мы можем больше..

Теперь возможно задать CNC MILL LOCATION для Базовой точки файла обработки. Для этого упражнения мы создадим необходимые данные.

Номер программы. 05222

По умолчанию **Ductpost** выводит ' 1 ' используя ' **ProgID** ', которая может быть переопределена вставкой в option-файл, например ' **program id start = 5222** '.

Это приведет к **фиксации** номера программы при использовании option-файла.

По умолчанию ' 1 ' может производить ' **0001** или **1000** ' в зависимости от формата определения **word ID**: ' **leading zeros = true** 'или' **trailing zeros = true** ' назначенных в блоке **define format (ID)**.

Различные доступные методы вывода переменной Номера программы:

1/

Ductpost имеет вывод Переменного номера программы, но это должно производиться ' **Prompted** ' во время постпроцессирования. Это может задаваться вставкой ' **use progid = false** ' в опции.

Однако, это игнорируется, если постпроцессирование выполняется через PowerMILL. Исправлением этого мы займемся вскоре.

2 Возможно это самый простой путь, который заключается в том, чтобы использовать **файл обработки** или **файл программы**. Имя производится через переменную ' **JobName** ', и файлы обработки или программы определяются как **5222.cut**, или **5222.tap**.

В опции ' **define block tape start** ' мы должны изменить **ID Progid** на **ID Jobname** . Этот метод мы будем испытывать в нашем примере.

(**Примечание:** Метод программы не функционирует с версиями Ductpost **до** DP1206)

3 Другой метод доступный нам – через ввод в PowerMILL **Имени детали**, с помощью чего мы можем вывести ограниченные данные. При получении **Part Number** из **Part Name** , необходимо этот формат поддерживать всегда, если используется данный метод.

Как пример мы возьмем следующий ввод:

(Здесь задано: **№ программы-макс. 5 цифр**, **Вид детали-макс. 25 цифр**, **Название компании - макс. 12 цифр**, **№ детали – макс. 6 цифр**, **Инициалы программиста – макс. 3 цифры** с разделением блоков с помощью **--**, и дополнением **0** или *****)

Part Name 05222-PORT WING STRUT OF B52***-BOEING INC**-325A59-AM*

Сперва нас интересует первая часть заданная **5 символами**, но 5222 только 4 числа, для дополнения был добавлен **начальный ноль**. **Начальные нули необходимы** для поддержки длины поля в 5 символах. Остальную часть последовательности мы рассмотрим позже.

В ' **define block tape start** ' должна быть изменена опция с ' **ID Progid** ' на ' **ID PartID** '.

Т.к. **word ID** в настоящее время ограничена **первыми 4 символами**, в последовательности ' **05222** ', выводится только **0522**, если **первоначальные нули** не заданы. Пятизначное число возможно, если **field width** из **word ID** переопределена и равна **5**.

PartID, выбирает первый символ как отправную точку в последовательности, и пределы *field width* из **word ID** ограничивают выводимое количество символов.

Поэтому наш выбор следующий:

```
define format ( ID )  
  leading zeros = true  
end define
```

или > **program id start = 5222**

```
define block tape start  
  N ; "%"  
  N ; ID ProgID  
  .....и т.д.  
end define
```

```
define block tape start  
  N ; "%"  
  N ; ID ProgID  
  ..... и т.д. .  
end define
```

```
%  
O0001  
.... и т.д.
```

```
%  
O5222  
.... и т.д.
```

```
cut file name ( и/или tap file от DP1206 )  
5222.cut ( 5222.tap )
```

или > **define format (ID)**
field width = 5
end define

```
define block tape start  
  N ; "%"  
  N ; ID JobName  
  ..... и т.д.  
end define
```

```
define block tape start  
  N ; "%"  
  N ; ID PartID  
  ..... и т.д.  
end define
```

```
%  
O5222  
.... и т.д.
```

```
%  
O5222  
.... и т.д.
```

Примечание: Видна тенденция использовать имя *tape* / *cut*-файла как Индикатор Программы, поэтому **JobName** вероятно наилучший выбор.

