Formal .pmap File Format Specification

Version: 1.0.0 (Draft)  
Date: 2025-09-14  
Author: Hans Kramer  
Status: Proposal  
License: Intent; Public Domain / Creative Commons Zero (CC0)

# 1. Introduction

## 1.1 Purpose

The .pmap file format is an open-source, text-based, domain-specific data standard for defining the port map of a two-stroke engine cylinder. Its sole purpose is to enable lossless interoperability of 2D port geometry between design, simulation, and manufacturing software.

## 1.2 Scope

This specification defines the syntax and semantics of the .pmap format. It describes a 2D representation of the unwrapped cylindrical surface of a two-stroke engine cylinder. The X-axis represents the circumference; the Y-axis represents both the stroke length and the barrel length.

## 1.3 Design Philosophy

• Open-source: The format should always be available (free of charge) to those whom require it.  
• Simplicity: The format uses simple group-code and value pairs.  
• Precision: All geometric data is defined with a precision of at least 1.0e-5 units.  
• Unambiguity: The specification is strict. Parsers must not guess intent.  
• Focus: The format exclusively describes port geometry and essential metadata. It excludes presentational data (colors, text, dimensions).

# 2. File Structure

A valid .pmap file is composed of two main sections, enclosed by SECTION and ENDSEC groups, and must be terminated by EOF.

PMAP\_FILE -> HEADER\_SECTION ENTITIES\_SECTION EOF  
  
HEADER\_SECTION -> "0" "SECTION" "2" "HEADER" HEADER\_DATA "0" "ENDSEC"  
  
ENTITIES\_SECTION -> "0" "SECTION" "2" "ENTITIES" (ENTITY)\* "0" "ENDSEC"

ENTITY -> PORT\_DEFINITION | GROUP\_DEFINITION

PORT\_DEFINITION -> "0" "POLYLINE" POLYLINE\_DATA "0" "ENDPOL"

PORT\_DEFINITION -> "0" "ARC" ARC\_DATA "0" "ENDARC"

GROUP\_DEFINITION -> "0" "GROUP" (ENTITY)\* "0" "ENDGRP"

# 3. Header Section

The header section contains global, required metadata. All header variables are defined by group code 9, followed by the variable name, and then the value on subsequent lines.

## 3.1 Mandatory Header Variables

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Example Value |
| $FORMATVERSION | The version of the PMAP spec this file conforms to. (e.g., "1.0.0"). Parsers must reject unsupported major versions. | 1.0.0 |
| $MEASUREMENTUNITS | The unit system for all numerical values in the file. Case-insensitive. Valid values: MM, IN. | MM |
| $NOMINALBORE | The original design bore diameter, in the declared units. Used to calculate the nominal circumference (X-axis limit: π \* $NOMINALBORE). | 56.40 |
| $NOMINALSTROKE | The original design stroke length, in the declared units. Defines the secondary Y-axis range. | 54.50 |
| $TOTALCYLINDERLENGTH | Total cylinder length, in the declared units (must be >= stroke). Defines the primary Y-axis range. | 95.50 |
| $EXHAUST\_CENTER\_X | Exhaust center location along bore circumference. Defines the 0° point on the X-axis. | 180.0 |
| $TDC\_OFFSET | Piston position in TDC relative to the top of the cylinder, in the declared units. Negative numbers represent protrusion. | -0.25 |
|  |  |  |

## 3.2 Optional Header Variables

|  |  |  |
| --- | --- | --- |
| Variable Name | Description | Example Value |
| * **For design purposes** |  |  |
| $CURRENTBORE | If the design is for an over-bored cylinder, this defines the new target bore diameter. | 58.00 |
| $RINGGAP\_COUNT | Amount of piston ring gaps defined (0 being top, 1 = second, etc.). | 2.0 |
| $RINGGAP\_0\_ANGLE | Position of the first piston ring gap in degrees relative to exhaust center location. | 45.0 |
| $RINGGAP\_0\_WIDTH | Width of first ring gap. | 0.35 |
| * **Remarks** |  |  |
| $CREATEDBY | Person whom created this file. | Hans Kramer |
| $CREATIONDATE | File creation date in ISO 8601 format. | 2025-09-11T14:30:00Z |
| $DESCRIPTION | Human-readable description of the file. | YZ125 Rev3 Port Map |

# 4. Entities Section

The entities section contains one or more POLYLINE or ARC entities, each representing a single port or a placeholder contour.

