`Profile` object format

The exact specification of a part is called a `Profile` and is modeled as a list of `Feature`s. A `Feature` can be a straight segment (sometimes called a 'leg'), a bend (aka angle), or a hem. A natural consequence of this model is that every `Profile` will have an odd number of `Features` in it's list, and the first and last `Feature` will always be legs. For example, a 3" U shaped part (two 90° bends and 3" on all sides) would be represented as five `Features`. They would be `[Segment(3"), Angle(90°), Segment(3"), Angle(90°), Segment(3")]`.

Hems are treated much like an angle with a short leg. The 'length' of the hem is just another segment. For example, a flat 3" part with ½" closed hems on each end is represented as `[Segment(½"), ClosedHem(), Segment(3"), ClosedHem(), Segment(½")]`.

Features

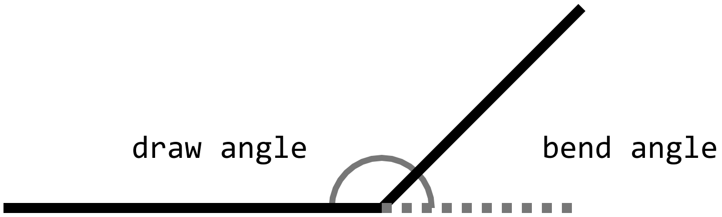
All feature objects have a `type` member. This must be set to one of `ClosedHem`, `OpenHem`, `TearDropHem`, `Angle`, `Straight`, or `Radius`.

Angle

The simple bends have `type` set as `Angle`.

```
{
    type: "Angle",
    angle: 45.5,
}
```

The `angle` member is the amount to bend in degrees. It can be a floating point number. The sign determines which way to bend. `-` for right (clockwise), `+` for left (counter clockwise). It refers to the *bend angle*, as noted in this diagram.



Hems

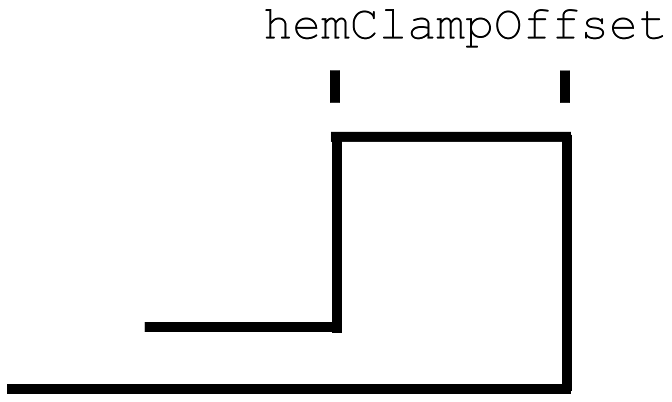
Hems have the type set as one of `OpenHem` , `ClosedHem` , or `TearDropHem` . All hems have `hemDirection` that defines to which side of the material the hem is bent. "Positive" means to the left (clockwise) and "Negative" means to the right (counterclockwise). For a closed hem, all you need is this:

```
{
    type: "ClosedHem",
    hemDirection: Positive // <-- or "Negative"
}
```

An open hem gets a `hemHeight` .

```
{
    type: "OpenHem",
    hemDirection: Positive,
    hemHeight: 0.0,
}
```

A tear drop hem requires more information.



```
{
    "type": "TearDropHem",
    "hemDirection": Positive,
    "hemClampOffset": 0.25
}
```

`hemClampOffset` is the length to leave outside the clamp during the clamp close.

Radius Bends

Radius bends are features that create a long, curving arc. At the machine a radius bend is performed as a series of multiple shallow bends separated by short equilength segments. Radius bends have a concept called 'quality'. The higher the quality means more shallow bends are used to make the radius bend, with the goal of a smoother curve.

```
{
    type: "Radius",
    angle: 75,
    radius: 3.0,
    radiusQuality: `Fine`
}
```

Picture an imaginary circle that the material is being folded around. `radius` is the distance from the center of that circle to the material. `angle` defines how far along the circumference to fold. As mentioned above, Pathfinder projects a radius bend into a series of shallow angles between short segments. The more shallow and short, the better the approximation of the curve. This is referred to as the `radiusQuality` , and can be one of `Coarse` , `Medium` , or

`Fine`. `Fine` will use many shallow bends, `Coarse` will use few. `Medium` is somewhere in between.

Segments (legs)

The straight portions of a profile are represented as `Straight` s. Every profile will start and end with a `Straight`.

```
{
    type: "Straight",
    length: 7.5
}
```

`Straight` s need only a length. The units on the length are always inches.

Profile

The list of features detailed above is attached to a parent `Profile` object. This object also has a variety of members that define various part level concerns like `Name`, `Description`, etc.

```
    "profileName": "29 5V Cleat",
    "description": "29 5V Cleat",
    "category": "All Cleats",
    "subCategory": "Small",
    "paintedSide": "Positive",
    "owningCatalogId": 72,
    "features": [<...>]
```

`ProfileName` and `Description` are free form string values. You are free to submit whatever you want.

`Category`, and `SubCategory` are also free form string values, and are used as a grouping mechanism in the Edge web portal. It might help to think of them as a folder structure, albeit one that only goes two levels deep.

`PaintedSide` denotes which side of the part is painted, if at all. It can be one of `Positive`, `Negative`, or `None`. For painted parts this value is interpreted as meaning the *left* (`Positive`) or *right* (`Negative`) side of the *first* segment in the `Features` list.

Catalog is a more formal concept than Category/SubCategory. It is the parent object of a set of profiles. The catalog is the group at which Edge applies permissions and machine sync configuration. For example, some catalogs might only be accessible by an administrator. Another example is whether a profile is sent to a Pathfinder machine is determined by the sync setting of it's parent catalog. Set the catalog of a profile via the `OwningCatalogId`. Get this id from the list of catalogs returned by the appropriate catalog list method. This value is optional. If not sent, the profile will be assigned to a default catalog.

`Features` is the list of objects that make up the shape of the profile.

Example

Here is a complete example of a submission to create a new profile. This profile is a "V" shape with a hem on each leg. The left hem is closed, the right is open and both hems are folded towards the outside of the V. The part is painted, and the painted side is inside the V.

```
    "ProfileName": "V Shaped Thingy",
    "Description": "A part in the shape of a V. For demonstration purposes only.",
    "Category": "Demo parts",
    "SubCategory": "Small",
    "PaintedSide": "Positive",
    "Features": [
      {
        type: 'Straight',
        length: 0.5,
      },
      {
        type: 'ClosedHem',
        hemDirection: "Negative"
      },
      {
        type: 'Straight',
        length: 3.0,
      },
      {
        type: 'Angle',
        angle: 135,
      },
      {
        type: 'Straight',
        length: 3.0,
      },
      {
        type: 'OpenHem',
        hemDirection: "Negative"
      },
      {
        type: 'Straight',
        length: 0.5,
      }
    ]
```