# XDFL User Guide

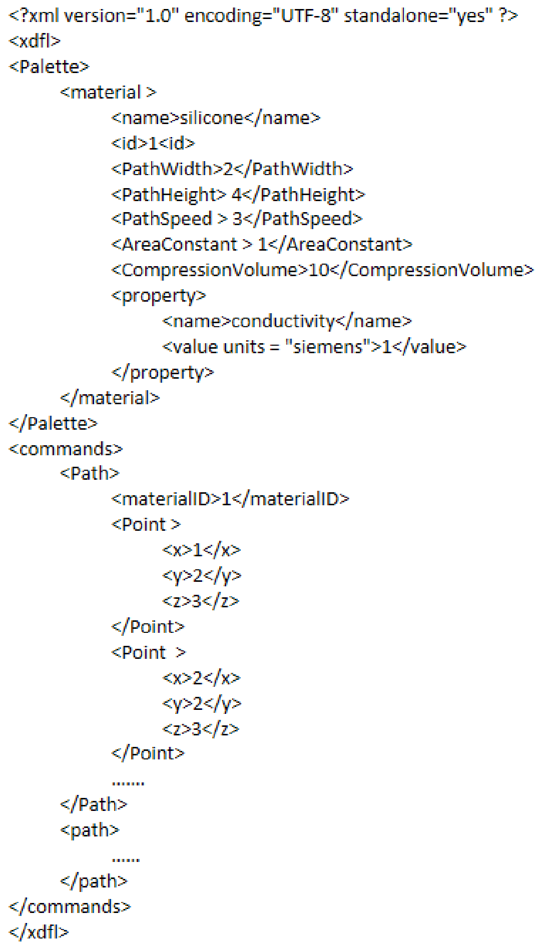
eXtensible Digital Fabrication Language

## Overview

XDFL is the command language for the Fab@Home system. It is designed to allow for quick modification of calibrations. XDFL is generated from STL files by the SeraphStudio$^{TM}$ program. The SeraphStudio program uses information contained in tool script files to process the geometry into the XDFL. The SeraphPrint$^{tm}$ system converts the XDFL commands into motions the machine will execute. The code that converts the commands into movements in contained in the printer configuration files. Each bay for each tool head contains a JavaScript file that tells the system how to respond to XDFL commands. The XML structure allows the system to be edited by hand and by computer programs. Almost all programming languages have readily accessible libraries for reading and writing XML documents. The SeraphPrint program is case insensitive so it is possible to use any case when developing an XDFL file.