## 4.1 The Entity

A POLYLINE is defined by a series of vertices that form a connected path.  
An ARC is defined by the center location (X/Y), it’s radius, the start & end angle.  
The data must be structured as follows:

| **Code Line** | **Value** | **Explanation** |
| --- | --- | --- |
| 0 | POLYLINE | **Start Marker.** Declares the beginning of a new polyline entity. |
|  | ARC | **Start Marker.** Declares the beginning of a new arc entity. |
|  | GROUP | **Start Marker.** Declares the beginning of a new group entity. This marks the beginning of a combination of polylines and/or arcs. |
|  | ENDPOL | **End Marker.** Declares the end of a polyline entity. |
|  | ENDARC | **End Marker.** Declares the end of a arc entity. |
|  | ENDGRP | **End Marker.** Declares the end of a group entity. |
| 8 | <layer\_name> | **Layer (Optional but Recommended).** A string to organize polylines by function (e.g., EXHAUST, TRANSFER, INTAKE, BOOST). |
| 70 | <flags> | **Flags (Mandatory).** An integer that defines the state and type of the entity using **bit-codes**. |

|  |  |  |
| --- | --- | --- |
| 4.1.1 Polylines |  |  |
| 10 | <X1> | **Vertex 1, X-Coordinate.** The first point defining the shape. |
| 20 | <Y1> | **Vertex 1, Y-Coordinate.** |
| 10 | <X2> | **Vertex 2, X-Coordinate.** The next point in the sequence. |
| 20 | <Y2> | **Vertex 2, Y-Coordinate.** |
| ... | ... | **Repeat** the 10/20 pattern for all subsequent vertices. |
| 10 | <Xn> | **Final Vertex, X-Coordinate.** Must close the loop (see validation rules). |
| 20 | <Yn> | **Final Vertex, Y-Coordinate.** |
| 4.1.2 Arcs |  |  |
| 10 | <X1> | **X-Coordinate.** The first point defining the center. |
| 20 | <Y1> | **Y-Coordinate.** |
| 40 | <R> | **Radius.** The radius of the arc from vertex X/Y. |
| 50 | <Angle> | **Starting point.** Starting point of the arc in degrees. The 0 degree point is along the positive X-axis (to the right), with angles increasing counterclockwise. |
| 51 | <Angle> | **End point.** End point of the arc in degrees. |
| 4.1.3 Groups and Nesting |  |  |
| 10 | <X1> | **X-Coordinate.** The first point in the group. |
| 20 | <Y1> | **Y-Coordinate.** |

## Group Composition

A Group (0 GROUP … 0 ENDGRP) may contain:

One or more POLYLINE entities.

One or more ARC entities.

One or more nested GROUP entities.

Groups can be recursively nested to an arbitrary depth.

## Closure Inside Groups

Each entity inside a group must still obey its own closure rules.

Entities must use their explicit end markers:

0 ENDPOL for polylines.

0 ENDARC for arcs.

Repetition of the first vertex alone is not considered valid closure.

## Nesting Semantics

Groups may contain other groups, which may themselves contain groups, without limit.

A parser must handle recursive descent until the matching ENDGRP is found.

Improperly terminated or overlapping group markers (GROUP without a matching ENDGRP) must cause the file to be rejected.

## Layer & Flag Inheritance

Groups may declare a Layer (group code 8) and Flags (group code 70).

Entities inside a group may also declare their own Layer and Flags.

Inheritance rules:

If an entity has no explicit Layer, it inherits the Layer of its parent Group.

If an entity has no explicit Flags, it inherits the Flags of its parent Group.

If both are defined, the entity’s values take precedence.

This inheritance also applies recursively through nested groups.

## Validation Rules for Groups

A Group is considered valid if:

All contained entities are valid.

All nested groups are properly closed.

A Group may mix closed and placeholder entities (e.g., some flagged 1). Placeholder handling rules still apply.

