

Technical Disclosure Commons

Defensive Publications Series

July 18, 2019

HANDLING MULTIPLE PROFILE RESOURCES IN OPEN MATERIALS

HP INC

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

INC, HP, "HANDLING MULTIPLE PROFILE RESOURCES IN OPEN MATERIALS", Technical Disclosure Commons, (July 18, 2019)

https://www.tdcommons.org/dpubs_series/2360



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Handling Multiple Profile Resources in Open Materials

Open Materials Framework allows that 3rd party companies can configure all the 3D printing process parameters to customize the use of their own powder material with some 3D printers. This process usually requires handling different kinds of resources. In some cases, it is also required to allow having multiple resources of the same kind for supporting different scenarios. In this invention disclosure we propose a mechanism, based on a custom file format and a REST web API, which provides the mechanisms required for profile resources handling. Depending on the resource type exposed from Open Materials Framework and settings; the REST API client can create, and delete the resource from the collection, and can select or eliminate the default resource for the device.

The configurations required for developing a new material require of different kinds of resources. Some examples are:

- *Build process*: Describes the actions that are executed by the FW engine during the phase being configured. For example, how the layers are generated while in printing phase or how the content is cooled down during the cooling phase.
- *Fluids pipeline*: Set of writing system transformations that are executed over the content voxels (and other properties like distances or color) to compute the halftone levels of fluids to be placed.
- *Dimensional profile*: Dimensional calibration which characterizes how the parts shrink during the printing and cooling process.
- *Color maps*: Conversion of input RGB color to contone levels of fluids for every colorant.

Open material Resource Package (ORP) file format

These resources usually consist on a set of files and metadata. For that, purpose our mechanism defines an OPC-based (Open Packaging Convention) [1] file format which includes a metadata XML file which allows the FW to identify the kind of resource. Mainly there are two files used for identifying the content, which name should match exactly the one defined:

- */_rels/.rels*: Relationships file of the root package. This file defines the content of the OPC package. All the files inside the ZIP package which are not defined here (or in a part-specific relationships file) will be ignored. For the ORP, it should define the relationships of the metadata file, which should be of type: <http://schemas.hp.com/3D/2019/omes/package> . An example of a root relationships file can be:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"
  <Relationship Id="rel0" Target="/OMES/profile.omes" Type="http://schemas.hp.com/3D/2019/omes/package"/>
</Relationships>
```

- *[Content_Types].xml*: Defines the type of content on files of a given extension. For the case of the ORP, in addition to identify the ".rels" file from the OPC package, it is also required to identify the OMES metadata with a content type: *application/vnd.hp.package.metadata+xml*. It is worth to mention that while the extension of the metadata file can be configured to any

value by producer, it is recommended to use OMES. An example of a content types file can be:

```
<?xml version="1.0"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
  <Default Extension="rels" ContentType="application/vnd.openxmlformats-package.relationships+xml" />
  <Default Extension="omes" ContentType="application/vnd.hp.package.metadata+xml" />
</Types>
```

Then, on top of the files defined from OPC, the proposed base ORP file includes a metadata file from which the printer FW will get all the required information for a resource package. It means that it is used for identifying the name, identifier assigned by the producer and the type of resource, so that it can be handled properly for the purpose of the package. It provides the following information:

- *ExternalID*: UUID assigned by the producer of the file. Once the resource is added to OMES it will get assigned another ID in the Printer FW (Internal ID).
- *Name*: The name assigned to the resource. This field may be localized for some types of resources (see section 1.4).
- *Type*: Defines the type of resource. This is used by the printer FW to determine how this resource file should be processed.
- *Description*: A free field assigned describing the content of the resource. This field may be localized for some types of resources (see section 1.4).
- *MaterialID*: The material identifier to which the resource is applied, as reported by the printer Web Service (e.g. 1000 for PA12). If not provided the resource is considered common for all the materials.
- *ProfileID*: The profile identifier to which the resource is applied, as reported by the printer Web Service (e.g. 1000.0.0 for PA12-Balanced). If not provided, the resource is considered common for all the profiles in the material.
- *MaterialRevision*: The revision of material to which this resource is valid. If not provided it is considered valid for any revision of material.
- *RequiredSkuFeature*: The SKU feature required to be enabled in the printer in order to be able to use this resource (e.g. color maps require that printer has color feature).
- *UserAccess*: The level of access for a user. If not specified it is considered as NoAccess.
- *UserAccessUi*: The level of access for a user in the UI. If not specified it is considered as NoAccess.
- *PartnerAccess*: The level of access for a partner. If not specified it is considered as NoAccess.
- *PartnerAccessUi*: The level of access for a partner in the UI. If not specified it is considered as NoAccess.

Dimensional Profiles case

While the proposed solution will be used for any kind of resource to be allowed to be updated in the printer from a given SW (e.g. Open Materials Framework tools), we describe the proposed solution using a concrete case from a real need: the Dimensional Profiles. Dimensional calibrations produce dimensional profiles which need to be uploaded to the printer from an external SW. It is also required that the printer is capable to manage a list of dimensional profiles because different calibrations may be generated for different scenarios: For instance, it may be generated a HW specific profile for hollow

parts, a different one for solid parts and another one for large parts. Then, the SW application has to be capable of selecting which dimensional profile from the dimensional profiles collection available in the printer is used for every job. The printer can be configured to select the default dimensional profile or to unselect/modify the default dimensional profile is used for the job by the SW application.

REST API for Dimensional Profiles

Dimensional profiles are exposed in the materials endpoint of the printer. When loading the material endpoint, for every printing profile in the material, a section containing the profile resources are shown. In the case that the printing profile does not have any dimensional profile resource, then it publishes a link to an HTTP POST method for creating a dimensional profile (see [Figure 1](#)).

Figure 1. POST endpoint on a printing profile without any dimensional profile available

In the case that there exists any dimensional profile, then several endpoints for accessing (HTTP GET) or removing (HTTP DELETE) the resource are published (see [Figure 2](#)). Additionally, at printing profile level, an HTTP POST and HTTP PATCH endpoints are published in order to create, set or modify the default dimensional profile for a given printing profile. When a new dimensional profile is uploaded, the SW client can determine whether the resource is being created is set as default one or not. It is also worth to mention that in the case that it is required to not use any dimensional profile for a job, it is possible to unselect the dimensional profile from being the default profile resource. As consequence of unselecting the profile, then no dimensional profile will be applied for processing the job. Also, if the dimensional profile from the device is Generic profile, which is installed in factory, then it will not be possible for SW client to perform HTTP DELETE for that resource. (see [Figure 3](#)).

Figure 2. Endpoints published for a printing profile with one dimensional profile available

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<hpmjfi:ProfileResource>
  <hpmjfi:Name hasStringResource="true">ProfileResourceDIMENSIONALPROFILE</hpmjfi:Name>
  <hpmjfi:Visible>true</hpmjfi:Visible>
  <hpmjfi:ReadOnly>true</hpmjfi:ReadOnly>
  <hpmjfi:links>
    <hpmjfi:link method="GET" rel="dimensionalprofile" uri="https://15.83.11.105/HP-HJF/PI/1.3/dimensionalprofiles/1000/1000.0.0/1000.0.0.0/" />
    <hpmjfi:link method="GET" rel="dimensionalprofilemetadata" uri="https://15.83.11.105/HP-HJF/PI/1.3/dimensionalprofiles/1000/1000.0.0/1000.0.0.0.0/metadata/" />
  </hpmjfi:links>
  <hpmjfi:Default>true</hpmjfi:Default>
  <hpmjfi:ID>1000.0.0.0.0</hpmjfi:ID>
  <hpmjfi:ResourceName hasStringResource="true">Zonal scaling PA12 Balanced</hpmjfi:ResourceName>
</hpmjfi:ProfileResource>
</?xml>

```

Figure 3. Endpoints published for a printing profile without HTTP DELETE method

The proposed mechanism has the following advantages:

1. We propose an OPC-based format which, being self-contained, allows the easy inter-change of resources from the SW to the printer.
2. The OPC-based format will be detectable as in the Open Material Framework has to be allowed to be modified by third parties.
3. The OPC-based format allows the printer to properly identify the kind of resource contained in the package and, based on the information read from the metadata file, handle its content in the proper way.
4. The REST web API allows that any SW client may be capable of managing the resource. This makes it feasible having, for instance, Terminus being capable of uploading Dimensional Profiles depending on the specific need of the Customer. The REST API also allows to upload the summary of the job data to the HP cloud to be analyzed. The summary contains the dimensional profile information of the job, the scaling and offset factors that were applied to the parts. and a different tool being capable of uploading Color Maps.
5. By allowing to configure the default resource for every kind of resource, the external SW may control which resource is used in every scenario.

References

- [1] Open Packaging Conventions. https://en.wikipedia.org/wiki/Open_Packaging_Conventions

Disclosed by Sergio Gonzales, Bengi Saglam and Jaime Machado Neto, HP Inc.