ID обычно задается 8-символьной шириной поля, т.к. большинство Систем ЧПУ поддерживают только такой номер.

2/ Стартовые данные программы (Ввод данных программы)

Далее мы приступим к оставшейся части – ввод Старта программы для вывода строк с3 по 10, для чего нам понадобится введение некоторых дополнительных определенных слов.

Начнем со строки 3 (**THIS PROGRAM MACHINES THE** [Описание дерали])

К option-файлу добавим следующее, разместив добавления после последнего блока слов:

```
define word ID2  
  address letter = "(THIS PROGRAM MACHINES THE "  
  address width = 27
```

```

    field width      = 33
end define

```

Добавим в скобках **ID2** к строке *word order* перед **TD**. Всегда проверяйте наличие пробелов между словами.

Теперь измените определение блока старта программы следующим образом:

```

define block tape start
    "% "
    ID Jobname ( или ID PartID )
    ID2 PartID ; EM =C
    N ; xy plane ; incremental data ; absolute data ; G6 54
    N ; tool number 99
end define

```

Сохраните и протестируйте. Вы увидите вывод подобный этому:

```

%
O5222
(THIS PROGRAM MACHIN    05222-PORT WING STRUT OF B52***-B )
G17G20G90G54
T99

```

Не совсем то, что требовалось. Адрес **ID2** был усечен по числу допускаемых символов в словах, ограничение равно **19**, поэтому нам необходимо реконструировать адрес. Также мы заметим ограничение на вывод данных – **33 символа** в *field width* из **ID2**

Переопределим адрес слова **ID2** для: " **(PROGRAM MACHINES-** " , и изменим адрес ширины на **19**, и сбросим *field width* на **25**

```

define word ID2
    address letter = "(PROGRAM MACHINES -"
    address width  = 19
    field width    = 25
end define

```

Также мы должны переопределить *tape start block* следующим образом:

```

define block tape start
    "% "
    ID Jobname ( или ID PartID)
    ID2 PartID[ 7 ] ; EM =C    ## ЗаметьтеPartID[7]
    N ; xy plane ; incremental data ; absolute data ; G6 54
    N ; tool number 99
end define

```

PartID[7] выберет 7-ой символ в последовательности как точку старта, а полевая ширина ограничит вывод последовательности только описанием детали.

В результате вывод должен быть следующим:

```

%
O5222
(PROGRAM MACHINES- PORT WING STRUT OF B52***)

```

3/ Стартовые данные программы (Ввод данных программы, продолжение)

Теперь займемся другими секциями следующим образом:

```
define word ID3          define word ID4          define word ID5
  address letter = "(FOR- "  address letter = "(PART NAME- "  address letter = "(PROGRAMMER- "
  address width = 6        address width = 12        address width = 13
  field width   = 12       field width   = 6        field width   = 3
end define               end define               end define
```

В опции добавьте новые слова, заданные выше, размещая их после **ID2** и другими дополнительными словами. Также добавьте их в **word order** после **ID2**.

Измените **define block tape start** следующим образом:

```
define block tape start
  "% "
  ID Jobname ( или ID PartID) ## PartID will start at 1st Character
  ID2 PartID[ 7 ] ; EM =C      ## PartID[ 7 ] starts at the 7th Character
  ID3 PartID[33] ; EM =C      ## PartID[33] starts at the 33rd Character
  ID4 PartID[46] ; EM =C      ## PartID[46] starts at the 46th Character
  ID5 PartID[53] ; EM =C      ## PartID[53] starts at the 53rd Character
  N ; xy plane ; incremental data ; absolute data ; G6 54
  N ; tool number 99
end define
```

Что приведет к следующему:

```
%
O5222
(PROGRAM MACHINES- PORT WING STRUT OF B52***)
(FOR- BOEING INC**)
(PART NAME- 325A59)
(PROGRAMMER- AM*)
```

4/ Стартовые данные программы (Опорная точка)

Следующее чем мы займемся -это **(CNC MILL LOCATION= X, Y, Z)**, которую мы рассматриваем как **Опорную Точку** для этого упражнения.