## 4.2 Layer Name (Group code 8)

An optional string to group and identify entities. Recommended values: MAIN\_EXHAUST, BRIDGED\_EXHAUST\_LEFT, BRIDGED\_EXHAUST\_RIGHT, AUXILIARY\_EXHAUST\_LEFT, AUXILIARY\_EXHAUST\_RIGHT, TRANSFER, INTAKE, BOOST, DRAFT, FIXTURE, etc.

## 4.3 Entity Flags (Group code 70)

| **Flag Value** | **Name** | **Description** |
| --- | --- | --- |
| **0** | **Standard Closure** | The classic closed shape. The parser **MUST** check that the first and last vertex (or derivatives for arcs) have identical (X, Y) coordinates. |
| **1** | **Open Contour (Placeholder)** | **This is not a valid port.** The entity is incomplete (e.g., a designer saved halfway through drawing). Must be ignored by manufacturing and simulation software. |
| **2** | **Connected to Y-min** | The entity is open to the **top** of the cylinder (typically Y=0). The parser checks that a vertex lies on this edge. (Obviously no real-world application.) |
| **4** | **Connected to Y-max** | The entity is open to the **bottom** of the cylinder (Y=$TOTALCYLINDERLENGTH). The parser checks that a vertex lies on this edge. |
| **8** | **Connected to X-seam** | The entity wraps around the seam of the cylinder (X=0 and X=Circumference). The parser understands these points are physically adjacent. |

**Combining Flags:** Flags can be added together for complex cases.

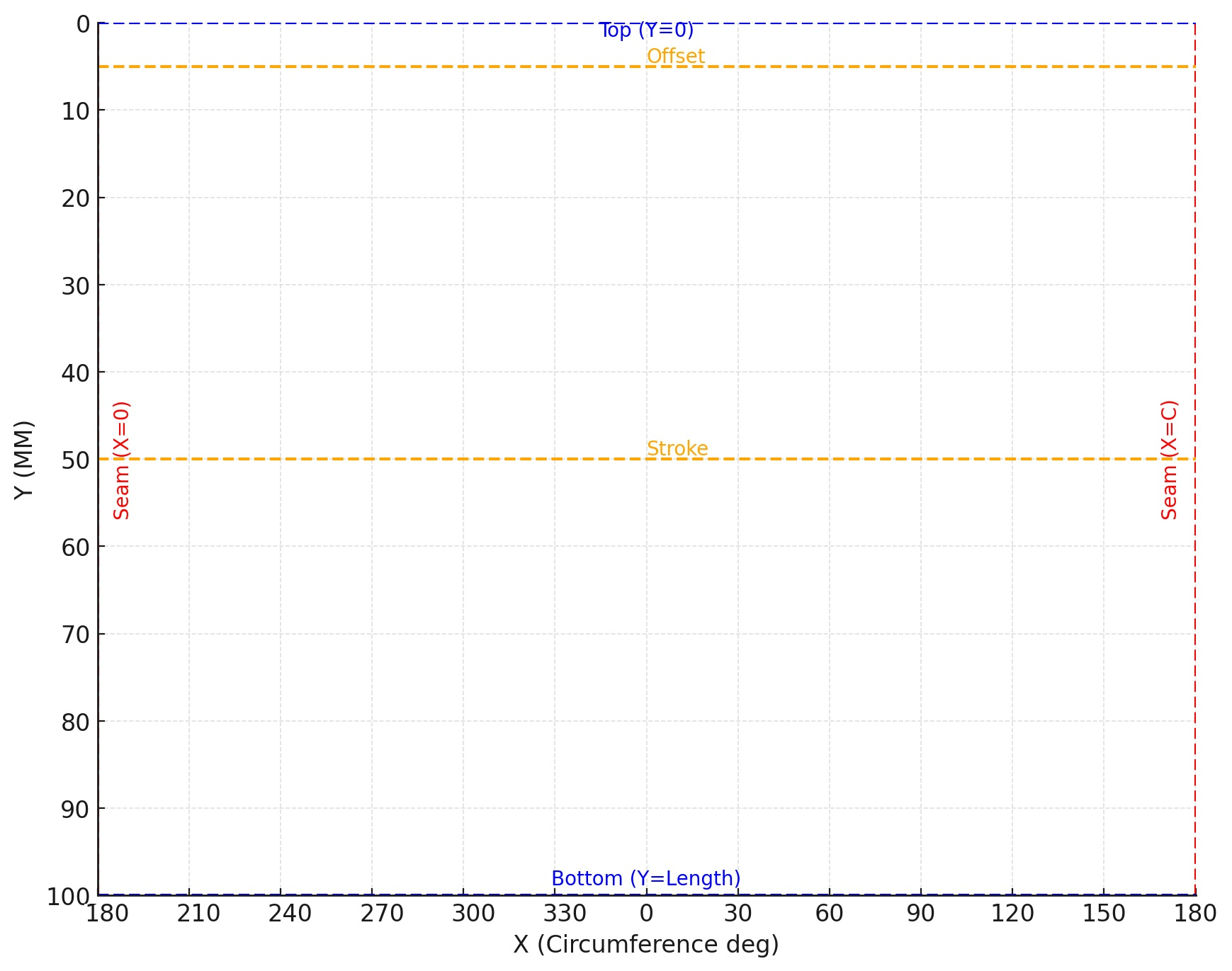
* A port open to the bottom that also wraps: 4 (Y-max) + 8 (X-seam) = 12
* The placeholder flag (1) is **mutually exclusive** and should never be combined with closure flags. This means uneven entity flags will never be allowed for future versions.

