3MF Test Specification

Specification & Reference Guide



|  |  |
| --- | --- |
| **Version** | 0.92 |
| **Status** | Draft |

THESE MATERIALS ARE PROVIDED “AS IS.” The contributors expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL ANY MEMBER BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER MEMBER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Revisions:

|  |  |  |
| --- | --- | --- |
| **Version** | **Changes** | **Date/Editor** |
| 0.50 | Initial Draft | JZ – 8/8/18 |
| 0.51 | Minor edits | JZ – 8/12/18 |
| 0.52 | Minor Edits | JZ – 8/21/18 |
| 0.54 | Minor Edits | JZ – 9/11/18 |
| 0.90 | Changed “add” to “mix” for blendmethods | JZ – 9/18/18 |
| 0.91 | Minor Edits | JZ – 9/18/18 |
| 0.92 | Minor Edits | Jz – 10/12/18 |

Contents

[Revisions: 2](#_Toc521558120)

[1 Introduction 4](#_Toc521558121)

[2 Terms and Acronyms 4](#_Toc521558122)

[3 Scope 4](#_Toc521558123)

[4 Test Suite Organization 5](#_Toc521558124)

[4.1 Test Case Numbering 5](#_Toc521558125)

[4.2 Test Case Template 6](#_Toc521558126)

[5 Test Case Definitions 7](#_Toc521558127)

[5.1 Positive OPC Test Cases 7](#_Toc521558128)

[5.2 Negative OPC Test Cases 9](#_Toc521558129)

[5.3 Positive 3MF Core Test Cases 11](#_Toc521558130)

[*5.4* Negative 3MF Core Test Cases 20](#_Toc521558131)

[5.5 Positive 3MF Material Extension Test Cases 27](#_Toc521558132)

[5.6 Negative Material Extension Test Cases 45](#_Toc521558133)

[*5.7* Positive Production Extension Test Cases 48](#_Toc521558134)

[5.8 Negative Production Extension Test Cases 53](#_Toc521558135)

[5.9 Miscellaneous 3MF Test Cases 55](#_Toc521558136)

[5.10 Positive 3MF Slice Extension Test Cases 60](#_Toc521558137)

[*5.11* Negative Slice Extension Test Cases 65](#_Toc521558138)

[Appendix A - Test Object Library 67](#_Toc521558139)

[Appendix B – Color and Texture Tables 74](#_Toc521558140)

[Appendix C - Test Case to Test Suite Mapping 81](#_Toc521558141)

# Introduction

The 3D Manufacturing Format, or 3MF, describes the set of conventions for the use of XML and other widely available technologies to describe the content and appearance of one or more 3D models. It is written for developers who are building systems to process 3MF content.

A primary goal of this specification is to ensure the interoperability of independently created software and hardware systems that produce or consume 3MF content. This specification defines a set of test cases that can be used to validate 3MF consumer and producer implementations.

# Terms and Acronyms

The following terms and abbreviations are used in this document:

|  |  |
| --- | --- |
| **Term** | **Description** |
| 3MF | 3D Manufacturing Format |
| OPC | Open Packaging Conventions |
| DUT | Device Under Test |
|  |  |

# Scope

Execution of a 3MF test suite will provide a robust characterization of the Device Under Test’s (DUT) behavior by providing a wide variety of both valid and invalid 3MF content based upon each of the conformance statements and schemas in the specifications that define 3MF. Collectively, running all the test cases developed as part of this test suite will provide the following test coverage:

* The DUT can successfully process 3MF files that include valid permutations of mandatory characteristics defined in the supported 3MF XML schemas
* The DUT can successfully process 3MF files that both include and exclude valid permutations of optional characteristics defined in the supported 3MF XML schemas
* The DUT can successfully process 3MF files that conform to valid permutations of the mandatory and optional OPC, Core, Material, Slice, and Production conformance requirements defined in 3MF technical specifications
* The DUT can gracefully handle invalid 3MF file content

Test cases are based on the following versions of the 3MF Specifications:

* *Office Open XML File Formats – Open Packaging Conventions – December 2012*
* *3MF Core Specification – Version 1.2.2*
* *3MF Materials and Properties Extension – Version 1.2*
* *3MF Production Specification – Version 1.1.1*
* *3MF Slice Specification – Version 1.1*

# Test Suite Organization

Test cases defined in this specification many support one or more 3MF extensions and may belong to one or more test suites. The table below documents the six supported test suites.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Suite Name** | **Core** | **3MF Extensions** | | |
| **Slice** | **Production** | **Material** |
| **Test Suite 1** | • | • | • |  |
| **Test Suite 2** | • |  | • | • |
| **Test Suite 3** | • |  |  |  |
| **Test Suite 4** | • | • |  |  |
| **Test Suite 5** | • |  | • |  |
| **Test Suite 6** | • |  |  | • |

**Note that Appendix C provides a mapping as to which test cases are contained in which test suites.**

## Test Case Numbering

Test Cases will use the following syntax for numbering: **U\_VWX\_YYYY\_ZZ**

* U: P=positive, N=Negative
* V: S = Slice Used, X = Slice not used
* W: P = Production Used, X = Production not used
* X: M = Material Used, X = Material not used
* YYYY: Test case number
* ZZ: Test case iteration

Examples:

* **P\_XSP\_0123\_02** – This is positive test case 123, iteration 02, that uses the Slice and Production Extensions
* **N\_MXX\_0234\_04** – This is negative test case 234, iteration 04, that uses the Material Extension

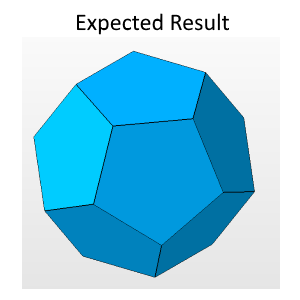
## Test Case Template

The following template will be used for test case definition. Note a general test scenario is followed by one or more iterations of a test case required to validate the conformance requirements targeted.

**N\_???\_0123 Test Scenario Name**

|  |  |
| --- | --- |
| **Test Scenario Description** | 3MF test files where the StartPart relationship defined in \_rels/.rels does not point to the root 3d Model part |
| **Pass/Fail Criteria** | Printer rejects 3MF File |
| **Test Case Iterations** | **01 -** StartPart points at non-existent part  **02 -** StartPart points at non-root model  **03 -** StartPart points at Thumbnail  **03 - Etc.** |

Where practical, each 3MF test file will contain a 300 X 300-pixel thumbnail showing the expected result of rendering the 3MF file on a consumer as shown below. A few samples are shown below. These thumbnails are captured from a variety of applications and are not meant to be used as a definitive acceptance criterion.



# Test Case Definitions

## Positive OPC Test Cases

### P\_???\_0101 Content Type Variation

|  |  |
| --- | --- |
| **Test Scenario Description** | Verify that content type mapping is case insensitive and that overrides do take precedence of the default extension definitions. |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a content type where the case of the content type extension differs from the case used by the model part. The extension should be case insensitive.  **02** **–** Create a 3MF 3D model part with no extension, and use a content type override to define the content type of that part.  **03** **–** Define a content type override for a 3D model part whose case differs from the 3D model part name. The override should be case insensitive. |

### P\_???\_0102 Content Type Override

|  |  |
| --- | --- |
| **Test Scenario Description** | Various permutations of content type overrides |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create a test case where content types contains no default for “.model” but rather has an explicit override for each model part. Rename the extensions of the model parts to something other than “.model”  **02** **–** Create a test case where content types contains a default for “.model” and an override for the 3D model part. Rename the extensions of the model part reference in the override to something other than “.model”  **03** **–** Create a test file with a thumbnail. Leave the default mapping for “.model” as is. Rename the thumbnail extension to “.model” and define an override for the thumbnail mapped to the appropriate type for a thumbnail. |

### P\_???\_0103 Unused Default Mapping

|  |  |
| --- | --- |
| **Test Scenario Description** | Unused default content type mapping |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Include an unused default mapping (to Thumbnail) that is not used (i.e., no Thumbnail in package) |

### P\_???\_0104 Model Part Root Names

|  |  |
| --- | --- |
| **Test Scenario Description** | Various character mappings in part names |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create 3MF file with model part names that use alphanumeric characters from 0-9, a-z, A-Z, and the following unreserved characters: \_ - ~  **02** **–** Create 3MF file with model part names that use alphanumeric characters that include the following reserved characters: @ ! $ ( ) + , ; =  **03** – Create 3MF file with model part root names that use a UTF-8 Unicode character from outside the standard ASCII character set, excluding symbols (use Windows charmap.exe) |

### P\_???\_0106 Thumbnail Image Size Range

|  |  |
| --- | --- |
| **Test Scenario Description** | Various thumbnail image sizes |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | **01–** Use a large jpg thumbnail image typical of a photo from a phone  **02–** Use an extremely small (20 pixel) PNG Thumbnail image |

### P\_???\_0107 TargetMode

|  |  |
| --- | --- |
| **Test Scenario Description** | Valid relationship target mode permutations |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Root 3MF file with relationship to a non-root model file that omits the TargetMode attribute  **02** **–** Root 3MF file with relationship to a non-root model file that uses a TargetMode = “Internal” attribute |

## Negative OPC Test Cases

### N\_???\_0202 Path Segment Period Ending

|  |  |
| --- | --- |
| **Test Scenario Description** | Path segment that ends with a period |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a path segment that ends with a period. As the operating system will not allow subdirectories ending in a period and zip utilities also disallow periods at the end, the best this test case can do is use a period at the end of an existing referenced path in the XML |

### N\_???\_0203 Path Segment Only Period

|  |  |
| --- | --- |
| **Test Scenario Description** | Path segment with only a period |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a path segment with only a period. As the operating system will not allow subdirectories with just a period and zip utilities also disallow standalone periods, the best this test case can do is use a period as part of an existing referenced path in the XML |

### N\_???\_0204 Relationships Part Content Type Parameter

|  |  |
| --- | --- |
| **Test Scenario Description** | Add a parameter to the relationships part |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Add a parameter to the relationships part content type (i.e ?cow=&quot;Moo&quot;). Printer should reject. |

### N\_???\_0205 Duplicate Content Type

|  |  |
| --- | --- |
| **Test Scenario Description** | Duplicate content type mappings |
| **Pass/Fail Criteria** | 01 to 02 – Printer should generate error |
| **Test Case Iterations** | **01** **–** Add duplicate default content type mappings for the “.model” extension  **02** **–** Add a duplicate override content type for a 2D model part |

### N\_???\_0206 Empty Extension Content Type

|  |  |
| --- | --- |
| **Test Scenario Description** | Content type with an empty extension |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | 01-Create a content type with an empty extension (i.e. “”) |

### N\_???\_0207 Empty Partname String

|  |  |
| --- | --- |
| **Test Scenario Description** | Override with an empty partname string |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a content type override with an empty partname string |

### N\_???\_0208 Self Reference Relationship

|  |  |
| --- | --- |
| **Test Scenario Description** | Add a self-reference relationship to root model file |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Add a self-reference relationship to root model in the model.res file |

## Positive 3MF Core Test Cases

### P\_???\_0302 StartPart Location and Name

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then reposition and rename the StartPart pointed to by the .rels file |
| **Pass/Fail Criteria** | 01 to 03 – Successfully render object with no processing errors |
| **Test Case Iterations** | **01 –** Reposition StartPart at root of package.  **02 –** Reposition the StartPart (3DModel.model) several sub folders deep  **03 –** Rename the root model to something other than 3DModel.Model (maintain. model extension) |

### P\_???\_0304 Part Relationships

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a 3MF file that contains the appropriate relationship pointers to a Print Ticket, Thumbnail, and Core Properties Parts |
| **Pass/Fail Criteria** | 01 – Successfully render object with no processing errors  02 – Printer should ignore extra thumbnail  03 – Printer should process the file correctly  04 – Printer should process the file correctly |
| **Test Case Iterations** | **01 –** Add an Empty PrintTicket, empty Core Properties, and PNG Thumbnail to a 3mf file, include the root relationships to the Core Properties and Thumbnail, and 3D Model relationship for PrintTicket.  **02** – Include two package level thumbnails in the same OPC package    **03** – Have a package level and object level thumbnail reference point to the same png file.  **04** **–** Change the ID of the StartPart relationship ID in the root .rels file to something other than “rel0” |

### P\_???\_0306 Units

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify the 3D model files to support each of the supported enumerations for the unit attribute of the Model element |
| **Pass/Fail Criteria** | 01 to 07 – Printer should process correctly |
| **Test Case Iterations** | Modify the build transform matrix of the 3D model file such that each of the images have the same size and positioning regardless of the unit used.  **01 –** Micron  **02 –** Millimeter  **03 –** Centimeter  **04 –** Inch  **05 –** Foot  **06 –** Meter   1. **–** Unspecified (should default to millimeter) |

### P\_???\_0307 Metadata - Core

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then add metadata elements from the core specification to the 3D model parts |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Include the metadata values defined in table 8-1 of the Core specification into the 3D model parts. Syntax for metadata is…  <metadata name="title">this is a title</metadata> |

### P\_???\_0308 Metadata - Vendor

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then add vendor specific metadata elements to the 3D model parts. |
| **Pass/Fail Criteria** | 01 - Printer should process correctly, ignoring the vendor specific metadata values |
| **Test Case Iterations** | **01 –** Include the vendor specific metadata values defined to 3D model files  xmlns:v="http://schemas.qualitylogic.com/vendorspecific"  <metadata name="v:anyname">this is a test</metadata> |

### P\_???\_0309 Overlapping objects

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a 3MF file such that two objects are overlapping such that the positive full rule is applied by the printer |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Use 3D Builder to position the two instances of the test object such that the two objects overlap with empty space in the middle of the overlapping segments of the object. Make sure objects stay at least 30mm x 35mm x 30mm away from the origin of the print bed. |

### P\_???\_0310 Build Item

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file with two objects, but with only one item referenced in a build item |
| **Pass/Fail Criteria** | 01 – Printer should process correctly, although only the object in the build item should be rendered |
| **Test Case Iterations** | **01 –** Add both test objects to 3D builder, confirm two items are listed in the build element (modify if needed), then remove one item from the build object. The resulting test file should render a single object although it contains two objects |

### P\_???\_0311 Build Item Transform

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file with one object, but two build items each referencing the same object, applying a different build item transform on one item |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Modify one build item transform matrix to position the two objects such that they are non-intersecting |

### P\_???\_0312 Base Material References

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file with one object and add a basematerial to the 3dmodel file |
| **Pass/Fail Criteria** | 01 – Printer should NOT render the colors, but should print the object in its native material color |
| **Test Case Iterations** | **01 –** Specify the color red in the object element via a reference to the basematerials definition. Include pid, p1, p2, p3 attributes for blue in one triangle. NOTE: Basematerials references are for display only and will not impact the printer rendered output. |

### P\_???\_0313 JPEG Thumbnail

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file with a JPEG Thumbnail |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a 3MF with a jpeg thumbnail at the package level |

### P\_???\_0314 solidsupport, support, surface

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a 3MF document with two objects. Modify the object type attribute in the 3D object such that one object had a type of “model” and the other one of the following: “solidsupport”, “support” |
| **Pass/Fail Criteria** | 01 – Printer should process correctly, although rendering of support my be device dependent |
| **Test Case Iterations** | **01 –** model and solidsupport  **02 –** model and support  **03** – model and support where support only has 3 triangles  **04** – model and surface  **05** - model and solidsupport not inside component |

### P\_???\_0315 Name Attribute White Space

|  |  |
| --- | --- |
| **Test Scenario Description** | White space in name attribute of object |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Modify the name attribute of the object element to include leading, trailing, and intermediate white space including the space character and tab |

### P\_???\_0316 Model Element Language Attribute

|  |  |
| --- | --- |
| **Test Scenario Description** | Omit the lang attribute |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Omit the lang attribute from the model element |

### P\_???\_0317 Duplicates of Multiple Mesh Objects

|  |  |
| --- | --- |
| **Test Scenario Description** | Duplicates of multiple mesh objects |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** A test case that use 3 different mesh objects defined in the same root model parts to build 24 objects, 8 of each via build item element on the build platform. Objects should be positions both adjacent in XY space, and stacked in the Z space. |

### P\_???\_0318 Positive Fill Rule

|  |  |
| --- | --- |
| **Test Scenario Description** | Objects with patterns that trigger positive fill rule |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Test case with mesh object definition requiring application of the positive fill rule. Scenario should include at least the first two examples of the positive file rule shown in section 4.1.1 of the core specification utilizing 3D objects. |

### P\_???\_0319 Undetermined Language

|  |  |
| --- | --- |
| **Test Scenario Description** | Undetermined language |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Define test file where model lang attribute is “und”. |

### P\_???\_0321 Negative Determinant

|  |  |
| --- | --- |
| **Test Scenario Description** | Create mesh with negative determinant |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create transform with negative determinant and a mesh with negative object volume |

### P\_???\_0322 JPEG APP1 Marker

|  |  |
| --- | --- |
| **Test Scenario Description** | JPEG thumbnail Image with APP1 Marker |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | 1. **–** JPEG thumbnail Image and texture with APP1 Marker |

### P\_???\_0323 PNG Specification Support

|  |  |
| --- | --- |
| **Test Scenario Description** | Various headers in PNG files |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | **01** – PGN texture with tRNS and iCCP (include one or more in test case)  **02** – PNG thumbnail with one of the MUST ignore items: sRGM, cHRM, gAMA, sBIT  (include one or more in test case) |

### P\_???\_0324 Object Thumbnail Relationship

|  |  |
| --- | --- |
| **Test Scenario Description** | Thumbnail relationship to non-root model |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Associate a thumbnail via relationship and object attribute with a non-root model file. |