В этом случае нужно добавить следующую строку в **define block tape start**

Примечание: кавычки " **обязательны**. Также стоит заметить что **пробел(ы)** может быть вставлен в начале последовательности текста " **(CNC.. , но не нужны в конце ..ON ="** ;

```
define block tape start
  "% "
  D Jobname ( or ID PartID)
  ID2 PartID[7] ; EM =C
  ID3 PartID[33] ; EM =C
  ID4 PartID[46] ; EM =C
  ID5 PartID[53] ; EM =C
  "(CNC MILL LOCATION = " ; X FromX ; Y FromY ; Z FromZ ; )"
  N ; xy plane ; incremental data ; absolute data ; G6 54
  N ; tool number 99
end define
```

" **X FromX** etc "., будет выводиться **Tool From**, или **Datum**, координаты.

С этими добавлениями вывод должен выглядеть как показано ниже:

```
%
O5222
(PROGRAM MACHINES- PORT WING STRUT OF B52***)
(FOR- BOEING INC--)
(PART NAME- 325A59)
(PROGRAMMER- AM*)
(CNC MILL LOCATION =X0Y0Z.5)
```

5/ Старт программы (Ввод данных программы)

Наше заключительное требование – некоторая форма даты.

Здесь нужен компромисс, т.к. мы не можем обеспечить оба требуемых вывода:

```
(DATE LAST RAN:)
(DATE LAST EDITED:)
```

Лучшее, что мы можем сделать – это одна строка, типа: (**DATE COMPILED : 10 / 01 / 99**) которая является датой программы при постпроцессировании. **Примечание:** *Европейский* формат даты - DD / MM / YY , *Американский* - MM / DD / YY.

Теперь нам требуются некоторые определенные слова и форматы (если требуется).
Включите новые слова после предыдущих добавлений, и форматы после других форматов блоков.

define word %M address letter = "(DATE COMPILED :-" address width = 18 field width = 2 end define	define word %D address letter = " / " address width = 3 field width = 2 end define	define word %Y address letter = " / " address width = 3 field width = 2 end define
--	---	---

Поскольку форматирование включено в определения, нет ни какой необходимости в **define format block**

Добавьте новые слова к порядковому списку слов:

```
word order = ( + ID2 ID3 ID4 ID5 %D %M %Y TD TN TD1 TR )
```

Затем мы изменим определение старта программы (*Американский* формат):

```
define block tape start  
    "% "  
    .... etc.  
    "(CNC MILL LOCATION =" ; X FromX ; Y FromY ; Z FromZ ; ") "  
    %M Month ; %D Day ; %Y Year ; ") "  
    N ; xy plane ; incremental data ; absolute data ; G6 54  
    N ; tool number 99  
end define
```

Что приведет к следующему выводу:

```
%  
O5222  
... etc.  
(CNC MILL LOCATION =X0Y0Z5)  
(DATE COMPILED :- 06 / 01 / 99)
```

Для вывода *Европейского* формата, выполните следующее:

Измените **define word %M** на **%D** и **%D** на **%M** и сделайте тоже самое для **define block tape start**.

```
%D Day ; %M Month ; %Y Year ; ")"
```

Получим следующий вывод:

```
%  
O5222  
(PROGRAM MACHINES- PORT WING STRUT OF B52***)  
(FOR- BOEING INC--)  
(PART NAME- 325A59)  
(PROGRAMMER- AM*)  
(CNC MILL LOCATION =X0Y0Z.5)  
(DATE COMPILED :- 01 / 06 / 99)  
G17G20G90G54  
T99  
T1  
M6  
S2800M3 (TOOL 1 Dia. .5 Tip Rad. .02)
```