# 5. Coordinate System

**All coordinates and dimensions within the .pmap format are defined in the linear units declared in the $MEASUREMENTUNITS header variable (MM or IN).** This choice is fundamental to ensuring consistent precision and interoperability across different cylinder designs. Using linear units (e.g., millimeters) for both the circumferential (X) and axial (Y) axes, rather than expressing the X-axis in angular degrees, provides a critical advantage: **it guarantees uniform resolution regardless of the cylinder's bore diameter.**

For example, on a large-bore cylinder, a single degree of circumference represents a much larger linear distance than on a small-bore cylinder. Defining ports using angular coordinates would therefore result in a loss of precision and manufacturing consistency for larger engines. By using linear units, a vertex positioned at X=1.5 always represents the same physical distance from X=1.4 on any cylinder, ensuring port geometry is defined with consistent, high fidelity.

**X-Axis:** Represents the circumference of the cylinder. The valid range is from 0.0 to C, where C = π \* $NOMINALBORE and the coordinate 0.0 defines the intersection of the top of the cylinder barrel and the $EXHAUST\_CENTER\_X.  
**Y-Axis:** Represents the cylinder length along the cylinder's axis. The value 0.0 typically represents the top of the barrel (cylinder deck). The value $TOTALCYLINDERLENGTH represents the bottom of the cylinder. The points (0.0, Y) and (C, Y) are adjacent, representing the same physical seam on the cylinder.



# 6. Validation Rules

1. A conforming parser MUST implement the following validation checks:  
   • **File Structure:** The file must contain exactly one HEADER and one ENTITIES section, in that order, terminated by EOF.  
   • **Mandatory Header Vars:** The parser must reject the file if any mandatory header variable ($FORMATVERSION, $MEASUREMENTUNITS, $NOMINALBORE, $NOMINALSTROKE, $TOTALCYLINDERLENGTH, $EXHAUST\_CENTER\_X, $TDC\_OFFSET) is missing or has an invalid value.  
   • **Entity Closure:** For an entity to be valid (i.e., not a placeholder), it must be "closed". The parser must check this based on its flags:
   * Flag 0: The first and last vertex (or derivatives for arcs) must be identical (within a tolerance of 1.0e-5 units).
   * Flag 2: The first or last vertex (or derivatives for arcs) must have a Y-coordinate <= 0.0 (within tolerance).
   * Flag 4: The first or last vertex (or derivatives for arcs) must have a Y-coordinate >= $TOTALCYLINDERLENGTH (within tolerance).
   * Flag 8: The entity is defined as closed via the cylindrical seam. No specific vertex check is required, but the entity should meaningfully use the seam (e.g., have vertices near X=0 and X=C).

.  
• **Placeholder Handling:** Entities with flag 1 must be parsed but flagged to the user as incomplete. They should be ignored for simulation or manufacturing purposes  
• Use tolerance of 1.0e-5 units for comparisons.