### P\_???\_0325 Two Segment Model Part Name

|  |  |
| --- | --- |
| **Test Scenario Description** | Two segment model part name |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Modify Content Types so that extension “.model” is “.part”, then model balance of test file to use this extension placing parts in the default /3D folder |

### P\_???\_0326 Identity Singular Transform Matrix

|  |  |
| --- | --- |
| **Test Scenario Description** | Object with Identity and singular matrix transform |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create a build item with no transform matrix  **02 –** Create a build item with an identity transform matrix  **03 –** Create a build item with a singular transform matrix |

### P\_???\_0327 Interlocking Objects

|  |  |
| --- | --- |
| **Test Scenario Description** | 3MF test file with interlocking objects |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Use build item transforms to create a set of 2 interlocking, but not overlapping objects. |

### P\_???\_0328 Overlapping Objects

|  |  |
| --- | --- |
| **Test Scenario Description** | 3MF test file with Overlapping objects |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Specify build item transform so that two objects overlap |

### P\_???\_0329 Part Number Attribute

|  |  |
| --- | --- |
| **Test Scenario Description** | Use of PartNumber attribute of Object |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Include a PartNumber attribute in object element in a test file |

### P\_???\_0330 Minimal Self-Intersections

|  |  |
| --- | --- |
| **Test Scenario Description** | Mesh object with self-intersections |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a mesh object with self-intersections. Printer will most likely correct self-intersection triangles in mesh object. |

### P\_???\_0331 Non-Degeneracy

|  |  |
| --- | --- |
| **Test Scenario Description** | Mesh object with zero area triangle |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a mesh image with a zero-area triangle. Printer will most likely ignore zero-area triangle in mesh object. |

### P\_???\_0333 Decimal Precision

|  |  |
| --- | --- |
| **Test Scenario Description** | Various permutations of decimal precision |
| **Pass/Fail Criteria** | 0 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Use 3D vertex values with no decimal places to define mesh object  **02 –** Use 3D vertex values with 10 decimal places to define mesh object  **03 –** Use 3D vertex and unit of meter such that changes in very small decimal values impact the object shape in a meaningful way |

### P\_???\_0334 JPEG Image Markers

|  |  |
| --- | --- |
| **Test Scenario Description** | Thumbnail JPEG images with different markers |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Test case with JPEG Thumbnail that contains the APP0 marker.  **02** **–** Test case with JPEG Thumbnail that contains the APP2, APP13, and APP14 marker. |

### P\_???\_0335 Object Thumbnail Attribute

|  |  |
| --- | --- |
| **Test Scenario Description** | Thumbnail JPEG images with different markers |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
|  | **01** **–** Define a thumbnail from an object with mesh data using the thumbnail attribute. Use a PNG Thumbnail image. Make sure to include appropriate relationship pointers.  **02 –** Define a thumbnail from an object with a component reference using the thumbnail attribute. Use a JPEG thumbnail image. Make sure to include appropriate relationship pointers.  **03** – Use a thumbnail attribute on a model element in a root model, and if the production extension is supported, on a non-root model part  **04** – Use a texture rather than thumbnail relationship type associated with thumbnails at the package, model, and object level |

### P\_???\_0336 Custom Part - Preserve

|  |  |
| --- | --- |
| **Test Scenario Description** | Custom part with and without a root relationship mustPreserve type |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
|  | **01** **–** Define a custom part with no mustPreserve root relationship  **02** **–** Define a custom part with a mustPreserve root relationship |

### P\_???\_0337 metadatagroup, preserve, type

|  |  |
| --- | --- |
| **Test Scenario Description** | Test metadatagroup in build item and object. Use preserve and type attributes in metadata where it is allowed to appear |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
|  | **Use the table below to construct various iterations of metadata at the model, build item, and object level, conditionally using the preserve and type attribute.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Iteration** | **Model**  **Metadata** | **Object**  **metadatagroup** | **Build item**  **metadatagroup** | **Preserve** | **Type** | **Other** |
| **01** | **Description** |  |  | **false** |  |  |
| **Title** |  |  | **true** |  |  |
| **Copyright** |  |  |  | **string** |  |
| **CreationDate** |  |  |  | **date** |  |
| **LicenseTerms** |  |  | **true** | **string** |  |
| **x:vendor1** |  |  | **false** |  |  |
| **x:vendor2** |  |  | **true** |  |  |
| **x:vendor3** |  |  |  | **string** |  |
| **x:vendor4** |  |  |  | **date** |  |
| **x:vendor5** |  |  | **true** | **integer** |  |
| **02** |  | **x:vendor1** |  |  |  |  |
|  | **x:vendor2** |  | **false** |  |  |
|  | **x:vendor3** |  | **true** |  |  |
|  | **x:vendor4** |  |  | **string** |  |
|  | **x:vendor5** |  |  | **date** |  |
|  | **x:vendor6** |  | **true** | **integer** |  |
| **03** |  |  | **x:vendor1** |  |  |  |
|  |  | **x:vendor2** | **false** |  |  |
|  |  | **x:vendor3** | **true** |  |  |
|  |  | **x:vendor4** |  | **string** |  |
|  |  | **x:vendor5** |  | **date** |  |
|  |  | **x:vendor6** | **true** | **integer** |  |
| **04** | **x:vendor1** |  |  | **true** | **string** |  |
|  | **x:vendor2** |  | **true** | **string** |  |
|  |  | **x:vendor3** | **true** | **string** |  |
| **05** | **Description** |  |  | **true** | **string** |  |
|  | **Title** |  | **true** | **string** | **s/ignore** |
|  |  | **LicenseTerms** | **true** | **string** | **s/ignore** |
| **06** | **x:vendor1** |  |  | **true** | **string** |  |
|  | **x:vendor2** |  | **true** | **string** | **Non-root** |
|  |  | **x:vendor1** | **true** | **string** |  |

## Negative 3MF Core Test Cases

### N\_???\_0402 Invalid StartPart

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify so that the relationship StartPart such that does not resolve to a valid model root file |
| **Pass/Fail Criteria** | 01 to 04 - Printer error should be generated stating that the file cannot be processed |
| **Test Case Iterations** | **01 –** Use an incorrect folder name in the StartPart Target  **02 –** Use an incorrect file name in the StartPath Target  **03 –** Point the StartPart to a thumbnail or non-root file  **04 –** Specify a StartPart relationship with a TargetMode=”External” and a URL Target (i.e. <http://www>...) |

### N\_???\_0403 External References

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify so that there are external relationships in the root relationship file |
| **Pass/Fail Criteria** | 01 – printer error |
| **Test Case Iterations** | **01 –** Specify a Thumbnail relationship in the root rels file withTargetMode=”External” and a root model URL Target (i.e. <http://www>...) |

### N\_???\_0404 Invalid Content Type

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify to create invalid combinations of content types in the [Content.Types].xml file. Refer to Appendix C of Core Spec. |
| **Pass/Fail Criteria** | 01 to 04 – Printer error should be generated stating that the file cannot be processed |
| **Test Case Iterations** | **01 –** In the [Content.Types].xml, modify the extension “model” to “item”, such that a content type for “model” does not exist  **02 –** In the [Content.Types].xml, modify the ContentType for “model” to “application/vnd.ms-package.xxxxx-3dmodel+xm”, such that the content type for “model” is invalid  **03 –** In the [Content.Types].xml, modify the ContentType for “.rels” to “application/vnd.openxmlformats-package.xxxxx-relationships+xml”, such that the content type for “.rel” is invalid  **04 –** In the [Content.Types].xml, modify the ContentType for “png” to “image/xxxpng”, such that the content type for “png” is invalid |

### N\_???\_0405 Invalid Relationships

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify to create invalid relationship pointers. The test file will require a Thumbnail image. |
| **Pass/Fail Criteria** | 01 to 05 – Printer Error |
| **Test Case Iterations** | **01 –** Add a Thumbnail to a 3MF file with invalid root relationship target to the Thumbnail part  **02** **–** Add an incorrect relationship “Type” attribute value in the root .rels file for the relationship that points to the StartPart  **03** **–** Add an incorrect relationship “Type” attribute value in the 3dmodel.model.rels file for the relationship that points to a non-root model file  **04** **–** Deleted  **05 –** Add an incorrect relationship “Type” attribute value in root .rels part for the relationship that points to the Thumbnail |

### N\_???\_0406 Duplicate Relationship

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify to create duplicate relationship pointers |
| **Pass/Fail Criteria** | 01 to 02 – Printer Error |
| **Test Case Iterations** | **01 –** Modify the root .rels file so that there are identical instances of the start part pointer to the root model file, but using unique IDs (rel0 and rel1)  **02 –** Modify the 3dmodel.model.rels file so that there are two instances of a pointer to the same non-root model file, but with unique IDs. |

### N\_???\_0407 Missing Relationship Target

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify relationship pointers |
| **Pass/Fail Criteria** | 01 to 02 – Printer Error |
| **Test Case Iterations** | **01 –** Modify the 3dmodel.model.rels file so that the relationship target part name for the non-root model part does not match the part name in the 3MF document.  **02 –** Rename the 3dmodel.model.rels file so that it has a name that does not map to the 3dmodel.model part (i.e. wrongmodel.model.rels.) |

### N\_???\_0409 Use of Space Attribute

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify XML encoding of root element to use the unsupported “space” attribute |
| **Pass/Fail Criteria** | 01 – Printer Error |
| **Test Case Iterations** | **01 –** Modify the second line of the 3dmodel.model file to include the following attribute definition to the model element: xml:space="preserve" |

### N\_???\_0410 Invalid Metadata

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then add unrecognized vendor specific metadata elements to the 3D and 2D model parts, as well as duplicate names in metadata |
| **Pass/Fail Criteria** | 01 to 04 – Printer should generate error |
| **Test Case Iterations** | **01 –** Include the vendor specific metadata value to the 3D model part with no declared namespace prefix in the Model element  <metadata name="x:anyname">this is a test</metadata>  **02 –** Include the vendor specific metadata value to the 2D model part  Use the following metadata value in both test cases 01 and 02:  <metadata name="x:anyname">this is a test</metadata>  **03 –** Include the same metadata name from Table 8-1 of the core specification twice in the 3D model part  <metadata name="Title">this is a title</metadata>  <metadata name="Title">this is another title</metadata>  **04** - Include the same metadata name from Table 8-1 of the core specification twice in the 2D model part  Use the following metadata values in both test cases 03 and 04:  <metadata name="Title">this is a title</metadata>  <metadata name="Title">this is another title</metadata> |

### N\_???\_0411 Non Unique Triangle Indices

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file, then modify on triangle indices so that the vertex references are not unique |
| **Pass/Fail Criteria** | 01 – Printer should generate error, although it is unclear if the printer will parse the 3D file in this level of detail. |
| **Test Case Iterations** | **01 –** Modify on triangle indices so that the vertex references are not unique |

### N\_???\_0412 Invalid Index Range

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file that has out of range index references to vertex values |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01** – Modify a 3DModel triangle vertex attribute (v1, v2, or v3) so that it is +2 larger than the number of vertexes  **02** – Modify a 2DModel polygon segment vertex attribute (v2) so that it is +2 larger than the number of vertexes  **03** **–** Modify a 2DModel polygon startv attribute (v2) so that it is +2 larger than the number of vertexes  **04** – Add a test case where the vertices element is missing altogether from a slice stack, but polygons are defined with invalid vertex index references. |

### N\_???\_0413 Non-Unique ID Values

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a simple 3MF file that has non- unique values for relationships in the same. rels file and the resources group (object ID) of a part |
| **Pass/Fail Criteria** | 01 and 02 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a 3MF file with two items in the build item path. Modify the 3dmodel.model.rels file to create duplicate IDs.  **02 –** Create a 3MF file with two objects with the same ID values |

### N\_???\_0415 Absolute Path Names

|  |  |
| --- | --- |
| **Test Scenario Description** | Scenarios involving invalid paths |
| **Pass/Fail Criteria** | 01 to 04 – Printer should generate error |
| **Test Case Iterations** | **01 –** Define a model part name with a leading period and reference that part in a component reference  **02 –** Attempt to reference a non-root model part using a relative component path attribute  **03 –** Attempt to reference a non-root model part using a relative build path attribute  **04 –** Attempt to reference a non-root model part in path using the path only in the component path element, omitting the part name |

### N\_???\_0416 Negative Volume Mesh

|  |  |
| --- | --- |
| **Test Scenario Description** | Object with negative volume mesh |
| **Pass/Fail Criteria** | 01 and 02 – Printer should generate error |
| **Test Case Iterations** | **01** - Create negative volume mesh  **02** – Positive volume mesh with negative determinant transform |

### N\_???\_417 Prior Object References

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a forward reference scenario in the root model file |
| **Conformance Statement ID(s)** | Core-M1.60, Core-M1.33 |
| **Base Test Object(s)** | S11\_cube\_NA.3mf |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a forward reference by defining an object with a reference to a slicestackID before that slice stack appears in the XML stream |

### N\_???\_0418 Inward Facing Normal Face

|  |  |
| --- | --- |
| **Test Scenario Description** | Mesh with inward facing normal face |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01**-Reverse order of vertices such that normal face is pointing inwards. |

### N\_???\_0419 CMYK Images

|  |  |
| --- | --- |
| **Test Scenario Description** | JPEG thumbnail Image with CMYK |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** JPEG thumbnail Image with CMYK |

### N\_???\_0420 Data Type Definitions

|  |  |
| --- | --- |
| **Test Scenario Description** | DTD declaration |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Include a DTD declaration as follows after the XML header:    <!DOCTYPE note [  <!ELEMENT note (to,from,heading,body)>  <!ELEMENT to (#PCDATA)>  <!ELEMENT from (#PCDATA)>  <!ELEMENT heading (#PCDATA)>  <!ELEMENT body (#PCDATA)>  ]> |

### N\_???\_0421 Transforms

|  |  |
| --- | --- |
| **Test Scenario Description** | Various transforms that result in invalid printable objects |
| **Pass/Fail Criteria** | 01 to 02 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a simple test case, then change the build item transform so that it is outside the printable area of the printer with a portion of the image in a negative quadrant  **02 –** Create a simple test case, then change the build item transform so that the object has close to zero volume (Note that printer may accept even if nothing is rendered – Consider moving to positive case |

### N\_???\_0422 Commas and German locale

|  |  |
| --- | --- |
| **Test Scenario Description** | Specifying non en-US locale in model element |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01** –specify the lang attribute in all model elements (core model) to “de-de” and modify all decimal places as commas (,). |

### N\_???\_0424 Material in Object with Component

|  |  |
| --- | --- |
| **Test Scenario Description** | Add basematerial reference to object with a component |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01** – Create an 3mf file with a colorgroup or basematerial defined, and specify a pid and pindex attribute in an object element that contains a component sub-element that point to other objects. |

### N\_???\_0426 Model with Less Than 4 Triangles

|  |  |
| --- | --- |
| **Test Scenario Description** | Mesh object with only 3 triangles |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create invalid mesh object with only 3 triangles |

### N\_???\_0427 Duplicate 3D Vertex and Non-Manifold Edge

|  |  |
| --- | --- |
| **Test Scenario Description** | Mesh object with duplicate 3D Vertex and non-manifold edges |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Create a non-manifold mesh image and duplicate 3D vertex. Printer may ignore non-manifold mesh object and duplicate vertex. |

## Positive 3MF Material Extension Test Cases

*The Material Extension test cases will utilize the predefined colors, gradients, textures, and multiproperties defined in Appendix B. To the extent that specific colorgroups, textures, gradients, or multiproperties are not specified in the test case definition, the test case developer shall iterate through a random selection of the defined resources in Appendix B.*

### P\_???\_0501 Default Material Color

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate use of object pid and pindex attributes as a default material color if a triangle does not have a material color specified |
| **Pass/Fail Criteria** | 01 to 09 – Printer should process correctly |
| **Test Case Iterations** | **01** - Multiproperties as default object color. Apply colorgroup to at least one triangle to demonstrate that only triangles with undefined material color are impacted by the default.  **02** - Texture as default object color. Apply colorgroup to at least one triangle to demonstrate that only triangles with undefined material color are impacted by the default.  **03** - Colorgroup as default object color. Texture at least one triangle to demonstrate that only triangles with undefined material color are impacted by the default.  **04** - No material color specified in object or triangles  **05** - Deleted  **06 -** Deleted.  **07** – Texture as a default color where tex2coord pointed to by pindex is a u v value greater than 1 1   1. – Multipropertes as a default color where tex2coord pointed to by pindex is a u v value greater than 1 1 2. Texture with tilestyle of “none” as a default color where tex2coord pointed to by pindex is a u v value greater than 1 1 . This may result in system dependent behavior. |

### P\_???\_0502 Triangle P1, 2, 3 Usage

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that If p1 is not specified, then the default property is assigned to the triangle. If p2 or p3 is unspecified then p1 is used for the entire triangle. |
| **Pass/Fail Criteria** | 01 to 05 – Printer should process correctly |
| **Test Case Iterations** | **01 -** Just p1 Specified  **02** - Just p1 and p2 Specified  **03** - Just p1 and p3 Specified  **04** - p2 and p3 Specified, but not p1 – With default object color defined  **05** - p1, p2, and p3 unspecified |