## Structure

The first line in an XDFL file is an XML declaration. It sets the XML version and encoding standard for the characters. The seconds line in the root tag of the file. The root tag is always “xdfl”. The xdfl tag has two children, the “pallete” tag and “commands” tag. The pallet contains the calibration information for each material. The commands contain the deposition paths and the non-deposition paths for a print job.  The system will execute the commands in order, so positioning of the commands matters for the file.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<xdfl>
<Palette>
        <material >
                <name>silicone</name>
                <id>1<id>
                <PathWidth>2</PathWidth>
                <PathHeight> 4</PathHeight>
                <PathSpeed > 3</PathSpeed>
                <AreaConstant > 1</AreaConstant>
                <CompressionVolume>10</CompressionVolume>
                <property>
                        <name>conductivity</name>
                        <value units = "siemens">1</value>
                </property>
        </material>
</Palette>
<commands>
        <Path>
                <materialID>1</materialID>
                <Point >
                        <x>1</x>
                        <y>2</y>
                        <z>3</z>
                </Point>
                <Point  >
                        <x>2</x>
                        <y>2</y>
                        <z>3</z>
                </Point>
                .......
        </Path>
        <path>
                ......
        </path>
</commands>
</xdfl>
```

Figure 1: The Layout of a XDFL File.

## Pallet

The Pallet contains all of the information needed for each material to ensure the machine can extrude the material. The opening tag is a “palletE” tag. Each material calibration is given its own tag “material” This tag contains and id tag and a name tag. The name lists the materials name for use by human identification. The id tag contains a locally unique ID for the calibration. The paths will reference the id number.

The system operates on the basis of balancing the volumetric flow of material from a tool head. Each path has a height, width and cross sectional area. These paths are traversed at set speeds. This leads to a required volumetric flow rate (Q). The SeraphPrint system extrudes material from the tool head to match the material required for the path. Since no tool head is without the need for priming, and some materials are compressible, the material also need to list a compression volume. This is the volume the system will “extrude” before a path and suckback after a path. The system uses mm for distance, seconds for time, and mm$^3$ for volumes

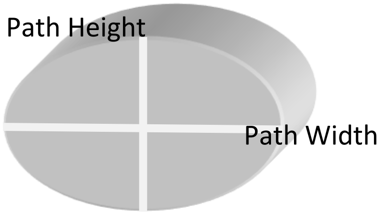
As a result of these requirements, each material must have the following tags:

- name
- id
- PathWidth
- PathHeight
- PathSpeed
- AreaConstant
- CompressionVolume

$$\frac{Vol}{Sec} = PathWidth * PathHieght * AreaConstant * PathSpeed$$

$$Cross\ Section = AreaConstant * PathWidth * Path\ Height$$

(a)



(b)

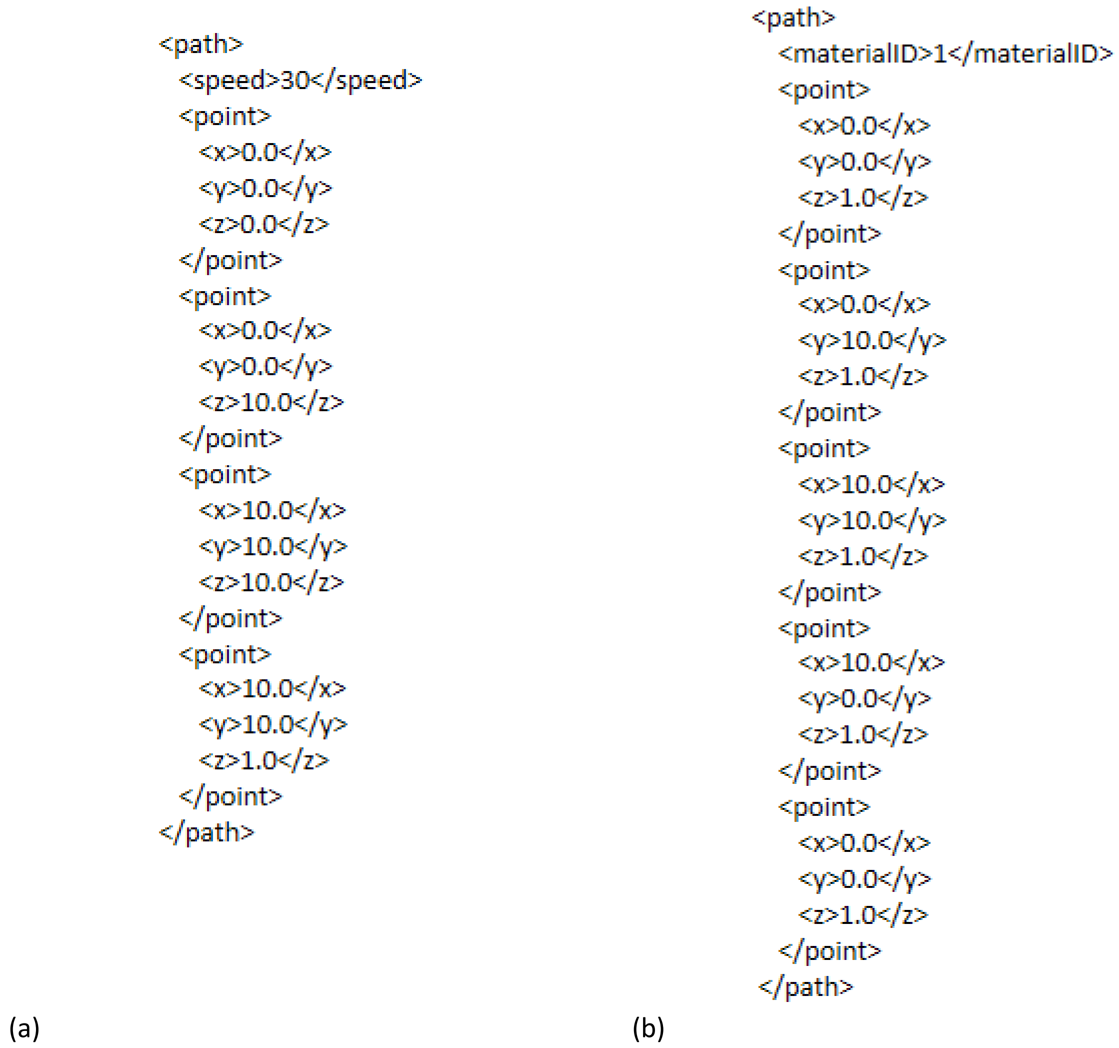
Figure 2: The equations governing XDFL path flow rates (a), and a diagram of a given path (b)

## Commands

The commands section is a list of paths the system will move along and voxels that the system will deposit. Paths are the most common form of deposition and the only form generated by SeraphStudio.

### Paths

These paths are executed in the order in which they appear in the file. There are two types of paths, deposition paths, and movement paths. Both paths contain a series of points that define the line segments the machine will move along. Each point contains a tag of x,y and z  which defines the location in space of the point. Deposition paths contain a “materialid” tag that has an integer which references the id of a material in the pallet. This reference defines the speed along which the path will be moved and tells the system to extrude material. Movement paths contain a “speed” tag that defines how quickly the system will move along the path. Movement paths are often used to create a movement clearance to prevent the tool head from intersecting the printed object when moving from section to section.