Другие типы управления текстом

Другой требуемый вывод текста в программе показан ниже, где " (Двойные кавычки) необходимы:

```
N24 G99 [GEOTRA TRANS 1 (R0,R0,[VDH])]  
N25 G99 [TEXT 1 ( "LET OP --->" )]  
N26 G99 [TEXT 2 ( "****CONTROLE VDH****=",VDH)]  
N27 M00
```

Эти способы вывода могут быть достигнуты только на **UNIX** до версии DP1330, но на базе NT может использоваться следующий метод:

```
N ; " G99 [GEOTRA TRANS 1 (R0,R0,[VDH])]"  
N ; " G99 [TEXT 1 (\ "LET OP --->\")]"  
N ; " G99 [TEXT 2 (\ "****CONTROLE VDH****=\ ",VDH)]"  
N ; opt stop ( M1 0 )
```

Если **двойные кавычки** требуются в символе адреса, тогда необходимы \ (двойная левая дробь):

```
define word ID4  
address letter = " ( \ " DOUBLE QUOTES "
```

Теперь Структура Option-файла такая:

```
machine fanuc
#2
# History
# Option for North American Support
# am 15/12/98 Ref.No. 0280 Issue 1.1
# Prog.Number,Text handling,Tool data,and Date added.
# am 06/01/99 Issue 1.2 (Up dated 01/02/99)
#
#9
define word TN
  address letter = " (TOOL "
  address width = 7
  field width = 2
end define
#
define word TD
  address letter = " Dia. "
  address width = 6
  field width = 6
  scale factor = 2
end define
#
define word TR
  address letter = " Tip Rad. "
  address width = 10
  field width = 5
end define
#
define word ID2
  address letter = "(PROGRAM MACHINES- "
  address width = 19
  field width = 25
end define
#
define word ID3
  address letter = "(FOR- "
  address width = 6
  field width = 12
end define
#
define word ID4
  address letter = "(PART NAME- "
  address width = 12
  field width = 6
end define
#
define word ID5
  address letter = "(PROGRAMMER- "
  address width = 13
  field width = 3
end define
#
define word %M
```

```

    address letter = "(DATE COMPILED :-"
    address width  = 18
    field width    = 2
end define
#
define word %D
    address letter = " / "
    address width  = 3
    field width    = 2
end define
#
define word %Y
    address letter = " / "
    address width  = 3
    field width    = 2
end define
#
define format ( TD TR )
    metric formats
    decimal point = true
    decimal places = 3
    leading zeros  = false
    trailing zeros = false
    imperial formats = metric formats
end define
#
define format ( N )
    not permanent
end define
#
define format ( ID )
    address letter = "O"
end define
#
define format ( M1 M2 )
    field width = 4
end define
#
define format ( F )
    imperial formats
    decimal point = true
    decimal places = 1
end define
#
word order = ( + ID2 ID3 ID4 ID5 %M %D %Y TN TD TR )
#
define codes
    spin coolant on cw  = M1 13
    spin coolant on ccw = M1 14
    spin coolant off    = M1 05
    coolant on mist     = M1 07
    coolant on          = M1 08
    coolant on flood    = M1 57
    coolant off         = M1 09

```

```

end define
#
message output = false
integer 12      = 0
units           = input
use progid      = true
block order     = true
#
define block tape start
  "%"
  ID JobName
  ID2 PartID[7] ; EM =C
  ID3 PartID[33] ; EM =C
  ID4 PartID[46] ; EM =C
  ID5 PartID[53] ; EM =C
  "(CNC MILL LOCATION =" ; X FromX ; Y FromY ; Z FromZ ; ")"
  %M Month ; %D Day ; %Y Year ; ")"
  N ; xy plane ; compensation off ; absolute data ; G6 54
  N ; tool number 99
end define
#
define block tool change first
end define
#
define block tool change
  N ; tool number ToolNumber
  N ; change tool
  N ; spindle ToolSpeed[ToolNum] ; spindle on cw ; TN ToolNumber ;
  TD ToolRadius[ToolNum] ; TR TipRadius[ToolNum] ; EM =C
end define
#
define block move rapid
  N ; rapid ; G2 ; G3 ; G5 ; G6 ; x coord ; y coord ; z coord ; tool length ; M1 ; M2
end define
#
define block move linear
  N ; linear ; G2 ; x coord ; y coord ; z coord ; tool radius ; feedrate ; M1 ; M2
end define
#
define block tape end
  N ; tool number 99
  N ; change tool
  N ; end of tape
  "%"
end define
#
end