### P\_???\_0503 Ignoring Unsupported Materials

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that the printer ignores both basematerials and compositematerials |
| **Pass/Fail Criteria** | 01 to 08– Printer should process correctly |
| **Test Case Iterations** | **01** - Use basematerials as object default. Also map colorgroup to one triangle  **02** - Use compositematerials as object default. Also map colorgroup to one triangle  **03** - Use basematerials as triangle PID/P1 with colorgroup as object default  **04** - Use compositematerials as triangle PID/P1 with texture as object default  **05** - Use basematerials as multiproperties pids component. Map multiproperties as object default. The multiproperties should also include a colorgroup which should get rendered.  **06** - Use composite material as multiproperties pids component. Map multiproperties as object default. The multiproperties should also include a colorgroup which should get rendered.  **07** - Use basematerials as multiproperties pids component. Map multiproperties as triangle PID/P1 with texture as object default. The multiproperties should also include a colorgroup which should get rendered.  **08** - Use compositematerials as multiproperties pids component. Map multiproperties as triangle PID/P1 with colorgroup as object default. The multiproperties should also include a colorgroup which should get rendered. |

### P\_???\_0504 Opaque first Layer

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that the first layer of material color applied to an object is opaque both when defined as an object default and as triangle specific |
| **Pass/Fail Criteria** | 01 to 07 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Use a colorgroup color with transparent alpha values to demonstrate the first layer applied to an object is opaque  **02 –** Use a texture with transparent alpha values to demonstrate the first layer applied to an object is opaque  **03 –** Use a colorgroup color with transparent alpha values to demonstrate the first layer of multiproperties is opaque. The second layer should also be partially transparent to illustrate the effect using a texture.  **04 –** Use a texture with transparent alpha values to demonstrate the first layer of multiproperties is opaque. The second layer should also be partially transparent to illustrate the effect.  **05** – Use a multiproperties with 3 materials/colors, with the 2nd and 3rd layer being partially opaque. Demonstrate that the 1st and 2nd layer are opaque once merged together.  **06 -** Use a texture that uses tilestyle of “none” with uv values great than 1 with a transparent object default color  **07** – Use a multiproperties with the first layer as basematerials and the second layer a transparent texture. |

### P\_???\_0505 Pindices List

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate the behavior when too few or too many items are listed in the pindices list. |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01 -** In multiproperties have a pindices list that is one values shorter than the pids list. Demonstrate that an index value of zero is used for the omitted pindices value. For this test case have the omitted pids be colorgroup  **02 -** In multiproperties have a pindices list that is one value shorter than the pids list. Demonstrate that an index value of zero is used for the omitted pindices value. For this test case have the omitted pids be texture  **03** – Demonstrate in multiproperties that if pindices includes one extra value greater that the number of pids values, the printer should ignore the extra value |

### P\_???\_0506 Multiple Material Colors and Groups

|  |  |
| --- | --- |
| **Test Scenario Description** | Utilities define multiple groups of material colors and utilize multiple colors from each group. |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01** – Define and utilize multiple colorgroup references on an object, using multiple indexed items from each colorgroup  **02** – Define and utilize multiple texture2dgroup references on an object, using multiple indexed items from texture2dgroup  **03** - Define and utilize multiple multiproperties references on an object, using more than one material/color in each multiprop1erty  **04** – On the same object use one each of the following: Colorgroup, texture, and multiproperties |

### P\_???\_0507 Material Stress Tests

|  |  |
| --- | --- |
| **Test Scenario Description** | Large numbers of material color components in a test file |
| **Pass/Fail Criteria** | 01 to 09 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Use 120 texture2d images mapped to 120 texture2groups. Paint one object with each of the defined textures  **02 –** Use 100,000 text2coord mappings on one object  **03 –** Use more than 1,000 colorgroups each with one color. Paint one object with each of the defined colorgroups.  **04 –** Use more than 1,000 colors in one colorgroup to paint an object  **05 –** Define120 multiproperties each with three multi sub elements and two pids values. Paint one object with each of the multiproperties.  **06 – Use** 2,500+ multi sub element indices in one multiproperties with two pids values. Paint one object with each of the defined multiproperties values.  **07 –** A multiproperties listing 7 entries in pids and pindices. The same texture pids references can be interleaved.  **08 -** Extremely small text2coord values (at the level of a pixel)  **09 -** Extremely large text2coord values (such to cause 1000 tiles) |

### P\_???\_0508 Real World Textured Objects

|  |  |
| --- | --- |
| **Test Scenario Description** | Paint textures on a variety of real world objects |
| **Pass/Fail Criteria** | 01 to 14 – Printer should process correctly |
| **Test Case Iterations** | **The following table listing specific textures from Appendix B to specific real world objects in Appendix A:**  01 - brmarble.jpg - N22\_ChessHorse\_high  02 - droplets.png - M11\_box\_NA  03 - grmarble.jpg - M12\_Extruder\_Bowden\_Adapterc\_low  04 - oak.png - M11\_Ventilated Build Platform\_low  05 - pitissue.jpg - M12\_role\_drum\_NA  06 - purmesh.png - N33\_Duck\_NA  07 - quads.jpg - M12\_SW\_Extruder-Hinged-Block\_high  08 - photo\_1.jpg - M11\_stereographic\_maze\_lowres\_NA  09 - photo\_1.png\_16 - M12\_Tristruder\_18mm\_Probe\_Mount\_high  10 - photo\_2.jpg - M21\_flex\_coupler\_NA  11 - photo\_3.png - N32\_alligator\_228\_low  12 - photo\_4.jpg - M22\_FPV\_Pod\_Camera\_Plate\_NA  13 - photo\_5.jpg - M22\_FPV\_Pod\_Half\_NA  14 - photo\_6.png - N23\_Deer\_high |

### P\_???\_0509 Positive Ordering

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that colorgroup, texture2group, and multiproperties can be interleaved in resources as long as there are no forward references |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 – D**efine model part with the following ordering of materials in the resources element: colorgroup, texture2d, texture2dgroup, multiproperties, texture2d, colorgroup, texture2dgroup, multiproperties, object. There should be no forward references and all materials should be referenced in an object's triangles. |

### P\_???\_0510 Color Groups

|  |  |
| --- | --- |
| **Test Scenario Description** | Description |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01 -** Define one colorgroup with 10 colors, use all colors on one object  **02 -** Define two colorgroups, each with 10 colors. Interleave use of colors from the two different colorgroups on the same object's triangles  **03 -** Define one colorgroup with 10 colors, use each color as the default color on a separate object  **04 -** Define two colorgroups, each with 4 colors. Interleave use of colors from the two different colorgroups as the default color on separate objects |

### P\_???\_0511 Color Values

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate iterating through values for the color attribute of colorgroup |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01** - Iterate through all 256 red channel values displaying each value in a triangle in the same object while omitting the alpha channel (#XX0000)  **02** - Iterate through all 256 green channel values displaying each value in a triangle in the same object while omitting the alpha channel (#00XX00)  **03** - Iterate through all 256 blue channel values displaying each value in a triangle in the same object while omitting the alpha channel (#0000XX)  **04** - Iterate through all 256 RGB channel values concurrently with an opaque alpha channel value in a triangle in the same object (#XXXXXXFF) |

### P\_???\_0512 Gradients

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate use various gradients using the standard gradient patterns defined in Appendix B |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01 -** Use Gradient\_1, Gragient\_2, and Gradient\_3 using a color value that omits alpha (#XXXXXX)  **02 -** Use Gradient4, Gragient\_5, and Gradient\_6 using a color value that includes an opaque alpha (#XXXXXXFF)  **03\_**Use Gradient\_7, Gragient\_8, and Gradient\_9 using a color value that includes a 50% transparent alpha (#XXXXXX9F) as part of a multiproperties definition where texture is the first layer and partially transparent gradient in the second layer  **04-** Use a gradient as the 2nd layer in a multiproperties with a mixture of color values for p1, p2, and p3 that have no alpha channel (#XXXXXX) and a partially transparent color value (#XXXXXX2f)  **06 –** Create gradient across all 6 sides of a cube such that the transition between triangles and faces of the cube appear seamless |

### P\_???\_0513 Transparency

|  |  |
| --- | --- |
| **Test Scenario Description** | Use multiproperties to effect various alpha channel transparency behaviors |
| **Pass/Fail Criteria** | 01 to 05 – Printer should process correctly |
| **Test Case Iterations** | **01** - Iterate through all 256 alpha channel values for color using a static RGB color for triangle attributes P1, P2, and P3 as the second layer of a multiproperties  **02** - With a solid color as the first layer, use the alpha channel versions of the brmarble\_A and droplets\_A Small Texture Swatches (Appendix H) as the second layer in multiproperties  **03** - With a Gradient\_5 as the first layer, use the alpha channel versions of the grmarble\_A and oak\_A Small Texture Swatches (Appendix H) as the second layer in multiproperties  **04** - With a brmarble.jpg texture as the first layer, use the alpha channel versions of the pitissue\_A and purmesh\_A Small Texture Swatches (Appendix H) as the second layer in multiproperties    **05** –Multiproperties with second layer with tilestyle of “none” and UV values greater than 1 and a default color that is partially transparent. |

### 

### P\_???\_0514 Textures

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate the use of textures in a variety of ways. These test cases use a variety of texture patterns defines in Appendix B |
| **Pass/Fail Criteria** | 01 to 12 – Printer should process correctly |
| **Test Case Iterations** | **01**-Create a cube where each side of the cube uses one of the jpg versions of the Small Texture Swatches  **02**-Create a cube where each side of the cube uses one of the png versions of the Small Texture Swatches  **03**-Create a cube where each side of the cube alternates use of jpg and PNG images from the Small Texture Swatches  **04** - Create objects that are painted completely with the photo\_1 Large Texture Images. The test file should have 4 objects, each utilizing jpg and png forms of the Photo\_1 Image.  **05** - Create objects that are painted completely with the photo\_2 Large Texture Images. The test file should have 2 objects, each utilizing jpg and png forms of the Photo\_2 Image.  **06** - Create objects that are painted completely with the photo\_3 Large Texture Images. The test file should have 2 objects, each utilizing jpg and png forms of the Photo\_3 Image.  **07** - Create objects that are painted completely with the photo\_4 Large Texture Images. The test file should have 2 objects, each utilizing jpg and png forms of the Photo\_4 Image.  **08** - Create objects that are painted completely with the photo\_5 Large Texture Images. The test file should have 2 objects, each utilizing jpg and png forms of the Photo\_5 Image.  **09** - Create objects that are painted completely with the photo\_6 Large Texture Images. The test file should have 2 objects, each utilizing jpg and png forms of the Photo\_6 Image.  **10** - Define 7 texture2dgroups each using a different texture2d image from the Small Texture Swatches without alpha channel data. Use each texture2d group to paint at least 1 triangle on the same object  **11** - Define 7 texture2dgroups each using a different texture2d image from the Small Texture Swatches with alpha channel data. Include each texture2d group as the second layer in a multiproperties, then paint at least 1 triangle on the same object with each multiproperties  **12** – Map a texture with no area to a triangle. Repeat twice for two different triangles. One where p2, and p3 point to the same coordinate on the texture. And a second where p1, p2, and p3 point to the same coordinate. Coordinate points used shall point to distinctly different colors to illustrate the effect. |

### P\_???\_0515 PNG Formats

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate each of the allowable basic png formats utilizing the Public Domain PngSuite shown in Appendix B |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01 -** Positive - Use each of the png files defined in the Basic Format Public Domain PNG Suite (Appendix B) as a triangle texture on the same object  **02** - Use each of the PNG files defined in the Public Domain PNG Suite (Appendix B) that have an alpha channel as the second layer of a multiproperties such that the alpha behavior of the png files is apparent.    **03 –** All png images in the public test suite, including basic format tests. |

### P\_???\_0516 Multiproperties

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate various combinations of colorgroup and texture as part of multiproperties. This test cases references a standard set of multiproperties combinations defined in Appendix B |
| **Pass/Fail Criteria** | 01 to 08 – Printer should process correctly |
| **Test Case Iterations** | **01** - Define Multiproperties\_1 and \_2, and use each of them to paint on triangle on an object  **02** - Define Multiproperties\_3 and \_4, and use each of them to paint on triangle on an object  **03** - Define Multiproperties\_5 and \_6, and use each of them to paint on triangle on an object  **04** - Define Multiproperties\_7 and \_8, and use each of them to paint on triangle on an object  **05 -** Multiproperties with only one property defined in pids  **06** - Use the same texture multiple times in a pids/pindex reference in multiproperties  **07** - Demonstrate monochrome and color images with the default alpha channel blending using multiproperties  **08** - Demonstrate monochrome and color images using the multiply blend method using multiproperties |

### P\_???\_0517 Texture2D

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate various texture2D bounding box and tile style behaviors |
| **Pass/Fail Criteria** | 01 to 35 – Printer should process correctly |
| **Test Case Iterations** | **01** - Demonstrate that the default tilestyleu and tilestylev attribute default is wrap  **02** - Demonstrate that the default box attribute values are "0 0 1 1"  **04** - Specify the tilestyleu attributes without the box or tilestylev attribute using a non-default value for tilestyleu (mirror or clamp)  **05** - Specify the tilestylev attribute without the box or tilestyleu attribute using a non-default value for tilestylev (mirror or clamp)  Note: Demonstrate various combinations of wrap, mirror, and clamp as part of color and image transparency. Default tex2coord greater than 1 to trigger tiling.  **08** - Texture: u=wrap v=wrap  **09** - Texture: u=mirror v=mirror  **10** - Texture: u=clamp v=clamp  **11** - Texture: u=wrap v=mirror  **12** - Texture: u=mirror v=wrap  **13** - Texture: u=clamp v=wrap  **14** - Texture: u=wrap v=clamp  **15** - Texture: u=mirror v=clamp  **16** - Texture: u=clamp v=mirror  **20** – Demonstrate clamp, mirror and wrap behavior on separate triangles of the same object using the photo\_4.jpg, photo\_5.png, and photo\_5.jpeg images respectively  **21** – Use negative text2cood values to texture a triangle with wrap and mirror. The origin tile will be flipped in both the u and v direction.  **22** – Use negative v text2cood values to texture a triangle using a clamp and mirror tilestyle.  **23** – Use wrap on all 6 sides of a cube  **24** – Use mirror on all 6 sides of a cube  **25** – Use clamp on all 6 sides of a cube  **26** – Use a texture with an alpha channel with a tilestyle of mirror.  Note: Demonstrate various combination of a tilestyle of “none” with wrap, mirror, and clamp as part of color and image transparency. Default tex2coord greater than 1 to trigger tiling and a default object color unless specified otherwise  **27** - Texture: u=none v=none (negative uv to show wrap on 4 sides)  **28** - Texture: u=none v=none, no default object color , negative uv to show wrap on 4 sides  **29** - Texture: u=none v=wrap  **30** - Texture: u=wrap v=none  **31** – Texture: u=mirror v=none  **32** – Texture: u=none v=mirror  **33** – Texture: u=clamp v=none  **34** - Texture: u= none v=clamp  **35** – Tilestyle of “none’ on all 6 sides of a cube |

### P\_???\_0518 Texture2dGroup

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate various mapping of text2coord attribute values of the texture2dgroup element |
| **Pass/Fail Criteria** | 01 to 15 – Printer should process correctly |
| **Test Case Iterations** | **01** - text2coord u v values between 0 and 1 mapped to vertices such that the image aspect ratio is maintained  **02** - text2coord u v values between 0 and .1 mapped to vertices such that the image aspect ratio is maintained  **03** - text2coord u v values between 5 and 10 mapped to vertices such that the image aspect ratio is maintained and tiling is triggered  **04** - text3coord u values such to create an obvious and exaggerated stretching of the image in the u axis  **05** - text3coord v values such to create an obvious and exaggerated stretching of the image in the v axis  **06** - text3coord u v values such to create an obvious and exaggerated stretching of the image in both the u v axis  **07** - text3coord u values such to create an obvious and exaggerated stretching of the image in both the u v axis as well as tiling of at least 3 instances of the image in a triangle  **08** - Use negative text2coord u and v values to map the texture to a triangle using the quad.jpg Small Texture Swatch with specified uv coordinate where the distance between vertex points is not greater than 1 (no tiling). Device UV coordinates: once with just u negative, once with just v negative, and once with both u and v negative. Use all the coordinates to map a texture to a single triangle.  **09** - use negative text2coord u and v values to map the texture to triangles using the oakNumbers.png Small Texture Swatch with UV coordinate values between -1 and -3 (tiling). Use all the coordinates to map a texture to a single triangle.  **10** – Repeat test case 09 with a tilestyle of mirror for u and v  **11** - Repeat test case 09 with a tilestyle of clamp for u and v  **12** - Texture with mapped uv values all greater than 1.  **13** - Repeat test case 08 with tilestyle of “none”  **14** - Repeat test case 09 with tilestyle of “none”  **15** - Repeat test case 12 with tilestyle of “none” |

### P\_???\_0519 “e” Notation

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate the use of e notation rather than decimal values in one location to conform that implementation can handle that representation of a float |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Demonstrate the use of e notation with negative values on mesh vertex coordinate. |

### P\_???\_0520 Overlapping Material Colors

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that the last voxel rendered takes precedence when two objects overlap in a coplanar fashion. |
| **Pass/Fail Criteria** | 01 to 03 – Last color of last coplanar object rendered should have precedence |
| **Test Case Iterations** | **01** - 2 objects overlapping with gradient and textured surfaces coplanar both referenced from build items  **02** - 2 cubes defined in same object mesh with overlapping gradient and textured surfaces coplanar (Note in Thumbnail master that the result may be device dependent where two layers are coplanar), although rules state that the last overlapping triangle rendered should have precedence.  **03** - 2 cubes defined in separate objects with overlapping gradient and textured surfaces coplanar with each object referenced from components in the same object, with a single build item referenced to the assembly |

### P\_???\_0521 Units of Measure

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that units of measure declared in the model units attribute does not impact the appearance of textures. |
| **Pass/Fail Criteria** | 01– Printer should process correctly |
| **Test Case Iterations** | **01 -** Display textured object (photo\_4.jpg) in the same OPC package multiple times using separate model files, with each model file containing a different unit attribute value. Use the default (no units - millimeter), inch, and centimeters. Objects should be generated such that the object vertex values are the native units of measure with the object being the same size without the need for transform scaling. |