# 7. Example Files Example 1: Minimal Valid File

0  
SECTION  
2  
HEADER  
9  
$FORMATVERSION  
1.0.0  
9  
$MEASUREMENTUNITS  
MM  
9  
$NOMINALBORE  
56.40  
9  
$NOMINALSTROKE  
54.50  
9  
$TOTALCYLINDERLENGTH  
95.50  
9  
$EXHAUST\_CENTER\_X  
180.0  
9  
$TDC\_OFFSET  
-0.25  
0  
ENDSEC  
0  
SECTION  
2  
ENTITIES  
0  
POLYLINE  
8  
EXHAUST  
70  
0  
10  
25.0  
20  
30.0  
10  
35.0  
20  
25.0  
10  
25.0  
20  
30.0  
0  
ENDSEC  
0  
EOF

## Example 2: Edge Connection (Y-max flag) 0 SECTION 2 HEADER 9 $FORMATVERSION 1.0.0 9 $MEASUREMENTUNITS MM 9 $NOMINALBORE 56.40 9 $NOMINALSTROKE 54.50 9 $TOTALCYLINDERLENGTH 95.50 9 $EXHAUST\_CENTER\_X 180.0 9 $TDC\_OFFSET -0.25 0 ENDSEC 0 SECTION 2 ENTITIES 0 POLYLINE 8 TRANSFER 70 4 10 10.0 20 54.5 10 20.0 20 50.0 10 10.0 20 50.0 10 10.0 20 54.5 0 ENDSEC 0 EOF

(Polyline connected to bottom edge: flag 4 = Y-max.)

## Example 3: Arc Entity

0  
SECTION  
2  
HEADER  
9  
$FORMATVERSION  
1.0.0  
9  
$MEASUREMENTUNITS  
MM  
9  
$NOMINALBORE  
56.40  
9  
$NOMINALSTROKE  
54.50  
9  
$TOTALCYLINDERLENGTH  
95.50  
9  
$EXHAUST\_CENTER\_X  
180.0  
9  
$TDC\_OFFSET  
-0.25  
0  
ENDSEC  
0  
SECTION  
2  
ENTITIES  
0  
ARC  
8  
BOOST  
70  
0  
10  
28.2  
20  
20.0  
40  
6.5  
50  
0.0  
51  
180.0  
0  
ENDSEC  
0  
EOF

Example 4: Group Entity  
0

SECTION

2

HEADER

... [Header Data] ...

0

ENDSEC

0

SECTION

2

ENTITIES

0

GROUP

8

BRIDGED\_EXHAUST

70

0

0

POLYLINE

8

BRIDGED\_EXHAUST\_LEFT

70

0

10

90.0

20

40.0

10

100.0

20

35.0

10

100.0

20

25.0

10

90.0

20

40.0

0

ENDPOL

0

POLYLINE

8

BRIDGED\_EXHAUST\_RIGHT

70

0

10

110.0

20

35.0

10

120.0

20

40.0

10

110.0

20

25.0

10

110.0

20

35.0

0

ENDPOL

0

ENDGRP

0

ENDSEC

0

EOF

Example 5: Seam-Wrapping Port (flag 8)  
0  
SECTION  
2  
HEADER  
9  
$FORMATVERSION  
1.0.0  
9  
$MEASUREMENTUNITS  
MM  
9  
$NOMINALBORE  
56.40  
9  
$NOMINALSTROKE  
54.50  
9  
$TOTALCYLINDERLENGTH  
95.50  
9  
$EXHAUST\_CENTER\_X  
180.0  
9  
$TDC\_OFFSET  
-0.25  
0  
ENDSEC  
0  
SECTION  
2  
ENTITIES  
0  
POLYLINE  
8  
INTAKE  
70  
8  
10  
0.0  
20  
20.0  
10  
5.0  
20  
35.0  
10  
177.3  
20  
35.0  
10  
177.3  
20  
20.0  
10  
0.0  
20  
20.0  
0  
ENDSEC  
0  
EOF

## Appendix A: Versioning Policy

**Major Version Increment (X.0.0):** Breaking changes. Parsers for older versions cannot read the new format.  
**Minor Version Increment (1.X.0):** Backwards-compatible additions (e.g., new optional header variables). Parsers for older versions must ignore new group codes they don't understand.  
**Patch Version Increment (1.0.X):** Backwards-compatible bug fixes or clarifications to the spec text.

**Disclaimer:** This is a draft specification and is subject to change based on community feedback and implementation experience.