```

В заключение осталось обеспечить охлаждение, чем мы займемся в **Уроке 3**.

Пример Структуры Option-файла.

Урок 3

Введение

После **Урока 1**, и **Урока 2**, мы займемся управлением охлаждения, уместная часть файла программы приведена ниже:

```
T1
M6
S2400M3(TOOL 1 IS A 1/2" HOG BALL)
G0X0Y0
G43Z3.H1
M8          ## Вызо охлаждения
.....
.....
T2
M6
S2400M3(TOOL 2 IS A 1/4" CARB BALL)
G0X0Y0
G43Z3.H1
M8          ## Вызов охлаждения
.....
.....
G0Z.5
G1Z0F30.
T99
M6
M2
%
```

Управление охлаждением обиспечивается вводом и форматом требуемым клиентом. Пример приведенный ниже охватывает основы **ON** и **OFF**. (Как уже было замечено, в вышеупомянутом примере имеется **M9** – Вызов **ВЫКЛ.** охлаждения перед сменой инструмента или после окончания программы.

Есть следующие 4 типа управления охлаждением:

<u>DUCT</u>	<u>PowerMILL</u>
APT COOLANT ON	ON
APT COOLANT FLOOD	FLOOD
APT COOLANT MIST	MIST
APT COOLANT TAP	TAP
APT COOLANT OFF	OFF

(PowerMILL через " **Active Tool Path Output Form - Add Coolant Command** ", по умолчанию **OFF**).

Следует обратить внимание, что в PowerMILL выключение охлаждения можно обеспечить в конце файла обработки, но не при множественной смене инструмента.

Машинные коды охлаждения

Коды стандарта ISO используемые большинством Систем ЧПУ:

ON (вкл.) - M08
 OFF (выкл.) - M09
 MIST (туман) - M07
 с FLOOD / TAP обычно - M08.

Однако, есть разновидности управления охлаждением с помощью переопределения в Ductpost кодов в option-файле.

(Примечание: Впередистоящий " 0 " - дополнение)

Структура Охлаждения в Ductpost, по умолчанию

Если мы исследуем наш *[fanuc].dmp* файл, то к охлаждению относятся следующие части:

```
define word M1
  address letter = "M"
end define
```

```
define word M2
  address letter = "M"
end define
```

```

define format ( / G6 S T M1 M2 L P D E H O )
  address width      = 1
  field width        = 2
  exponent width     = 0
  scale factor        = 1
  tape position      = 0
  print position     = 1
  sign               = none
  not permanent
  not modal
  metric formats
  leading zeros       = false
  trailing zeros      = true
  decimal point       = false
  decimal places      = 0
  imperial formats
  leading zeros       = false
  trailing zeros      = true
  decimal point       = false
  decimal places      = 0
end define

```

define codes

```

.....
spin coolant on cw    = M1 13
spin coolant on ccw   = M1 14
spin coolant off      = M1 5
coolant on mist       = M1 7
coolant on            = M1 8
coolant on flood      = M1 8
coolant off           = M1 9

```

.....

end define

```

coolant output = ( 2 1 ) - ИЛИ -
integer 6      = 2
integer 7      = 1

```

```

spindle output = ( 1 1 1 1 )- ИЛИ -
integer 10      = 1
integer 11      = 1
integer 12      = 1
integer 13      = 1      ( See Array Data )

```

Символы слов **M1** или **M2**, с определенными кодами обработки, в этом примере являются **M1**. Способ вызова охлаждения контролируется списками " **coolant output** " и " **spindle output** ", или " **integers 6, 7, и 12, 13** ".