### P\_???\_0522 Transform Impact

|  |  |
| --- | --- |
| **Test Scenario Description** | Illustrate the impact that transforms have on texture and gradient patterns. These test cases use the predefined gradients and textures defined in Appendix B. |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | **01 -** Create an object with both gradient\_5 and texture quads.jpg. Reference the same object from 4 build items, with each reference using a different transform matrix scaling  **02 -** Create an object with both gradient\_5 and texture quads.jpg. Reference the same object from one object with 4 component references, with each component reference using a different transform matrix for scaling and positioning. |

### P\_???\_0523 OPC Package Location

|  |  |
| --- | --- |
| **Test Scenario Description** | Define objects with Colorgroups, textures, and multipropertiescan in different locations in the OPC package. Each object should utilize a solid color from colorgroups, a gradient from colorgroups, a texture, and multiproperties that uses both colorgroup as a base layer and a texture with an alpha channel. |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01** - Root and non-root model parts in same package  **02** - non-root model parts only, referenced by root build item  **03** – non-root model parts only, referenced by component  **04** – Use tilestyle combinations of wrap, mirror, clamp, and none on various sides of a single cube that whose object is in a non root model. |

### 

### P\_???\_0524 Whitespace in Delimited Values

|  |  |
| --- | --- |
| **Test Scenario Description** | Use multiple tab, space, CR, LF characters between space delimited attributes. The schema validation |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | For each of the space delimited attributes noted in the test cases below, include multiple space characters, tab CR, and LF in one set of space delimited values.  **01** – Include multiple whitespace characters in the following attributes: pids and pindices.  **02** - Include multiple whitespace values in both a build item and component transform matrix. |

### P\_???\_0525 Permutations of Layers

|  |  |
| --- | --- |
| **Test Scenario Description** | Iterate through a set of test cases, each using a unique combination of materials, alpha data, padding, tiling, gradients, and other characteristics. This is not intended to be an exhaustive list, but rather a reasonable sampling of the possibilities. Basic combinations of layers to include:   * colorgroup + texture * texture + color group * colorgroup + texture + texture * texture + colorgroup + texture |
| **Pass/Fail Criteria** | 01 to 55 – Printer should process correctly |
| **Test Case Iterations** | **The table on the following page details each test case and the layer characteristics using the following legend:**  CG = Colorgroup  TX = Texture  MP = Multiproperties  OP = Opaque  TR = Transparent  SC = Solid Color  GC = Gradient Color  GA = Gradient Alpha  WR = Wrapped (presumes tex2coord uv > 1)  MR = Mirrored (presumes tex2coord uv > 1)  CL = Clamped (presumes tex2coord uv > 1)  NO = None (presumes tex2coord uv > 1) – One or both tilestyles as none |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Iteration** | **Object Default** | **Layer 1** | **Layer 2** | **Layer 3** |
| 01 | CG | CG\_SC\_OP | TX\_TR |  |
| 02 | CG | CG\_SC\_TR | TX\_TR\_WR |  |
| 03 | CG | CG\_GC\_OP | TX\_TR\_MR |  |
| 04 | CG | CG\_SC\_OP | TX\_TR\_CL |  |
| 05 | Deleted |  |  |  |
| 06 | MP | CG\_SC\_TR | TX\_TR |  |
| 07 | TX | CG\_GC\_OP | TX\_TR\_WR |  |
| 08 | CG | CG\_SC\_TR | TX\_TR\_MR |  |
| 09 | MP | CG\_GC\_OP | TX\_TR\_CL |  |
| 10 | TX | CG\_SC\_OP | TX\_OP |  |
| 11 | TX | CG\_SC\_TR | TX\_OP\_WR |  |
|  |  |  |  |  |
| 12 | CG | TX\_OP | CG\_SC\_TR |  |
| 13 | CG | TX\_TR | CG\_SC\_TR |  |
| 14 | CG | TX\_OP\_WR | CG\_GC\_TR |  |
| 15 | CG | TX\_OP\_MR | CG\_SC\_GA |  |
| 16 | CG | TX\_OP\_CL | CG\_GC\_GA |  |
| 17 | CG | TX\_OP | CG\_SC\_TR |  |
| 18 | MP | TX\_OP\_WR | CG\_GC\_TR |  |
| 19 | TX | TX\_OP\_MR | CG\_SC\_GA |  |
| 20 | CG | TX\_OP\_CL | CG\_GC\_GA |  |
|  |  |  |  |  |
| 21 | CG | CG\_SC\_OP | TX\_TR | TX\_TR\_MR |
| 22 | CG | CG\_SC\_TR | TX\_TR\_WR | TX\_TR\_MR |
| 23 | CG | CG\_GC\_OP | TX\_TR\_MR | TX\_TR\_WR |
| 24 | CG | CG\_SC\_OP | TX\_TR\_CL | TX\_TR |
| 25 | MP | CG\_SC\_TR | TX\_TR | TX\_TR\_CL |
| 26 | TX | CG\_GC\_OP | TX\_TR\_WR | TX\_TR |
| 27 | CG | CG\_SC\_TR | TX\_TR\_MR | TX\_TR\_CL |
| 28 | MP | CG\_GC\_OP | TX\_TR\_CL | TX\_TR\_MR |
| 29 | TX | CG\_SC\_OP | TX\_OP | TX\_TR\_WR |
| 30 | TX | CG\_SC\_TR | TX\_OP\_WR | TX\_TR |
|  |  |  |  |  |
| 31 | CG | TX\_TR | CG\_SC\_TR | TX\_TR\_MR |
| 32 | CG | TX\_OP\_WR | CG\_GC\_TR | TX\_TR\_WR |
| 33 | CG | TX\_OP\_MR | CG\_SC\_GA | TX\_TR |
| 34 | CG | TX\_OP\_CL | CG\_GC\_GA | TX\_TR\_CL |
| 35 | CG | TX\_OP | CG\_SC\_TR | TX\_TR\_MR |
| 36 | MP | TX\_OP\_WR | CG\_GC\_TR | TX\_TR\_CL |
| 37 | TX | TX\_OP\_MR | CG\_SC\_GA | TX\_TR\_MR |
| 38 | CG | TX\_OP\_CL | CG\_GC\_GA | TX\_TR\_WR |
|  |  |  |  |  |
| 39 | CG | TX\_TR | TX\_OP\_MR |  |
| 40 | CG | TX\_TR\_WR | TX\_TR\_CL |  |
|  |  |  |  |  |
| 41 | CG | CG\_SC\_TR | TX\_TR\_NO\_OP |  |
| 42 | TX | CG\_GC\_OP | TX\_TR\_NO\_TR |  |
| 43 | TX | CG\_SC\_TR | TX\_OP\_NO\_TR |  |
| 44 | CG | CG\_SC\_TR | TX\_TR\_NO\_OP |  |
| 45 | CG | TX\_OP\_NO\_OP | CG\_GC\_TR |  |
| 46 | MP | TX\_OP\_NO\_TR | CG\_GC\_TR |  |
| 47 | CG | CG\_SC\_TR | TX\_TR\_NO\_TR | TX\_TR\_MR |
| 48 | CG | CG\_GC\_OP | TX\_TR\_MR | TX\_TR\_NO\_TR |
| 49 | TX | CG\_GC\_OP | TX\_TR\_NO\_TR | TX\_TR |
| 50 | Basematerials | CG\_SC\_OP | TX\_TR\_NO (no default) | TX\_TR\_NO(no default) |
| 51 | TX | CG\_SC\_TR | TX\_OP\_NO\_TR | TX\_TR |
| 52 | CG | TX\_OP\_NO\_TR | CG\_GC\_TR | TX\_TR\_NO\_TR |
| 53 | Basematerials | TX\_OP\_NO (no default) | CG\_GC\_TR | TX\_TR\_CL |
| 54 | CG | TX\_OP\_CL | CG\_GC\_GA | TX\_TR\_NO\_TR |
| 55 | CG | CG\_GC\_OP | TX\_TR\_WR | TX\_TR\_NO\_TR |
|  |  |  |  |  |

### P\_???\_0526 Filters

|  |  |
| --- | --- |
| **Test Scenario Description** | Test the various valid enumerations for the texture2d elements filter attribute |
| **Pass/Fail Criteria** | 01 to 09 – Printer should process correctly |
| **Test Case Iterations** | Use the table below to render adjacent triangles using various filter and scaling combinations. The image selected and the scaling factors should be selected to make the interpolation method used as obvious as possible.  Each test iteration has two scenarios with each presented on the upper or lower horizontal surface of a rectangular object. Most test cases will use a jpg image. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Iteration** | **TextureGroup** | **Multi** | **Downscale** | **Upscale** | **No Filter**  **Attribute** | **Auto** | **Linear** | **Nearest** |
| **01** | **X (B)** |  |  | **X** | **X** |  |  |  |
| **X (T)** |  |  | **X** |  | **X** |  |  |
| **02** | **X (B)** |  |  | **X** |  | **X** |  |  |
| **X (T)** |  |  | **X** |  |  | **X** |  |
| **03** | **X (B)** |  |  | **X** |  | **X** |  |  |
| **X (T)** |  |  | **X** |  |  |  | **X** |
| **04** | **X (B)** |  |  | **X** |  |  | **X** |  |
| **X (T)** |  |  | **X** |  |  |  | **X** |
| **05** | **X (B)** |  | **X** |  | **X** |  |  |  |
| **X (T)** |  | **X** |  |  | **X** |  |  |
| **06** | **X (B)** |  | **X** |  |  |  | **X** |  |
| **X (T)** |  | **X** |  |  |  |  | **X** |
| **07** |  | **X (B)** | **X** |  |  |  | **X** |  |
|  | **X (T)** |  | **X** |  |  |  | **X** |
| **08** |  | **X (B)** |  | **X** |  | **X** |  |  |
|  | **X (T)** |  | **X** |  |  | **X** |  |
| **09**  **(1)** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
|  |  |  |  |  |  |  |  |

1)A single object that uses a variety of filter configurations using png images

### P\_???\_0529 Display Properties

|  |  |
| --- | --- |
| **Test Scenario Description** | Include an example of each supported display property. The printer should ignore the display property and render the mesh without influence from the display property values. |
| **Pass/Fail Criteria** | 01– 05 Printer should process correctly |
| **Test Case Iterations** | **01 –** colorGroup displaypropertyid mapped to pbspeculardisplayproperties  **02 –** colorGroup displaypropertyid mapped to pbmetallicdisplayproperties  **03 –** texture2dgroup displaypropertyid mapped to pbmetallictexturedisplayproperties  **04 –** texture2dgroup displaypropertyid mapped to pbspeculartexturedisplayproperties  **05 –** basematerials displaypropertyid mapped to translucentdisplayproperties |

### P\_???\_0530 Blend Method

|  |  |
| --- | --- |
| **Test Scenario Description** | Exercise blendmethods in a variety of permutations of colorgroup, texture, any tylestyle enumerations. Note that the default “mix” behavior is tested extensively elsewhere, so the focus of this testing will be on the use of multiply either independently or in combination with mix. |
| **Pass/Fail Criteria** | 01 to 08–Printer should process correctly |
| **Test Case Iterations** | **See tables below.** |

**Characteristics combinations uses in test case definitions below.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Mix**  **(1)** | **Multiply**  **(1)** | **Solid Color** | **Gradient Color** | **Texture** | **Wrap** | **Mirror** |  | **Clamp** | **None** |
| **A** | x |  | x |  |  |  |  |  |  |  |
| **B** | x |  |  | x |  |  |  |  |  |  |
| **C** |  | X | x |  |  |  |  |  |  |  |
| **D** |  | X |  | x |  |  |  |  |  |  |
| **E** | x |  |  |  | x | x |  |  |  |  |
| **F** | x |  |  |  | x |  | x |  |  |  |
| **G** | x |  |  |  | x |  |  |  | x |  |
| **H** | x |  |  |  | x |  |  |  |  | x |
| **I** |  | X |  |  | x | x |  |  |  |  |
| **J** |  | X |  |  | x |  | x |  |  |  |
| **K** |  | X |  |  | x |  |  |  | x |  |
| **L** |  | X |  |  | x |  |  |  |  | x |

1)When used as layer 2 or above

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Iteration** | **Layer 1** | **Layer 2** | **Layer 3** | **Comment** |
| 01 | I | C |  | Multiply color to texture |
| 02 | B | J |  | Multiply texture to gradient |
| 03 | K | A | L | Multiply color to texture, then multiply texture |
| 04 | B | L | I | Multiply texture to gradient, multiply texture |
| 05 | J | A | K | Mix color to texture, multiply texture |
| 06 | L | D | H | Multiply gradient to texture, then Mix texture |
| 07 | Base | J | I | Multiply with basematerials as 1st layer |
| 08 | A | K | L | Multiply with basematerials as default object color |

## Negative Material Extension Test Cases

### N\_???\_0601 No Default Color

|  |  |
| --- | --- |
| **Test Scenario Description** | Define a material color on a triangle without the required default material color specified on the object |
| **Pass/Fail Criteria** | 01 to 02 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Triangle with material color, root model object with no default material color  **02** – Triangle with material color, non- root model object with no default material color |

### N\_???\_0602 Duplicate IDs

|  |  |
| --- | --- |
| **Test Scenario Description** | Duplicate material color IDs in model file. |
| **Pass/Fail Criteria** | 01 to 04 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Duplicate colorgroup ID attribute values  **02 –** Duplicate texture2dgroup ID attribute values  **03** – Duplicate texture2d ID attribute values  **04** – Duplicate multiproperties ID attribute values |

### N\_???\_0603 Invalid EXIF Tag

|  |  |
| --- | --- |
| **Test Scenario Description** | Define a non-standard EXIF Tag (value of 2) in a jpeg APP1 marker |
| **Pass/Fail Criteria** | 01– Printer should generate an error |
| **Test Case Iterations** | **01 –** Define and use a texture that utilizes a JPG image with a EXIF colorspace tag (A001) of 99. Note that only 1 and FFFF are typically used (sRGB and unspecified respectively) |

### N\_???\_0604 Multiproperties pids References

|  |  |
| --- | --- |
| **Test Scenario Description** | Verify that the printer rejects multiple colorgroup references multiproperties and ignores the use of another multiproperties as part of a multiproperties pids reference |
| **Pass/Fail Criteria** | 01 to 03 – Printer should generate an error |
| **Test Case Iterations** | **01** – Include two colorgroup references in multiproperties pids  **02 –** Include references to another multiproperties group in multiproperties pids  **03 –** basematerials as layer 2 of multiproperties |

### N\_???\_0605 Invalid Texture Relationship Mapping

|  |  |
| --- | --- |
| **Test Scenario Description** | Description |
| **Pass/Fail Criteria** | 01 to 02 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Define a texture relationship in a .rels file that uses in incorrect type (use a “model” relationship type) in a test file that uses a texture  **02 –** Omit a relationship in .rels to a texture file used in a test case |

### N\_???\_0606 Material Forward Reference

|  |  |
| --- | --- |
| **Test Scenario Description** | Define forward references for material color resources where a prerequisite component has not yet been defined when referenced. |
| **Pass/Fail Criteria** | 01 to 03 – Printer should generate an error |
| **Test Case Iterations** | **01 -** Texture2dgroup in resources before referenced texture2d  **02 -** Multiproperties in resources before referenced texture2dgroup  **03 -** Multiproperties in resources before referenced colorgroup |

### N\_???\_0607 Out of Order Resources

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate that the object element(s) must appear at the end of the list of resources defined in a model part. |
| **Pass/Fail Criteria** | 01 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Define model part with the following ordering of materials in the resources element: colorgroup, texture2d, texture2dgroup, multiproperties, object (This is invalid), texture2d, colorgroup, texture2dgroup, multiproperties, object. There should be no forward references and all materials should be referenced in an object's triangles. Note that this text case can be modeled after the “Positive Ordering” test case in the positive material test cases. |

### N\_???\_0608 Out of Range Color

|  |  |
| --- | --- |
| **Test Scenario Description** | Use invalid color value |
| **Pass/Fail Criteria** | 01 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Use a value outside hex range in color (#FFHFFF) for a color defined in a colorgroup |

### N\_???\_0609 Incorrect Material IDs and Indexes

|  |  |
| --- | --- |
| **Test Scenario Description** | Demonstrate various incorrect ID references |
| **Pass/Fail Criteria** | 01 to 11 – Printer should generate an error |
| **Test Case Iterations** | **01** - Incorrect multiproperties pids reference  **02** - Incorrect texture2dgroup reference to texture2d (textid)  **03** - Incorrect (max+1) index to color in multiproperties(pindices)  **04** - Incorrect (max+1) index to texture in multiproperties (pindices)  **05** - Incorrect (max+1) triangle p1 index to a colorgroup  **06** - Incorrect (max+1) triangle p2 index to a texture  **07** - Incorrect (max+1) triangle p3 index to a multiproperties  **08** - Incorrect (max+1) object pindex index to a colorgroup  **09** - Incorrect (max+1) object pindex index to a texture  **10** - Incorrect (max+1) object pindex index to multiproperties  **11** - Incorrect object pid reference |

### N\_???\_0610 Misc Path and ContentType

|  |  |
| --- | --- |
| **Test Scenario Description** | Miscellaneous invalid values |
| **Pass/Fail Criteria** | 01 to 03 – Printer should generate an error |
| **Test Case Iterations** | **01 –** invalid path to texture file from texture2d  **02 –**Invalid texture2d ContentType attribute value (other than image/jpeg or image/png)  **03 –**Invalid [content\_Types].xml ContentType attribute value (other than image/jpeg or image/png) |

## Positive Production Extension Test Cases

The table below maps the possible permutations of parts and XML objects required to traverse from the StartPart object to the slices. Each set of relationships is defined by a letter, with the green highlight showing the target of the relationship. Test cases will be defined by listing a sequence of relationship letters together, such as “A, F, R, J”. This table will be used to define test cases in subsequent sections.