```
<path>
  <speed>30</speed>
  <point>
    <x>0.0</x>
    <y>0.0</y>
    <z>0.0</z>
  </point>
  <point>
    <x>0.0</x>
    <y>0.0</y>
    <z>10.0</z>
  </point>
  <point>
    <x>10.0</x>
    <y>10.0</y>
    <z>10.0</z>
  </point>
  <point>
    <x>10.0</x>
    <y>10.0</y>
    <z>1.0</z>
  </point>
</path>
```

(a)

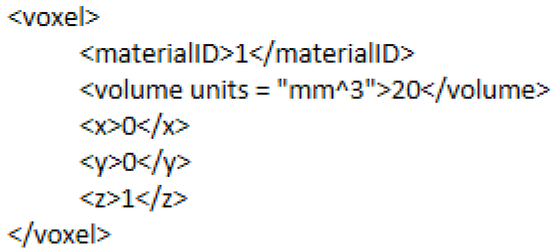
```
<path>
  <materialID>1</materialID>
  <point>
    <x>0.0</x>
    <y>0.0</y>
    <z>1.0</z>
  </point>
  <point>
    <x>0.0</x>
    <y>10.0</y>
    <z>1.0</z>
  </point>
  <point>
    <x>10.0</x>
    <y>10.0</y>
    <z>1.0</z>
  </point>
  <point>
    <x>10.0</x>
    <y>0.0</y>
    <z>1.0</z>
  </point>
  <point>
    <x>0.0</x>
    <y>0.0</y>
    <z>1.0</z>
  </point>
</path>
```

(b)

Figure 3: The paths are the children of the commands tag. There are two types. A movement path (a) and a deposition path (b)

## Voxels

Voxels extrude a single material at a point. They are children of the commands node and list the location, volume and material to be extruded.



```
<voxel>
        <materialID>1</materialID>
        <volume units = "mm^3">20</volume>
        <x>0</x>
        <y>0</y>
        <z>1</z>
</voxel>
```

Figure 4: a Voxel tag that extrudes 20 mm$^3$ of material at 0,0,1