Для проверки функции охлаждения, советуем иметь файл обработки с соответствующими режимами.

Критерии Ductpost, по умолчанию

Для идентификации вывода мы можем сделать следующие изменения формата кодов, которые используются вместо кодов ISO, и Option-файл будет временно изменен следующим вводом:

(Этот метод может использоваться для отладки опций в других приложениях)

```
define format ( M2 )
  field width = 4
end define
#
define codes
  .....
  spin coolant on cw      = M1 13
  spin coolant on ccw     = M1 14
  spin coolant off        = M2 "SPOF"
  coolant on mist         = M2 "MIST"
  coolant on               = M2 "ON"
  coolant on flood        = M2 "FLD"
  coolant off              = M2 "OFF"
  .....
end define
```

Следует отметить, что в вышеупомянутом примере мы использовали **M2 "string definitions"** вместо целочисленных значений, следовательно кавычки увеличили полевую ширину **M2**. Кроме того **M2** используется как опциональный носитель во избежание записи по верх других значений **M1**, что может быть сделано с " **string definitions** ".

Вывод по умолчанию:

```
T1
M6
S1500M3 (TOOL 1 Dia. .394 Tip Rad. .197)
G0X.49Y.4303
G43Z1.752H1
G1F98Z1.748
.....
G0Z1.9488
T2
M6
S1500M3 (TOOL 2 Dia. .394 Tip Rad. .197)
X.49Y.4303
G43Z1.752H2MON [M08]
.....
G0Z1.9488
T4
M6
S1500M3 (TOOL 4 Dia. .394 Tip Rad. .197)
X.49Y.4303
G43Z1.752H4MFLD [M08]
.....
G0Z1.9488
T5
M6
```

S1500M3 (TOOL 5 Dia. .394 Tip Rad. .197)
X.49Y.4303
G43Z1.752H5**MMIST** [M07]

.....
G0Z1.9488
MOFF [M09]
T99
M6
M2
%

Из всего вышеупомянутого мы можем заметить, что **НЕТ** никакого вызова для Инструмента 1 , охлаждение **ВКЛ** [M08] для Инструмента 2, охлаждение **FLOOD** [M08] для Инструмента 4, охлаждение **ТУМАН** [M07] для Инструмента 5, и охлаждение **ВЫКЛ** [M09] только в конце программы.

Вызов охлаждения при **Нарезании резьбы** будет игнорироваться, т.к. в выводе РМ нет определенных ключей для Ductpost.

Охлаждение **ВЫКЛ** при смене инструмента может игнорироваться поскольку **M06** действует как **ВЫКЛ**.

Метод Опций, по умолчанию

Т.к. маловероятно использование нами " **text strings** " для кодов, и т.к. невозможно проверить последовательность текста примера, вернемся к нормальному определению кодов, за исключением охлаждения **FLOOD**, которое будет переопределено как **M57**.

```
define codes
.....
coolant on flood = M1 57
.....
end define
```

Так же мы можем удалить блок " **define format (M2)** ", который был временно добавлен.

Только в " **define block tool change** ", **добавим** следующее:

```
define block tool change
if ( word[29] = 8 or word[29] = 7 or word[29] = 57 )
  N ; M1 09
end if
N ; tool number ToolNumber
N ; change tool
N ; spindle ToolSpeed[ToolNum] ; spindle on cw ; TN ToolNumber ;
  TD ToolRadius[ToolNum] ; TR ToolRadius[ToolNum] ; EM =C
end define
#
```

if (формулировка) позволяет нам проверить, действительно ли вызов охлаждения выполнялся в начале обработки, и имеется ли вывод **ВЫКЛ** [M09].

word[29] = M1, **29-ое** определенное слово в файле **[control].dmp** используется для проверки кода охлаждения.