**Table 1.1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Relationships** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Build Item –**  objectid Only (local) | A | B | C |  |  |  |  |  |  |  |  |  |  |  |  |  | Q | R |  |  |  |  |  |  |  |
| **Root Model Build Item –**  objectid + Path (remote) |  |  |  | D | E |  |  |  |  |  |  |  |  |  |  |  |  |  | S |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Object –** w/o slicestackid Attr  Component -> objectid of local mesh object | A |  |  |  |  | F |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Object –** w/o slicestackid Attr  Component -> remote objectid + path of remote mesh object |  | B |  |  |  |  | G | H |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Object –** with slicestackid Attr  Component -> objectid of local mesh object |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Q |  |  | T | U |  |  |  |  |
| **Root Model Object –** with slicestackid Attr  Component -> remote objectid + path of remote mesh object |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | R |  |  |  |  |  | X | Y |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Object –** with slicestackid Attr  Local mesh |  |  | C |  |  | F |  |  | I | J |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Object –** w/o slicestackid Attr  Local mesh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | T |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Root Model Slice Stack –**  Local Slices (ID) |  |  |  |  |  |  |  |  | I |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Y |
| **Root Model Slice Stack –**  Remote Sliceref (slicestackid + slicepath) |  |  |  |  |  |  |  |  |  | J | K |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Non-Root Model Object –** w/o slicestackid  Component -> objectid of local mesh object |  |  |  | D |  |  | G |  |  |  |  | M |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Non-Root Model Object –** with slicestackid  Component -> objectid of local mesh object |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | S |  |  | V | W |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Non-Root Model Object –** with slicestackid  Local mesh |  |  |  |  | E |  |  | H |  |  |  | M | N | O |  |  |  |  |  |  |  |  |  |  |  |
| **Non-Root Model Object –** w/o slicestackid  Local mesh |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | V |  | X |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Non-Root Model Slice Stack –**  Local Slices (ID) |  |  |  |  |  |  |  |  |  |  | K |  | N |  |  |  |  |  |  |  | U |  | W |  |  |
| **Non-Root Model Slice Stack –**  Remote Sliceref (slicestackid + slicepath) |  |  |  |  |  |  |  |  |  |  |  |  |  | O | P |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Another Non-Root Model Slice Stack –**  Local Slices (ID) |  |  |  |  |  |  |  |  |  |  |  |  |  |  | P |  |  |  |  |  |  |  |  |  |  |

### PP\_701 Object and Slice Mapping

|  |  |
| --- | --- |
| **Test Scenario Description** | Construct sliced 3MF test files that iterate through the possible mapping relationships between Build Items, Objects, Components, and Slice Stacks. This will require slicing a number of individual 3MF files, then concatenating the data together in a single 3MF file. |
| **Conformance Statement ID(s)** | Prod-O1.3, Slice-M1.17, Slice-O1.1, Slice-O1.2, Prod-O1.2 |
| Base Test Object(s) | S11\_cube\_NA.3mf  S11\_octahedron\_NA.3mf  S11\_hex\_pyramid\_NA.3mf  S12\_cylinder\_low.3mf |
| **Pass/Fail Criteria** | 01 to 14 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a 3MF file with relationships “A, F, I” from table 1.1  **02 –** Create a 3MF file with relationships “A, F, J, K” from table 1.1  **03 –** Deleted  **04 –** Create a 3MF file with relationships “B, G, M, N” from table 1.1  **05** **–** Create a 3MF file with relationships “B, G, M, O, P” from table 1.1  **06** **–** Create a 3MF file with relationships “B, H, N” from table 1.1  **07** **–** Create a 3MF file with relationships “B, H, O, P” from table 1.1  **08** ­**–** Create a 3MF file with relationships “C, I” from table 1.1  **09** ­**–** Create a 3MF file with relationships “C, J, K” from table 1.1  **10** **–** Deleted  **11** **–** Create a 3MF file with relationships “D, M, N” from table 1.1  **12** **–** Create a 3MF file with relationships “D, M, O, P” from table 1.1  **13** **–** Create a 3MF file with relationships “E, N” from table 1.1  **14** **–** Create a 3MF file with relationships “E, O, P” from table 1.1 |

The table below maps the possible permutations of parts and XML objects required to traverse from the StartPart object to various mesh representations in the OPC file. Each set of relationships is defined by a letter, with the green highlight showing the target of the relationship. Test cases will be defined by listing a sequence of relationship letters together, such as “A, F, R, J”. This table will be used to define test cases in subsequent sections.

**Table 1.2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Relationships** |  |  |  |  |  |  |  |  |  |
| **Root Model Build Item –**  objectid Only (local) | A | B | C |  |  |  |  |  |  |
| **Root Model Build Item –**  objectid + Path (remote) |  |  |  | D | E |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Root Model Object –**  Component -> objectid of local mesh object | A |  |  |  |  | F |  |  |  |
| **Root Model Object –**  Component -> remote objectid + path of remote mesh object |  | B |  |  |  |  | G | H |  |
| **Root Model Object –**  Local mesh |  |  | C |  |  | F |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Non-Root Model Object -**  Component -> objectid of local mesh object |  |  |  | D |  |  | G |  | M |
| **Non-Root Model Object –**  Local mesh |  |  |  |  | E |  |  | H | M |

### P\_???\_0702 Object Mapping

|  |  |
| --- | --- |
| **Test Scenario Description** | Construct 3MF test files that iterate through the possible mapping relationships between Build Items, Objects, and Components. |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a 3MF file with relationships “A, F” from table 1.2  (root build item -> root object w/component -> root object w/mesh)  **02 –** Create a 3MF file with relationships “B, G, M” from table 1.2  (root build item -> root object w/component ->non-root object w/component -> non-root object w/mesh)  **03** **–** Create a 3MF file with relationships “B, H from table 1.2  (root build item -> root object w/component -> non-root object w/mesh)  **04** ­**–** Create a 3MF file with relationships “C” from table 1.2  (root build item -> root object w/mesh)  **05** **–** Create a 3MF file with relationships “D, M” from table 1.2  (root build item -> non-root object w/component -> non-root object w/mesh)  **06** **–** Create a 3MF file with relationships “E” from table 1.2  (root build item -> non-root object w/mesh) |

### P\_???\_0703 Object Mapping 2

|  |  |
| --- | --- |
| **Test Scenario Description** | Construct 3MF test files that iterate through the possible mapping relationships between Build Items, Objects, and Components. |
| **Pass/Fail Criteria** | 01 to 12 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create a 3MF file with two build items which point directly to root part mesh objects. Modify build item transform so they don't overlap.  **02** **–** Create a 3MF file with two build items which point at two objects that define components, which in turn point at locally defined mesh objects. Modify build item transform so they don't overlap.  **03** **–** Create a 3MF file with two build items which point at two objects that define components, which in turn point at remote parts with mesh objects and via the component path element. Modify build item transform so they don't overlap.  04 **–** Create a 3MF file with build items including one each of the following objects: Local mesh, local mesh via Object component, remote mesh via Object component. Modify build item transform so they don't overlap.  **05** **–** Create 3MF file with two build items pointed at the same local mesh object. Modify build item transform so they don't overlap.  **06** **–** Create a 3MF file with two build items pointed to the same remote part mesh object, modify build item transform so they don't overlap.  **07** **–** Create a 3MF file with an object with two components that contain objectid references to two local root part mesh objects. Modify component so they don't overlap.  **08** **–** Create a 3MF file with an object that points to another object with two components that contain objectid references to locally defined mesh objects. Modify component transform so they don't overlap.  **09** **–** Create a 3MF file with an object with two components which point at remote parts with mesh objects via the component path element. Modify component transform so they don't overlap.  **10** **–** Create a 3MF file with an object with three components that includes one each of the following objects: Local mesh, local mesh via another Object component, remote mesh via path. Modify component transform so they don't overlap.  **11** **–** Create 3MF file with an object with two components pointed at the same local mesh object. Modify build item transform so they don't overlap.  **12** **–** Create a 3MF file with an object with two components pointed to the same remote part mesh object via path. Modify build item transform so they don't overlap. Include model level metadata in non-root model file. |

### PP\_704 Object and Slice Mapping

|  |  |
| --- | --- |
| **Test Scenario Description** | Construct sliced 3MF test files that iterate through the possible mapping relationships between Build Items, Objects, Components, and Slice Stacks. This will require slicing a number of individual 3MF files, the concatenating the data together in a single 3MF file. |
| **Conformance Statement ID(s)** | Prod-O1.3, Slice-M1.17, Slice-O1.1, Slice-O1.2, Prod-O1.2 |
| Base Test Object(s) | S11\_cube\_NA.3mf  S11\_octahedron\_NA.3mf  S11\_hex\_pyramid\_NA.3mf  S12\_cylinder\_low.3mf |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create a 3MF file with a slicestack that contains two Sliceref slicepath objects pointing to remote slice stacks. The composite of the slice stacks should comprise one object.  The following test cases define the slicestackID in an object with component references. The downstream 3D objectID references will not contain a slicestackID. Note that the letters in parenthesis indicate that there is a reference from the object to both the slicestack and the 3D mesh object.  **02–** Create a 3MF file with relationships “Q, (TU)” from table 1.1  **03 –** Create a 3MF file with relationships “R, (XY)” from table 1.1  **04** – Create a 3MF file with relationships “S, (VW)” from table 1.1 |

### P\_???\_0705 Duplicates of Separate Parts

|  |  |
| --- | --- |
| **Test Scenario Description** | Multiple part build using production extensions |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** A test case that use 3 different mesh objects defined in separate parts to build 30 objects, 10 of each via build item element on the build platform. Objects should be positioned both adjacent in XY space, and Stacked in the Z space. |

### P\_???\_0706 Build Item Non-Root Model

|  |  |
| --- | --- |
| **Test Scenario Description** | Build item in a non-root model file |
| **Pass/Fail Criteria** | 01 – Printer should ignore |
| **Test Case Iterations** | **01 –** File with a build item in a non-root model file. Printer should ignore. |

## Negative Production Extension Test Cases

### N\_???\_0801 Incorrect Mapping of IDs and Paths

|  |  |
| --- | --- |
| **Test Scenario Description** | Construct 3MF test files that have incorrect mapping relationships between Build Items, Objects, and Components. The files generated in test case P\_???\_0701\_01 through P\_???\_0701\_18 will be used as a resource for these test cases. Test case developers can select a test case from this group that has the targeted mapping modified as defined below. |
| **Pass/Fail Criteria** | 01 to 09 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Modify existing test case to create an incorrect objectid mapping using relationship “A” defined in table 1.2 in section 5.7  **02 –** Modify existing test case to create an incorrect objectid mapping using relationship “E” defined in table 1.2 in section 5.7  **03 –** Modify existing test case to create an incorrect path mapping using relationship “E” defined in table 1.2 in section 5.7  **04 –** Modify existing test case to create an incorrect objectid mapping using relationship “F” defined in table 1.2 in section 5.7  **05 –** Modify existing test case to create an incorrect objectid mapping using relationship “H” defined in table 1.2 in section 5.7  **06 –** Modify existing test case to create an incorrect path mapping using relationship “H” defined in table 1.2 in section 5.7  **07 –** Modify existing test case to create an incorrect slicestackid mapping using relationship “I” defined in table 1.1 in section 5.7  **08 –** Modify existing test case to create an incorrect slicestackid mapping using relationship “K” defined in table 1.1 in section 5.7  **09 –** Modify existing test case to create an incorrect slicepath mapping using relationship “K” defined in table 1.1 in section 5.7 |

### N\_???\_0802 UUIDs

|  |  |
| --- | --- |
| **Test Scenario Description** | Create a 3MF document file with missing UUID values or duplicate UUIDs. |
| **Pass/Fail Criteria** | 01 to 05 – Printer should generate an error |
| **Test Case Iterations** | **01 –** Missing UUID in Build item  **02 –** Missing UUID in object with local mesh  **03 –** Missing UUID in component  **04 –** Duplicate UUID between two objects  **05 –** Missing UUID in Build element |

### N\_???\_0803 Restricted Mappings

|  |  |
| --- | --- |
| **Test Scenario Description** | The conformance rules restrict the use of path on the component element of non-root model parts. |
| **Pass/Fail Criteria** | 01– Printer should generate an error |
| **Test Case Iterations** | **01 –** Create a 3MF file that contains three 3D model parts. The root model has an object component reference to the second, and the second has an object component reference to the third. |

## Miscellaneous 3MF Test Cases

### P\_???\_0901 Test Synthetic Low Res

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the Synthetic low res and NA Test objects defined in Appendix A |
| **Pass/Fail Criteria** | 01 to 12 – Printer should process correctly |
| **Test Case Iterations** | **01 –** S11\_Cube  **02 –** S11\_Cube\_Fillet\_Low  **03 –** S11\_Dodecahedron\_NA  **04 –** S11\_Hex\_Pyramid\_NA  **05 –** S11\_Octahedron\_NA  **06 –** S11\_Pentagon\_Prism\_NA  **07 –** S11\_Rectangle\_Pyramid\_NA  **08 –** S11\_Cone\_Low  **09 –** S11\_Cylinder\_Low  **10 –** S11\_Ellipsoid\_Low  **11 –** S12\_Sphere\_Low  **12 –** S12\_Torus\_Low |

### P\_???\_0902 Synthetic High Res

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the Synthetic high res Test objects defined in Appendix A |
|  | N/A |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01 –** S11\_Cube\_Fillet\_High  **02 –** S12\_Cone\_High  **03 –** S12\_Cylinder\_High  **04 –** S12\_Ellipsoid\_High  **05 –** S12\_Sphere\_High  **06 –** S12\_Torus\_High |

### P\_???\_0903 Natural Low Res and NA

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the Natural low res and NA Test objects defined in Appendix A |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01 –** N22\_ChessHorse\_Low  **02 –** N23\_Deer\_Low  **03 –** N32\_Alligator\_228\_Low   1. **–** N32\_Shell\_Low   **06-** N33\_Duck\_NA |

### P\_???\_0904 Natural High Res

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the Natural high res Test objects defined in Appendix A |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01 –** N22\_ChessHorse\_High  **02 –** N23\_Deer\_High  **03 –** N32\_Alligator\_228\_High  **04 –** N32\_Shell\_High |

### P\_???\_0905 Real World Low Res and NA

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the Real World low res and NA Test objects defined in Appendix A |
| **Pass/Fail Criteria** | 01 to 13 – Printer should process correctly |
| **Test Case Iterations** | **01 –** M11\_Box\_NA  **02 –** M11\_Snorkel\_Low  **03 –** M11\_Stereographic\_Maze\_Low  **04 –** M11\_Ventilated Build Platform\_low  **05 –** M12\_Extruder\_Bowden\_Adapterc\_low  **06 –** M12\_role\_drum\_NA  **07 –** M12\_SW\_Extruder-Hinged-Block\_low  **08 –** M12\_Tristruder\_18mm\_Probe\_MouN\_???\_0low  **09 –** M21\_flex\_coupler\_NA  **10 –** M21\_headphone\_rest\_low  **12 –** M21\_stereographic\_flat\_math2\_low  **13 –** M22\_FPV\_Pod\_Half\_NA |

### P\_???\_0906 Real World High Res

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the Real World high res Test objects defined in Appendix A |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01 –** M11\_Snorkle\_high  **02 –** M11\_Ventilated Build Platform\_high  **03 –** M12\_Extruder\_Bowden\_Adapterc\_high  **04 –** M12\_SW\_Extruder-Hinged-Block\_high  **05 –** M12\_Tristruder\_18mm\_Probe\_MouN\_???\_0high  **06 –** M21\_headphone\_rest\_high |

### P\_???\_0907 Assembly Low Res and NA

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF files for each of the xx Assembly and NA Test objects defined in Appendix A |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01 ­­–** chainassembly\_low  **02 –** octohedron5\_NA  **03 –** raindrop\_low  **04 –** randomplacemeN\_???\_0NA |

### P\_???\_0909 Stress Tests

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate 3MF that may stress the resource or implementation boundaries of the printer |
| **Pass/Fail Criteria** | 01 to 08 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a small cube (8x8x8mm) object. Replicate both the object 1000 times within the root model part. Create a thumbnail for each object and reference in the object element and the root model .rels part. This test will exercise the following stress points:   * Number of items in a relationship file (1000) * Number of object elements in a resource element (1000) * Number of build items in a build element (1000)   **02** – Create a sliced 3D part that utilizes most of the allowable x axis build space, with a slice thickness of 80 microns. Divide the slicestack part into 1000 separate part slices and add the appropriate slicesref pointers to each of the parts to a single slicestack. This test will exercise the following stress points:  **03**– Create a 3D part and use 180-character long string that contains Cyrillic and Kanjii characters in metadata content, and the max value of 2147483647 in the following places in the XML:   * ID attribute value for an object * Partnumber attribute value object element   **04** – create a 3D part with characteristics likely to require more than 10,000 vectors in a single polygon to render a single layer. Use 5,000 small cylinders arranged as 100 spokes, each consisting of 50 slightly overlapping adjacent cylinders.  **05** – Create a 3D part that when slices will produce will produce 10,000 polygons for a single slice  **06** – create a 3D part with characteristics likely to require more than 10,000 separate polygons to render a single layer. Use a matrix of 10,000 small non overlapping rectangular objects  **07** – Leverage the test case defined in 01 above and place each of the 1000 objects in a separate non-root model file. Reference these objects from the build items using the path attribute  **08** – Leverage the test case define in 01 above and place each of the 1000 objects in a separate non-root model file. Reference these objects from the component path attributes, then have the build items point to the object containing the components. |

### P\_???\_0910 Transform Matrices

|  |  |
| --- | --- |
| **Test Scenario Description** | Modify the allowable transform matrix elements for a 3MF file |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
|  | **01 –** Render the same object multiple times using each of the various build item transforms. For tests with no slice stack, elements to modify include m00, m01, m02, m10, m11, m12, m20, m21m m22 m30, m31, and m32 in a build item transform. For tests that include slice stack data, modify m00, m01, m10, m11, m30, m31, and m32  **02 –** Render the same object multiple times using each of the various component item transforms. For tests with no slice stack, elements to modify include Modify m00, m01, m02, m10 m11, m12, m20, m21m m22 m30, m31, and m32 in a component item transform. For tests that include slice stack data, modify m00, m01, m10, m11, m30, m31, and m32  **03** – Use all 12 transform matrix elements (7 elements for tests with slice stack data) in a single build item transform such the impact of each element is obvious  **04** – Use all 12 transform matrix elements (7 elements for tests with slice stack data) in a single component transform such the impact of each element is obvious  **05** – Create a file with a build item pointing to an object with components, then an object with mesh. Modify the transforms in the build item and each of the object components to create a cascading effect using all 12 transform elements (7 elements for tests with slice stack data) in both the build item and component transforms.  **06 –** Create a file where the transform attribute is used in a non-root model component element |

### P\_???\_0911 Maze Geometry

|  |  |
| --- | --- |
| **Test Scenario Description** | A 3MF file that will result in a more complex geometry |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create an object that represents a complex maze involving a large number of vector variations |

### P\_???\_0912 XY Axis Positioning

|  |  |
| --- | --- |
| **Test Scenario Description** | Objects around perimeter of print bed |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create 3MF file with a number of object positioned around the periphery of the allowable XY axis |

### P\_???\_0913 Overlapping Objects

|  |  |
| --- | --- |
| **Test Scenario Description** | Create 3mF parts that are overlapping |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create 3MF file with a number of partially overlapping objects that coexist in the same XY plane. |

### P\_???\_0914 namespace prefixes

|  |  |
| --- | --- |
| **Test Scenario Description** | Modify namespace prefixes of extensions so they are something other than “m” and ‘p” for material and production |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Modify a simple 3MF file such that the namespace prefix used for the material and production extensions is something other than “m” or “p” by modifying the xmlns declarations in the model element. Also update the prefixes used in requiredextensions to match the new prefixes |

### P\_???\_0915 Object Pointers

|  |  |
| --- | --- |
| **Test Scenario Description** | A 3MF test job with various object relationships |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a 3MF file that uses Build item path references to two separate parts containing mesh objects. |

## Positive 3MF Slice Extension Test Cases

### P\_???\_1501 Meshresolution Attribute

|  |  |
| --- | --- |
| **Test Scenario Description** | Valid enumerations of meshresolution |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Use fullres meshresolution attribute  **02 –** Use lowres meshresolution attribute  **03** – Omit meshresolution from the model element |

### P\_???\_1502 Transform Matrices

|  |  |
| --- | --- |
| **Test Scenario Description** | Modify the allowable transform matrix elements for a sliced 3MF file (i.e. planar transformation) |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Modify M0, M11, M30, M31, and M32 in a build item transform  **02 –** Modify M0, M11, M30, M31, and M32 in a component transform  **03 –** Modify M0, M01, M10, M11, and M32 in a build item transform  **04 –** Modify M0, M01, M10, M11, and M32 in a component transform  **05** – Create a sliced file with a build item pointing to an object with components, then an object with mesh. Add or modify the transforms in the build item and each of the object components to create a cascading effect using transform elements M0, M01, M10, M11, M30, M31, and M32. |

### P\_???\_1503 Slice Increments

|  |  |
| --- | --- |
| **Test Scenario Description** | Vary Z-axis slice increments |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Generate 3MF file with small slice increments (10 microns)  **02 –** Generate 3MF file with large slice increments (2mm)  **03 –** Generate 3MF file with discontinuous slice increments between 80 and 500 microns |

### P\_???\_1504 Multiple Slicestack References, Mismatched Ztop and Zbottom

|  |  |
| --- | --- |
| **Test Scenario Description** | Multiple sliceref’s in a single slicestack |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create scenario where there are two Sliceref slicestack references, with a mismatch between the last ztop and the second Sliceref zbottom. |

### P\_???\_1505 Polygon Definition with Positive Fill Rule

|  |  |
| --- | --- |
| **Test Scenario Description** | Polygon positive fill rule permutations |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | Test case with Polygon definition requiring application of the positive fill rule. Scenario should include each of the positive file rule example shown in section 4.1.1 of the core specification.  **01 –** Example 1  **02 –** Example 2  **03 –** Example 3 |

### P\_???\_1506 Ignore Object Level Material Mapping

|  |  |
| --- | --- |
| **Test Scenario Description** | Scenario that uses basematerial references in polygon segment references. Printer should ignore. |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a slice stack that uses the segment p1, p2, and pid. Define two basematerials and multiple colors within each base materials. Use the attributes on several segments to override the object level material mapping. Printer should ignore attributes. |

### P\_???\_1507 Multiple Polygons Representing a Slice

|  |  |
| --- | --- |
| **Test Scenario Description** | 3MF object where the sliced data has multiple polygons in a single slice layer. |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Multiple polygons representing a slice |

### P\_???\_1508 Collapsing Proximal Vertices

|  |  |
| --- | --- |
| **Test Scenario Description** | Identical and near identical vertices used in a polygon definition |
| **Pass/Fail Criteria** | 01 to 02 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create a series of 2D vertices that are almost identical, then use those vertices in the polygon segment definition  **02** **–** Create a series of 2D vertices that are identical, then use those vertices in the polygon segment definition |

### P\_???\_1509 Small of number of vertices and polygons

|  |  |
| --- | --- |
| **Test Scenario Description** | Variations in number of vertices and polygons |
| **Pass/Fail Criteria** | 01 to 04 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Slicestack layers with 3 segments elements per polygon  **02** **–** Slicestack layer with approximately 400 segments elements per polygon  **03** **–** Slicestack layer with 1 polygon  **04** **–** Slicestack layer with approximately 100 polygons |

### P\_???\_1510 Complex 2D Geometries

|  |  |
| --- | --- |
| **Test Scenario Description** | A 3MF file that will result in a more complex sliced 2D geometry |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Slice an object that represents a complex maze involving a large number of 2D vector variations |

### P\_???\_1511 Z-Axis Offsets

|  |  |
| --- | --- |
| **Test Scenario Description** | 3MF files with varying slice thicknesses |
| **Pass/Fail Criteria** | 01 to 06 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Slice an object at 100 microns  **02** **–** Slice an object at 200 microns  **03** **–** Slice an object at 300 microns  **04** **–** Slice an object at 400 microns  **05** **–** Slice an object at 500 microns  **06** **–** Object sliced with 20% segments of slicestack at 100, 200, 300, 400, and 500 microns |

### P\_???\_1512 XY Axis Positioning

|  |  |
| --- | --- |
| **Test Scenario Description** | Objects around perimeter of print bed |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create 3MF file with a number of object positioned around the periphery of the allowable XY axis |

### P\_???\_1513 Multiple Slice Stacks

|  |  |
| --- | --- |
| **Test Scenario Description** | Create sliced 3mF files that contain multiple slice stacks |
| **Pass/Fail Criteria** | 01 to 03 – Printer should process correctly |
| **Test Case Iterations** | **01** **–** Create 3MF file with a number of non-overlapping objects that coexist in the same XY plane. Independent slice stacks for each object should be in root model part.  **02** **–** Create 3MF file with a number of non-overlapping objects that coexist in the same XY plane. Independent slice stacks for each object should exist in separate slice stack parts, each pointed to using a sliceref element in the root model part.  **03** **–** Create 3MF file with a number of partially overlapping objects that coexist in the same XY plane. Independent slice stacks for each object should exist in separate slice stack parts, each pointed to using a sliceref element in the root model part. |

### P\_???\_1514 Duplicate IDs

|  |  |
| --- | --- |
| **Test Scenario Description** | Duplicate slicestack ID values |
| **Pass/Fail Criteria** | 01 to 02– Printer should process correctly |
| **Test Case Iterations** | **01 –** Duplicate slicestack IDs split between a root model file and a non-root model file  **02** – Create a sliced 3MF package where two mesh objects that reside as separate parts use the same ID |

### P\_???\_1515 namespace prefixes

|  |  |
| --- | --- |
| **Test Scenario Description** | Modify namespace prefixes of extensions so they are something other than “s” and ‘p” for production and slice |
| **Pass/Fail Criteria** | 01 – Printer should process correctly |
| **Test Case Iterations** | **01 –** Modify a simple slice file such that the namespace prefix used for the slice and production extensions is something other than “s” or “p” by modifying the xmlns declarations in the model element. Also update the prefixes used in requiredextensions to match the new prefixes |

### P\_???\_1516 Slicestack Object Pointers

|  |  |
| --- | --- |
| **Test Scenario Description** | A 3MF test job with various slicestack to object relationships |
| **Pass/Fail Criteria** | 01 to 02– Printer should process correctly |
| **Test Case Iterations** | **01 –** Create a sliced 3MF file that uses Build item path references to two separate parts containing mesh objects. Those two mesh object parts should point to a single model file containing two separate slicestacks in the same file, with each object pointed at a separate stack  **02 –** Create a sliced 3MF file that uses Build item path references to two separate parts containing mesh objects. Those two mesh object parts should point to a single model file containing one slicestack in the file, with each object pointed at the same slicestack. Modify build item transform so objects do not overlap |

### P\_???\_1517 Polygon Slice

|  |  |
| --- | --- |
| **Test Scenario Description** | Odd polygon definition |
| **Pass/Fail Criteria** | 01 t0 02– Printer should ignore |
| **Test Case Iterations** | **01 –** Closed no area (Overlapping 2 segments)  **02** – No polygons |

## Negative Slice Extension Test Cases

### N\_???\_1601 Transform Matrices

|  |  |
| --- | --- |
| **Test Scenario Description** | Invalid values in transform matrix |
| **Pass/Fail Criteria** | 01 to 05 – Printer should generate error |
| **Test Case Iterations** | **01 –** Use a non-zero value in a transform for M02 in a build Item transform  **02 –** Use a non-zero value in a transform for M12 in a build Item transform  **03 –** Use a non-zero value in a transform for M20 in a build Item transform  **04 –** Use a non-zero value in a transform for M21 in a component transform  **05 –** Use a value other than 1 for M22 in a component transform |

### N\_???\_1604 Locally Defined Slice Stack and Sliceref

|  |  |
| --- | --- |
| **Test Scenario Description** | Invalid Sliceref self-reference |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Have Sliceref point to the same part that contains the Sliceref statement referring to a locally defined slice stack |

### N\_???\_1605 Two Layered Slicestack Reference Abstraction

|  |  |
| --- | --- |
| **Test Scenario Description** | Invalid 2 layers of abstraction in slicestack Sliceref references |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Define slicestack references with two layers of abstraction from the original slicestack |

### N\_???\_1606 Ztop Smaller Than Zbottom

|  |  |
| --- | --- |
| **Test Scenario Description** | Ztop Smaller Than Zbottom in sliced file |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Define slice with ztop smaller the zbottom value |

### N\_???\_1607 Ztop Lower Than Preceding Value

|  |  |
| --- | --- |
| **Test Scenario Description** | Ztop lower than preceding value in sliced file |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** ztop slice value that is lower than the preceding value |

### N\_???\_1608 Non-Distinct v2 Attributes

|  |  |
| --- | --- |
| **Test Scenario Description** | Non-distinct v2 references in polygon definition |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Sequential segments with the same v2 attribute index |

### N\_???\_1609 Polygon Slice Descriptions

|  |  |
| --- | --- |
| **Test Scenario Description** | Invalid polygon definitions |
| **Pass/Fail Criteria** | 01 to 03 – Printer should generate error |
| **Test Case Iterations** | **01** **–** Polygon with a single point  **02 –** Open Polygon |

### N\_???\_1610 Unique Slicestack ID

|  |  |
| --- | --- |
| **Test Scenario Description** | Duplicate slicestack ID values |
| **Pass/Fail Criteria** | 01– Printer should generate error |
| **Test Case Iterations** | **01** **–** Duplicate slicestack IDs in same local file |

### N\_???\_1612 Overlapping Slicestacks

|  |  |
| --- | --- |
| **Test Scenario Description** | Generate a sliced 3MF file with overlapping slicestacks |
| **Pass/Fail Criteria** | 01 – Printer should generate error |
| **Test Case Iterations** | **01 –** Scenario where there two slicerefs in a slice stack, but the Z-axis alignment of the vertically adjacent slicestacks referenced are overlapping |

# Appendix A - Test Object Library

Each 3MF test case will reference one or more test objects as defined in the following tables. Objects were generated from the following sources:

* Created by QualityLogic Staff using SolidWorks and exported to STL files
* SolidWorks SLDPRT files found on Thingiverse.com with Creative Commons license allowing commercial use, exported to STL files
* STL file found on Thingiverse.com with Creative Commons license allowing commercial use

Objects where we have the SLDPRT file provide the greatest control over the mesh resolution, and a SolidWorks icon is shown in the tables below where we have access to this file type.

The object names define the nature and complexity of the object using the following format:

**ABC\_DDDD\_EEEE**

A = Object type

* S – Synthetic object, primarily simple geometric objects
* N – Natural objects, things found in nature
* M – Man-made objects, typically manufactured
* A – Assembly objects, sets of objects that comprise an assembly

B = Level of Detail where 1 is the least detailed and 3 is the most detailed

C = Level of curvature – Impacts triangle generation, 1 is least, 3 is most

DDDD = Object name

EEEE = Triangle count (High, Low, NA), differentiate same file captured at differing triangle densities

The categorization and ratings of objects are very subjective, with the only intent of these ratings being to assist in the selection of objects for test files. Assembly objects will be defined or constructed from existing objects as the requirements for these objects become clearer.

**Synthetic Objects**

|  |  |  |  |
| --- | --- | --- | --- |
| Object Name | Triangles | Image | Size (x, y, z)  Attribution Link |
| S11\_cube\_NA | 12 | C:\Users\cory\Documents\archiveobjects\thumbs\cube.png | 100, 100, 100 mm |
| S11\_cube\_fillet\_high  S11\_cube\_fillet\_low | 2,092  84 | C:\Users\cory\Documents\archiveobjects\thumbs\button.png | 100, 100, 100 mm |
| S11\_dodecahedron\_NA | 36 | C:\Users\cory\Documents\archiveobjects\thumbs\dodecahedron.png | 130, 137, 110 mm |
| S11\_hex\_pyramid\_NA | 10 | C:\Users\cory\Documents\archiveobjects\thumbs\hexpyramid.png | 76, 65.82, 90.42 mm |
| S11\_octahedron\_NA | 8 | C:\Users\cory\Documents\archiveobjects\thumbs\octahedron.png | 57.74, 57.74, 100 mm |
| S11\_pentagon\_prism\_NA | 16 | C:\Users\cory\Documents\archiveobjects\thumbs\pentagonprism.png | 97.08, 92.33, 100 mm |
| S11\_rectangle\_pyramid\_NA | 8 | C:\Users\cory\Documents\archiveobjects\thumbs\rectanglepyramid.png | 100, 40, 34.64 mm |
| S12\_cone\_high  S12\_cone\_low | 1,440  62 | C:\Users\cory\Documents\archiveobjects\thumbs\cone.png | 55, 55, 95 mm |
| S12\_cylinder\_high  S12\_cylinder\_low | 2,880  120 | C:\Users\cory\Documents\archiveobjects\thumbs\cyllinder.png | 50, 50, 100 mm |
| S12\_ellipsoid\_high  S12\_ellipsoid\_low | 545,760  2530 | C:\Users\cory\Documents\archiveobjects\thumbs\ellipsoid.png | 100, 100, 30 mm |
| S12\_sphere\_high  S12\_sphere\_low | 516,960  2,352 | C:\Users\cory\Documents\archiveobjects\thumbs\sphere.png | 100, 100, 100 mm |
| S12\_torus\_high  S12\_torus\_low | 26,642  2,700 | C:\Users\cory\Documents\archiveobjects\thumbs\torus.png | 100, 100, 20 mm |

**Natural Objects**

|  |  |  |  |
| --- | --- | --- | --- |
| Object Name | Triangles | Image | Size (x, y, z)  Attribution Link |
| N22\_ChessHorse\_high  N22\_ChessHorse\_low | 44,612  7,126 | C:\Users\cory\Documents\archiveobjects\thumbs\chesshorse.png | 55.24, 67.07, 100 mm  [**chessHorse**](http://www.thingiverse.com/thing:994236) by [**jbarrettoda**](http://www.thingiverse.com/jbarrettoda) is licensed under the[**Creative Commons - Attribution**](http://creativecommons.org/licenses/by/3.0/) license. |
| N23\_Deer\_high  N23\_Deer\_low | 49,324  9,864 | C:\Users\cory\Documents\archiveobjects\thumbs\deer.png | 26.67, 75, 71.41 mm  [**Deer**](http://www.thingiverse.com/thing:182123) by [**YahooJAPAN**](http://www.thingiverse.com/YahooJAPAN) is licensed under the [**Creative Commons - Attribution**](http://creativecommons.org/licenses/by/3.0/) license |
| N32\_alligator\_228\_high  N32\_alligator\_228\_low | 209,652  56,862 | C:\Users\cory\Documents\archiveobjects\thumbs\alligator.png | 85, 121, 25  [**Alligator**](http://www.thingiverse.com/thing:260134) by [**willie**](http://www.thingiverse.com/willie) is licensed under the [**Creative Commons - Public Domain Dedication**](http://creativecommons.org/publicdomain/zero/1.0/) license |
| N32\_shell\_high  N32\_shell\_low | 250,498  62,520 | C:\Users\cory\Documents\archiveobjects\thumbs\shell.png | 64.75, 75, 56.73 mm  Unknown author |
| N33\_Duck\_NA | 11,578 | C:\Users\cory\Documents\archiveobjects\thumbs\duck.png | 84.11, 68.58, 53 mm  [**Duck**](http://www.thingiverse.com/thing:31627) by [**Roboduck**](http://www.thingiverse.com/Roboduck) is licensed under the [**Creative Commons - Attribution**](http://creativecommons.org/licenses/by/3.0/) license |