If (M1 07 or M1 08 or M1 57) - **true** то выводится **M09**, если - **not**, то не выводится.

Результат должен выглядеть так:

```
T1
M6
S1500M3 (TOOL 1 Dia. .394 Tip Rad. .197)
G0X.49Y.4303
G43Z1.752H1      ## Охлаждение не вызвано
G1F98Z1.748
.....
G0Z1.9488      ## ВЫКЛ. охлаждения нет
T4
M6
S1500M3 (TOOL 4 Dia. .394 Tip Rad. .197)
X.49Y.4303
G43Z1.752H4M57 ## Вызов Охлаждения Flood
.....
G0Z1.9488
M9           ## Охлаждение ВЫКЛ
T5
M6
S1500M3 (TOOL 5 Dia. .394 Tip Rad. .197)
X.49Y.4303
G43Z1.752H5M7  ## Вызов Охлаждение Туманом.
.....
G0Z1.9488
M9           ## Конец программы, охлаждение ВЫКЛ
T99
M6
M2
%
```

Этого вероятно достаточно, чтобы охватить большинство случаев, кроме тех когда клиент желает вызывать охлаждение с **Spindle ON**, т.е. **M13**. (или **M14**).

Пример ниже разрешает этот случай

Метод опции Шпинделя для Охлаждения

Для этого необходимо сделать некоторые изменения в структуре опций, поскольку вывод шпинделя всегда вызывает вывод **M3** в блоке смены инструмента.

Первое добавление – это две последовательности или соответствующие целочисленные значения, для вывода Охлаждения и Шпинделя. Обе приемлемы, но используется только один из двух форматов.

По умолчанию во встроенном **dump-файле**:

```
integer 6 = 2
integer 7 = 1
integer 10 = 2
integer 11 = 1
integer 12 = 1
integer 13 = 0
```

ИЛИ как альтернатива

```
coolant output = ( 2 1 )  
spindle output = ( 2 1 1 0 )
```

Вставьте ниже записи **block order = true**. Затем измените **define block tool change** стерев " **spindle on cw** " и вставьте **M1** , оставляя следующее:

```
N ; spindle ToolSpeed[ToolNum] ; M1 ; TN ToolNumber ;  
TD ToolRadius[ToolNum] ; TR TipRadius[ToolNum] ; EM =C
```

Если мы сохраним и протестируем изменения, то обнаружим, что **M3** отсутствует, и ничего касательно вывода охлаждения не изменилось. Для исправления этого мы должны сделать некоторые явные изменения в структуре option-файла:

Измените: **define codes**

```
spin coolant on cw = M1 13  
spin coolant on ccw = M1 14  
spin coolant off = M2 05  
coolant on mist = M2 07  
coolant on = M2 08  
coolant on flood = M2 57  
coolant off = M2 09
```

Измените: Последовательность (или соответствующие целочисленные значения

```
coolant output = ( 2 0 )  
spindle output = ( 1 1 1 1 )
```

Измените: **define block tool change**

Используя изменение **if (statement)** в

```
if ( word[30] = 8 or word[30] = 7 or word[30] = 57 or word[29] = 13 )
```

используя удаление **S ToolSpeed[ToolNum] ; M1 ;** из

```
N ; TN ToolNumber ; TD ToolRadius[ToolNum] ; TR TipRadius[ToolNum] ; EM =C
```

Измените: **define block move rapid**

используя добавление **spindle (S) ;**

```
N ; rapid ; G2 ; G3 ; G5 ; G6 ; x coord ; y coord ; z coord ; spindle ; M1 ; M2
```

Это должно производить следующий вывод, при каждом вызове охлаждения:

```
T2 M6  
G54 G90 X12.446 Y10.931 S1500 M13  
G43 Z44.5 H2  
G1 Z44.4
```

Это не на **100 %** гарантирует отсутствие ошибок, и иногда некоторые настройки option-файла нарушают это.