**Man-Made, Manufactured Objects**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Triangles** | **Image** | **Size (x, y, z)**  **Attribution Link** |
| M11\_box\_NA | 41,022 | C:\Users\cory\Documents\archiveobjects\thumbs\box.png | 170, 11, 48 mm  [**Geeetech GT2560 housing**](http://www.thingiverse.com/thing:1279978) by [**lukie80**](http://www.thingiverse.com/lukie80) is licensed under the [**Creative Commons - Attribution**](http://creativecommons.org/licenses/by/3.0/) license |
| M11\_Snorkle\_high  M11\_Snorkle\_low | 1,524,106  8,838 | C:\Users\cory\Documents\archiveobjects\thumbs\snorkle.png | 150, 17, 104 mm  [**FJ Cruiser Snorkel Grill**](http://www.thingiverse.com/thing:566097) by [**LordNova2**](http://www.thingiverse.com/LordNova2) is licensed under the [**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/) license |
| M11\_stereographic\_maze\_lowres\_NA | 16,524 | C:\Users\cory\Documents\archiveobjects\thumbs\mathNA.png | 199, 199, 175 mm  [**Customizable stereographic projection lowres**](http://www.thingiverse.com/thing:296967) by[**threonin**](http://www.thingiverse.com/threonin) is licensed under the [**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/) license |
| M11\_Ventilated Build Platform\_high  M11\_Ventilated Build Platform\_low | 210,302  12,638 | C:\Users\cory\Documents\archiveobjects\thumbs\buildplatform.png | 100, 100, 6 mm  [**Ventilated Build Platform**](http://www.thingiverse.com/thing:1551) by [**deherzog**](http://www.thingiverse.com/deherzog) is licensed under the [**Public Domain**](http://creativecommons.org/licenses/publicdomain/) license |
| M12\_Extruder\_Bowden\_Adapterc\_high  M12\_Extruder\_Bowden\_Adapterc\_low | 20,662  792 | C:\Users\cory\Documents\archiveobjects\thumbs\eba.png | 115, 19, 20 mm  [**bowden capable gregs extruder with adapters**](http://www.thingiverse.com/thing:66611) by[**nicksears**](http://www.thingiverse.com/nicksears) is licensed under the [**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/) license |
| M12\_role\_drum\_NA | 35,232 | C:\Users\cory\Documents\archiveobjects\thumbs\roledrum.png | 295, 296, 100 mm  [**Rouleau de PLA / ABS dispenser roll**](http://www.thingiverse.com/thing:12349) by [**Alf\_Arobase**](http://www.thingiverse.com/Alf_Arobase)is licensed under the [**GNU - GPL**](http://creativecommons.org/licenses/GPL/2.0/) license |
| M12\_SW\_Extruder-Hinged-Block\_high  M12\_SW\_Extruder-Hinged-Block\_low | 77,148  792 | C:\Users\cory\Documents\archiveobjects\thumbs\ehb.png | 125, 66, 28 mm  [**bowden capable gregs extruder with adapters**](http://www.thingiverse.com/thing:66611) by[**nicksears**](http://www.thingiverse.com/nicksears) is licensed under the [**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/) license |
| M12\_Tristruder\_18mm\_Probe\_MouN\_???\_0high  M12\_Tristruder\_18mm\_Probe\_MouN\_???\_0low | 12,856  7,384 | C:\Users\cory\Documents\archiveobjects\thumbs\tristruder.png | 82, 62, 38 mm  [**Prusa i3 Tristruder with 18mm Probe Mount**](http://www.thingiverse.com/thing:1582399) by[**insapio**](http://www.thingiverse.com/insapio) is licensed under the [**Creative Commons - Attribution**](http://creativecommons.org/licenses/by/3.0/) license |
| M21\_flex\_coupler\_NA | 14,558 | C:\Users\cory\Documents\archiveobjects\thumbs\flexcoupler.png | 50, 50, 100 mm  [**Flex Coupler with Embedded Hardware**](http://www.thingiverse.com/thing:896836) by[**chayesSAS**](http://www.thingiverse.com/chayesSAS) is licensed under the [**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/) license |
| M21\_headphone\_rest\_high  M21\_headphone\_rest\_low | 189,060  1,816 | C:\Users\cory\Documents\archiveobjects\thumbs\headphonerest.png | 91, 42, 65 mm  [**HyperX Cloud Headset rest/stand**](http://www.thingiverse.com/thing:902517) by [**thatcloudguy**](http://www.thingiverse.com/thatcloudguy) is licensed under the [**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/) license |
| M22\_FPV\_Pod\_Camera\_Plate\_NA | 17,912 | C:\Users\cory\Documents\archiveobjects\thumbs\cameraplate.png | 81.87, 35.03, 100 mm  [**FPV Pod (Pan) - Hawkeye 1700 - GoPro/Xiaomi Yi + Board Cam Mount**](http://www.thingiverse.com/thing:985040) by [**Bl0K3**](http://www.thingiverse.com/Bl0K3) is licensed under the[**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/)license |
| M22\_FPV\_Pod\_Half\_NA | 47,476 | C:\Users\cory\Documents\archiveobjects\thumbs\podhalf.png | 80, 72, 155 mm  [**FPV Pod (Pan) - Hawkeye 1700 - GoPro/Xiaomi Yi + Board Cam Mount**](http://www.thingiverse.com/thing:985040) by [**Bl0K3**](http://www.thingiverse.com/Bl0K3) is licensed under the[**Creative Commons - Attribution - Share Alike**](http://creativecommons.org/licenses/by-sa/3.0/)license |

**Assembly Objects**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Triangles** | **Image** | **Size (x, y, z)**  **Attribution Link** |
| chainassembly\_low | 13,020 | C:\Users\cory\Documents\archiveobjects\thumbs\chainassm.png | 173, 121, 223 mm |
| octohedron5\_NA | 40 | C:\Users\cory\Documents\archiveobjects\thumbs\octahedron5.png | 122, 95, 129 mm |
| raindrop\_low | 25,350 | C:\Users\cory\Documents\archiveobjects\thumbs\Untitled-1.png | 270, 382, 319 mm |
| randomplacemeN\_???\_0NA | 2608 | C:\Users\cory\Documents\archiveobjects\thumbs\randomplacement.png | 209, 191, 270 mm |

# Appendix B – Color and Texture Tables

The following textures and colors will be referenced in test case scenarios to simply the definition of test case intent as it applies to the 3MF Material and Properties Extension.

**Standard Colors (from 3D Builder)**

|  |  |  |
| --- | --- | --- |
| **Name** | **RGB** | **Swatch** |
| White | #FFFFFF  #FFFFFFFF |  |
| Black | #000000  #000000FF |  |
| Gray | #808080  #808080FF |  |
| Dark red | #880015  #880015FF |  |
| Red | #EC1B23  #EC1B23FF |  |
| Orange | #FF7F25  #FF7F25FF |  |
| Yellow | #FEF100  #FEF100FF |  |
| Lime | #B5E61D  #B5E61DFF |  |
| Green | #21BB4C  #21BB4CFF |  |
| Turquoise | #00A0E8  #00A0E8FF |  |
| Indigo | #3E47CB  #3E47CBFF |  |
| Purple | #A349A4  #A349A4FF |  |

**Saturated Colors**

|  |  |  |
| --- | --- | --- |
| **Name** | **RGB** | **Swatch** |
| Rgb | #FF0000  #FF0000FF |  |
| rGb | #00FF00  #00FF00FF |  |
| rgB | #0000FF  #0000FFFF |  |
| RGb | #FFFF00  #FFFF00FF |  |
| RgB | #FF00FF  #FF00FFFF |  |
| rGB | #00FFFF  #00FFFFFF |  |
| RGB | #FFFFFF  #FFFFFFFF |  |
| Rgb | #000000  #000000FF |  |

\*Upper case RGB = FF, Lower Case rgb = 00

**Gradient Combinations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **P1** | **P2** | **P3** |  |
| GradieN\_???\_01 | White | Black | Gray |  |
| GradieN\_???\_02 | Dark red | Red | Orange |  |
| GradieN\_???\_03 | Yellow | Lime | Green |  |
| GradieN\_???\_04 | Turquoise | Indigo | Purple |  |
| GradieN\_???\_05 | Rgb | rGb | rgB |  |
| GradieN\_???\_06 | RGb | RgB | rGB |  |
| GradieN\_???\_07 | RGB | Rgb | rgb |  |
| GradieN\_???\_08 | RGB | rGb | rgb |  |
| GradieN\_???\_09 | RGB | rgB | rgb |  |

**Small Texture Swatches (from 3D Builder)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name/File** | **Size** | **Swatch** | **Swatch W/Alpha** |
| brmarble.jpg  brmarble.png  brmarble\_A.png | 128 X 128 |  |  |
| droplets.jpg  droplets.png  droplets\_A.png  droplets\_mono.png  droplets\_mono\_A.png | 128 X 128 |  |  |
| grmarble.jpg  grmarble.png  grmarble\_A.png | 128 X 128 |  |  |
| oak.jpg  oak.png  oak\_A.png  oakNumbers.png | 128 X 128 | C:\Temp\oakNumbers1.png |  |
| pitissue.jpg  pitissue.png  pitissue\_A.png | 128 X 128 |  |  |
| purmesh.jpg  purmesh.png  purmesh\_A.png | 128 X 128 |  |  |
| quads.jpg  quads.png  quads\_A.png | 720 X 720 |  |  |

PNG without alpha are 24 bit, with alpha are 32 bit

**Large Texture Images (Pixabay – No Attribution Required)**

|  |  |  |
| --- | --- | --- |
| **Name/File** | **Size/**  **Original File Name** | **Photo** |
| photo\_1.jpg  photo\_1.png  photo\_1.png\_48  photo\_1.png\_16  16/48 = bit depth | 1280 X 833  background-1655938\_1280.jpg |  |
| photo\_2.jpg  photo\_2.png | 1280 X 829  background-2090828\_1280.jpg |  |
| photo\_3.jpg  photo\_3.png | 1280 X 1280  checkerboard-1943243\_1280.png |  |
| photo\_4.jpg  photo\_4.png | 1280 X 720  complex-664440\_1280.jpg |  |
| photo\_5.jpg  photo\_5.jpg | 1280 X 853  snail-shells-65358\_1280.jpg |  |
| photo\_6.jpg  photo\_6.png | 960 X 1280  substances-43315\_1280.jpg |  |

|  |  |  |
| --- | --- | --- |
| **Name/File** | **Size/**  **Original File Name** | **Photo** |
| Woman\_200.jpg  Woman­­­\_3000.jpg | 200 X 200  3000 X 3000  woman-101542.jpg |  |
| Horse\_350.png  Horse\_4000.png | 350 X 350  4000 X 4000  Created in Paint Program |  |

**Public Domain PngSuite – Basic Formats**

|  |
| --- |
| http://www.schaik.com/pngsuite2011/basn0g01.png     basn0g01 - black & white  http://www.schaik.com/pngsuite2011/basn0g02.png     basn0g02 - 2 bit (4 level) grayscale  http://www.schaik.com/pngsuite2011/basn0g04.png     basn0g04 - 4 bit (16 level) grayscale  http://www.schaik.com/pngsuite2011/basn0g08.png     basn0g08 - 8 bit (256 level) grayscale  http://www.schaik.com/pngsuite2011/basn0g16.png     basn0g16 - 16 bit (64k level) grayscale  http://www.schaik.com/pngsuite2011/basn2c08.png     basn2c08 - 3x8 bits rgb color  http://www.schaik.com/pngsuite2011/basn2c16.png     basn2c16 - 3x16 bits rgb color  http://www.schaik.com/pngsuite2011/basn3p01.png     basn3p01 - 1 bit (2 color) paletted  http://www.schaik.com/pngsuite2011/basn3p02.png     basn3p02 - 2 bit (4 color) paletted  http://www.schaik.com/pngsuite2011/basn3p04.png     basn3p04 - 4 bit (16 color) paletted  http://www.schaik.com/pngsuite2011/basn3p08.png     basn3p08 - 8 bit (256 color) paletted  http://www.schaik.com/pngsuite2011/basn4a08.png     basn4a08 - 8 bit grayscale + 8 bit alpha-channel  http://www.schaik.com/pngsuite2011/basn4a16.png     basn4a16 - 16 bit grayscale + 16 bit alpha-channel  http://www.schaik.com/pngsuite2011/basn6a08.png     basn6a08 - 3x8 bits rgb color + 8 bit alpha-channel  http://www.schaik.com/pngsuite2011/basn6a16.png     basn6a16 - 3x16 bits rgb color + 16 bit alpha-channel |

**Multi-Property Sets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Layer 1** | **Layer 2** | **Layer 3** |
| MultiProp\_1 | Solid Color  Lime | Texture  brmarble\_A.png |  |
| MultiProp\_2 | Gradient3 | Texture  grmarble\_A.png |  |
| MultiProp\_3 | Texture  oak.jpg | Solid Color  Yellow  Alpha #7f |  |
| MultiProp\_4 | Texture  photo\_3.jpg | GradieN\_???\_05  Alpha #00 on P1 |  |
| MultiProp\_5 | Texture  photo\_6.png | Texture  droplets\_A.png |  |
| MultiProp\_6 | GradieN\_???\_07 | Texture  droplets\_A.png | Texture  pitissue\_A.png |
| MultiProp\_7 | Texture  photo\_4.jpg | Solid Color  Orange  Alpha #7F | Texture  oak\_A.png |
| MultiProp\_8 | Texture  purmesh.png | Texture  brmarble\_A.png | GradieN\_???\_06  Alpha of #3f on P1, P2, P3 |

# Appendix C - Test Case to Test Suite Mapping

The tables below provide a mapping as to which test cases are supported in each of the 6 test suites:

**Positive Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case** | **Suites and Extensions Supported** | | | | | |
| **Suite 1**  **SPX** | **Suite 2**  **XPM** | **Suite 3**  **XXX** | **Suite 4**  **SXX** | **Suite 5**  **XPX** | **Suite 6**  **XXM** |
| OPC |  |  |  |  |  |  |
| 0101\_01 | • | • | • | • | • | • |
| 0101\_02 | • | • | • | • | • | • |
| 0101\_03 | • | • | • | • | • | • |
| 0102\_01 | • | • | • | • | • | • |
| 0102\_02 | • | • | • | • | • | • |
| 0102\_03 | • | • | • | • | • | • |
| 0103\_01 | • | • | • | • | • | • |
| 0104\_01 | • | • | • | • | • | • |
| 0104\_02 | • | • | • | • | • | • |
| 0104\_03 | • | • | • | • | • | • |
| 0106\_01 | • | • | • | • | • | • |
| 0106\_02 | • | • | • | • | • | • |
| 0107\_01 | • | • |  | • | • |  |
| 0107\_02 | • | • |  | • | • |  |
| Core |  |  |  |  |  |  |
| 0302\_01 | • | • | • | • | • | • |
| 0302\_02 | • | • | • | • | • | • |
| 0302\_03 | • | • | • | • | • | • |
| 0304\_01 | • | • | • | • | • | • |
| 0304\_02 | • | • | • | • | • | • |
| 0304\_03 | • | • | • | • | • | • |
| 0304\_04 | • | • | • | • | • | • |
| 0306\_01 | • | • | • | • | • | • |
| 0306\_02 | • | • | • | • | • | • |
| 0306\_03 | • | • | • | • | • | • |
| 0306\_04 | • | • | • | • | • | • |
| 0306\_05 | • | • | • | • | • | • |
| 0306\_06 | • | • | • | • | • | • |
| 0306\_07 | • | • | • | • | • | • |
| 0307\_01 | • | • | • | • | • | • |
| 0308\_01 | • | • | • | • | • | • |
| 0309\_01 | • | • | • | • | • | • |
| 0310\_01 | • | • | • | • | • | • |
| 0311\_01 | • | • | • | • | • | • |
| 0312\_01 | • | • | • | • | • | • |
| 0313\_01 | • | • | • | • | • | • |
| 0314\_01 | • | • | • | • | • | • |
| 0314\_02 | • | • | • | • | • | • |
| 0314\_03 |  | • | • |  | • | • |
| 0314\_04 |  | • | • |  | • | • |
| 0314\_05 |  | • | • |  | • | • |
| 0315\_01 | • | • | • | • | • | • |
| 0316\_01 | • | • | • | • | • | • |
| 0317\_01 | • | • | • | • | • | • |
| 0318\_01 | • | • | • | • | • | • |
| 0319\_01 | • | • | • | • | • | • |
| 0321\_01 | • | • | • | • | • | • |
| 0322\_01 | • | • | • | • | • | • |
| 0323\_01 | • | • | • | • | • | • |
| 0323\_02 | • | • | • | • | • | • |
| 0324\_01 | • | • |  | • | • |  |
| 0325\_01 | • | • | • | • | • | • |
| 0326\_01 | • | • | • | • | • | • |
| 0326\_02 | • | • | • | • | • | • |
| 0326\_03 |  | • | • |  | • | • |
| 0327\_01 | • | • | • | • | • | • |
| 0328\_01 | • | • | • | • | • | • |
| 0329\_01 | • | • | • | • | • | • |
| 0330\_01 | • | • | • | • | • | • |
| 0331\_01 | • | • | • | • | • | • |
| 0333\_01 | • | • | • | • | • | • |
| 0333\_02 | • | • | • | • | • | • |
| 0333\_03 |  | • | • |  | • | • |
| 0334\_01 | • | • | • | • | • | • |
| 0334\_02 | • | • | • | • | • | • |
| 0335\_01 | • | • |  | • | • |  |
| 0335\_02 | • | • | • | • | • | • |
| 0335\_03 | • | • | • | • | • | • |
| 0335\_04 | • | • | • | • | • | • |
| 0336\_01 | • | • | • | • | • | • |
| 0336\_02 | • | • | • | • | • | • |
| 0337\_01 | • | • | • | • | • | • |
| 0337\_02 | • | • | • | • | • | • |
| 0337\_03 | • | • | • | • | • | • |
| 0337\_04 | • | • | • | • | • | • |
| 0337\_05 | • | • | • | • | • | • |
| 0337\_06 | • | • |  |  | • |  |
| Material |  |  |  |  |  |  |
| 0501\_01 |  | • |  |  |  | • |
| 0501\_02 |  | • |  |  |  | • |
| 0501\_03 |  | • |  |  |  | • |
| 0501\_04 |  | • |  |  |  | • |
| 0501\_07 |  | • |  |  |  | • |
| 0501\_08 |  | • |  |  |  | • |
| 0501\_09 |  | • |  |  |  | • |
| 0502\_01 |  | • |  |  |  | • |
| 0502\_02 |  | • |  |  |  | • |
| 0502\_03 |  | • |  |  |  | • |
| 0502\_04 |  | • |  |  |  | • |
| 0502\_05 |  | • |  |  |  | • |
| 0503\_01 |  | • |  |  |  | • |
| 0503\_02 |  | • |  |  |  | • |
| 0503\_03 |  | • |  |  |  | • |
| 0503\_04 |  | • |  |  |  | • |
| 0503\_05 |  | • |  |  |  | • |
| 0503\_06 |  | • |  |  |  | • |
| 0503\_07 |  | • |  |  |  | • |
| 0503\_08 |  | • |  |  |  | • |
| 0504\_01 |  | • |  |  |  | • |
| 0504\_02 |  | • |  |  |  | • |
| 0504\_03 |  | • |  |  |  | • |
| 0504\_04 |  | • |  |  |  | • |
| 0504\_05 |  | • |  |  |  | • |
| 0504\_06 |  | • |  |  |  | • |
| 0504\_07 |  | • |  |  |  | • |
| 0505\_01 |  | • |  |  |  | • |
| 0505\_02 |  | • |  |  |  | • |
| 0505\_03 |  | • |  |  |  | • |
| 0506\_01 |  | • |  |  |  | • |
| 0506\_02 |  | • |  |  |  | • |
| 0506\_03 |  | • |  |  |  | • |
| 0506\_04 |  | • |  |  |  | • |
| 0507\_01 |  | • |  |  |  | • |
| 0507\_02 |  | • |  |  |  | • |
| 0507\_03 |  | • |  |  |  | • |
| 0507\_04 |  | • |  |  |  | • |
| 0507\_05 |  | • |  |  |  | • |
| 0507\_06 |  | • |  |  |  | • |
| 0507\_07 |  | • |  |  |  | • |
| 0507\_08 |  | • |  |  |  | • |
| 0507\_09 |  | • |  |  |  | • |
| 0508\_01 |  | • |  |  |  | • |
| 0508\_02 |  | • |  |  |  | • |
| 0508\_03 |  | • |  |  |  | • |
| 0508\_04 |  | • |  |  |  | • |
| 0508\_05 |  | • |  |  |  | • |
| 0508\_06 |  | • |  |  |  | • |
| 0508\_07 |  | • |  |  |  | • |
| 0508\_08 |  | • |  |  |  | • |
| 0508\_09 |  | • |  |  |  | • |
| 0508\_10 |  | • |  |  |  | • |
| 0508\_11 |  | • |  |  |  | • |
| 0508\_12 |  | • |  |  |  | • |
| 0508\_13 |  | • |  |  |  | • |
| 0508\_14 |  | • |  |  |  | • |
| 0509\_01 |  | • |  |  |  | • |
| 0510\_01 |  | • |  |  |  | • |
| 0510\_02 |  | • |  |  |  | • |
| 0510\_03 |  | • |  |  |  | • |
| 0510\_04 |  | • |  |  |  | • |
| 0511\_01 |  | • |  |  |  | • |
| 0511\_02 |  | • |  |  |  | • |
| 0511\_03 |  | • |  |  |  | • |
| 0511\_04 |  | • |  |  |  | • |
| 0512\_01 |  | • |  |  |  | • |
| 0512\_02 |  | • |  |  |  | • |
| 0512\_03 |  | • |  |  |  | • |
| 0512\_04 |  | • |  |  |  | • |
| 0512\_06 |  | • |  |  |  | • |
| 0513\_01 |  | • |  |  |  | • |
| 0513\_02 |  | • |  |  |  | • |
| 0513\_03 |  | • |  |  |  | • |
| 0513\_04 |  | • |  |  |  | • |
| 0513\_05 |  | • |  |  |  | • |
| 0514\_01 |  | • |  |  |  | • |
| 0514\_02 |  | • |  |  |  | • |
| 0514\_03 |  | • |  |  |  | • |
| 0514\_04 |  | • |  |  |  | • |
| 0514\_05 |  | • |  |  |  | • |
| 0514\_06 |  | • |  |  |  | • |
| 0514\_07 |  | • |  |  |  | • |
| 0514\_08 |  | • |  |  |  | • |
| 0514\_09 |  | • |  |  |  | • |
| 0514\_10 |  | • |  |  |  | • |
| 0514\_11 |  | • |  |  |  | • |
| 0514\_12 |  | • |  |  |  | • |
| 0515\_01 |  | • |  |  |  | • |
| 0515\_02 |  | • |  |  |  | • |
| 0515\_03 |  | • |  |  |  | • |
| 0516\_01 |  | • |  |  |  | • |
| 0516\_02 |  | • |  |  |  | • |
| 0516\_03 |  | • |  |  |  | • |
| 0516\_04 |  | • |  |  |  | • |
| 0516\_05 |  | • |  |  |  | • |
| 0516\_06 |  | • |  |  |  | • |
| 0516\_07 |  | • |  |  |  | • |
| 0516\_08 |  | • |  |  |  | • |
| 0517\_01 |  | • |  |  |  | • |
| 0517\_02 |  | • |  |  |  | • |
| 0517\_04 |  | • |  |  |  | • |
| 0517\_05 |  | • |  |  |  | • |
| 0517\_08 |  | • |  |  |  | • |
| 0517\_09 |  | • |  |  |  | • |
| 0517\_10 |  | • |  |  |  | • |
| 0517\_11 |  | • |  |  |  | • |
| 0517\_12 |  | • |  |  |  | • |
| 0517\_13 |  | • |  |  |  | • |
| 0517\_14 |  | • |  |  |  | • |
| 0517\_15 |  | • |  |  |  | • |
| 0517\_16 |  | • |  |  |  | • |
| 0517\_20 |  | • |  |  |  | • |
| 0517\_21 |  | • |  |  |  | • |
| 0517\_22 |  | • |  |  |  | • |
| 0517\_23 |  | • |  |  |  | • |
| 0517\_24 |  | • |  |  |  | • |
| 0517\_25 |  | • |  |  |  | • |
| 0517\_26 |  | • |  |  |  | • |
| 0517\_27 |  | • |  |  |  | • |
| 0517\_28 |  | • |  |  |  | • |
| 0517\_29 |  | • |  |  |  | • |
| 0517\_30 |  | • |  |  |  | • |
| 0517\_31 |  | • |  |  |  | • |
| 0517\_32 |  | • |  |  |  | • |
| 0517\_33 |  | • |  |  |  | • |
| 0517\_34 |  | • |  |  |  | • |
| 0517\_35 |  | • |  |  |  | • |
| 0518\_01 |  | • |  |  |  | • |
| 0518\_02 |  | • |  |  |  | • |
| 0518\_03 |  | • |  |  |  | • |
| 0518\_04 |  | • |  |  |  | • |
| 0518\_05 |  | • |  |  |  | • |
| 0518\_06 |  | • |  |  |  | • |
| 0518\_07 |  | • |  |  |  | • |
| 0518\_08 |  | • |  |  |  | • |
| 0518\_09 |  | • |  |  |  | • |
| 0518\_10 |  | • |  |  |  | • |
| 0518\_11 |  | • |  |  |  | • |
| 0518\_12 |  | • |  |  |  | • |
| 0518\_13 |  | • |  |  |  | • |
| 0518\_14 |  | • |  |  |  | • |
| 0518\_15 |  | • |  |  |  | • |
| 0519\_01 |  | • |  |  |  | • |
| 0520\_01 |  | • |  |  |  | • |
| 0520\_02 |  | • |  |  |  | • |
| 0520\_03 |  | • |  |  |  | • |
| 0521\_01 |  | • |  |  |  |  |
| 0522\_01 |  | • |  |  |  | • |
| 0522\_02 |  | • |  |  |  | • |
| 0523\_01 |  | • |  |  |  |  |
| 0523\_02 |  | • |  |  |  |  |
| 0523\_03 |  | • |  |  |  |  |
| 0523\_04 |  | • |  |  |  |  |
| 0524\_01 |  | • |  |  |  | • |
| 0524\_02 |  | • |  |  |  | • |
| 0525\_01 |  | • |  |  |  | • |
| 0525\_02 |  | • |  |  |  | • |
| 0525\_03 |  | • |  |  |  | • |
| 0525\_04 |  | • |  |  |  | • |
| 0525\_06 |  | • |  |  |  | • |
| 0525\_07 |  | • |  |  |  | • |
| 0525\_08 |  | • |  |  |  | • |
| 0525\_09 |  | • |  |  |  | • |
| 0525\_10 |  | • |  |  |  | • |
| 0525\_11 |  | • |  |  |  | • |
| 0525\_12 |  | • |  |  |  | • |
| 0525\_13 |  | • |  |  |  | • |
| 0525\_14 |  | • |  |  |  | • |
| 0525\_15 |  | • |  |  |  | • |
| 0525\_16 |  | • |  |  |  | • |
| 0525\_17 |  | • |  |  |  | • |
| 0525\_18 |  | • |  |  |  | • |
| 0525\_19 |  | • |  |  |  | • |
| 0525\_20 |  | • |  |  |  | • |
| 0525\_21 |  | • |  |  |  | • |
| 0525\_22 |  | • |  |  |  | • |
| 0525\_23 |  | • |  |  |  | • |
| 0525\_24 |  | • |  |  |  | • |
| 0525\_25 |  | • |  |  |  | • |
| 0525\_26 |  | • |  |  |  | • |
| 0525\_27 |  | • |  |  |  | • |
| 0525\_28 |  | • |  |  |  | • |
| 0525\_29 |  | • |  |  |  | • |
| 0525\_30 |  | • |  |  |  | • |
| 0525\_31 |  | • |  |  |  | • |
| 0525\_32 |  | • |  |  |  | • |
| 0525\_33 |  | • |  |  |  | • |
| 0525\_34 |  | • |  |  |  | • |
| 0525\_35 |  | • |  |  |  | • |
| 0525\_36 |  | • |  |  |  | • |
| 0525\_37 |  | • |  |  |  | • |
| 0525\_38 |  | • |  |  |  | • |
| 0525\_39 |  | • |  |  |  | • |
| 0525\_40 |  | • |  |  |  | • |
| 0525\_41 |  | • |  |  |  | • |
| 0525\_42 |  | • |  |  |  | • |
| 0525\_43 |  | • |  |  |  | • |
| 0525\_44 |  | • |  |  |  | • |
| 0525\_45 |  | • |  |  |  | • |
| 0525\_46 |  | • |  |  |  | • |
| 0525\_47 |  | • |  |  |  | • |
| 0525\_48 |  | • |  |  |  | • |
| 0525\_49 |  | • |  |  |  | • |
| 0525\_50 |  | • |  |  |  | • |
| 0525\_51 |  | • |  |  |  | • |
| 0525\_52 |  | • |  |  |  | • |
| 0525\_53 |  | • |  |  |  | • |
| 0525\_54 |  | • |  |  |  | • |
| 0525\_55 |  | • |  |  |  | • |
| 0526\_01 |  | • |  |  |  | • |
| 0526\_02 |  | • |  |  |  | • |
| 0526\_03 |  | • |  |  |  | • |
| 0526\_04 |  | • |  |  |  | • |
| 0526\_05 |  | • |  |  |  | • |
| 0526\_06 |  | • |  |  |  | • |
| 0526\_07 |  | • |  |  |  | • |
| 0526\_08 |  | • |  |  |  | • |
| 0526\_09 |  | • |  |  |  | • |
| 0529\_01 |  | • |  |  |  | • |
| 0529\_02 |  | • |  |  |  | • |
| 0529\_03 |  | • |  |  |  | • |
| 0529\_04 |  | • |  |  |  | • |
| 0529\_05 |  | • |  |  |  | • |
| 0530\_01 |  | • |  |  |  | • |
| 0530\_02 |  | • |  |  |  | • |
| 0530\_03 |  | • |  |  |  | • |
| 0530\_04 |  | • |  |  |  | • |
| 0530\_05 |  | • |  |  |  | • |
| 0530\_06 |  | • |  |  |  | • |
| 0530\_07 |  | • |  |  |  | • |
| 0530\_08 |  | • |  |  |  | • |
| Production |  |  |  |  |  |  |
| 0701\_01 | • |  |  |  |  |  |
| 0701\_02 | • |  |  |  |  |  |
| 0701\_04 | • |  |  |  |  |  |
| 0701\_05 | • |  |  |  |  |  |
| 0701\_06 | • |  |  |  |  |  |
| 0701\_07 | • |  |  |  |  |  |
| 0701\_08 | • |  |  |  |  |  |
| 0701\_09 | • |  |  |  |  |  |
| 0701\_11 | • |  |  |  |  |  |
| 0701\_12 | • |  |  |  |  |  |
| 0701\_13 | • |  |  |  |  |  |
| 0701\_14 | • |  |  |  |  |  |
| 0702\_01 |  | • |  |  | • |  |
| 0702\_02 |  | • |  |  | • |  |
| 0702\_03 |  | • |  |  | • |  |
| 0702\_04 |  | • |  |  | • |  |
| 0702\_05 |  | • |  |  | • |  |
| 0702\_06 |  | • |  |  | • |  |
| 0703\_01 | • | • |  |  | • |  |
| 0703\_02 | • | • |  |  | • |  |
| 0703\_03 | • | • |  |  | • |  |
| 0703\_04 | • | • |  |  | • |  |
| 0703\_05 | • | • |  |  | • |  |
| 0703\_06 | • | • |  |  | • |  |
| 0703\_07 | • | • |  |  | • |  |
| 0703\_08 | • | • |  |  | • |  |
| 0703\_09 | • | • |  |  | • |  |
| 0703\_10 | • | • |  |  | • |  |
| 0703\_11 | • | • |  |  | • |  |
| 0703\_12 | • | • |  |  | • |  |
| 0704\_01 | • |  |  |  |  |  |
| 0704\_02 | • |  |  |  |  |  |
| 0704\_03 | • |  |  |  |  |  |
| 0704\_04 | • |  |  |  |  |  |
| Misc |  |  |  |  |  |  |
| 0901\_01 | • | • | • | • | • | • |
| 0901\_02 | • | • | • | • | • | • |
| 0901\_03 | • | • | • | • | • | • |
| 0901\_04 | • | • | • | • | • | • |
| 0901\_05 | • | • | • | • | • | • |
| 0901\_06 | • | • | • | • | • | • |
| 0901\_07 | • | • | • | • | • | • |
| 0901\_08 | • | • | • | • | • | • |
| 0901\_09 | • | • | • | • | • | • |
| 0901\_10 | • | • | • | • | • | • |
| 0901\_11 | • | • | • | • | • | • |
| 0901\_12 | • | • | • | • | • | • |
| 0902\_01 | • | • | • | • | • | • |
| 0902\_02 | • | • | • | • | • | • |
| 0902\_03 | • | • | • | • | • | • |
| 0902\_04 | • | • | • | • | • | • |
| 0902\_05 | • | • | • | • | • | • |
| 0902\_06 | • | • | • | • | • | • |
| 0903\_01 | • | • | • | • | • | • |
| 0903\_02 | • | • | • | • | • | • |
| 0903\_03 | • | • | • | • | • | • |
| 0903\_04 | • | • | • | • | • | • |
| 0903\_06 | • | • | • | • | • | • |
| 0904\_01 | • | • | • | • | • | • |
| 0904\_02 | • | • | • | • | • | • |
| 0904\_03 | • | • | • | • | • | • |
| 0904\_04 | • | • | • | • | • | • |
| 0905\_01 | • | • | • | • | • | • |
| 0905\_02 | • | • | • | • | • | • |
| 0905\_03 | • | • | • | • | • | • |
| 0905\_04 | • | • | • | • | • | • |
| 0905\_05 | • | • | • | • | • | • |
| 0905\_06 | • | • | • | • | • | • |
| 0905\_07 | • | • | • | • | • | • |
| 0905\_08 | • | • | • | • | • | • |
| 0905\_09 | • | • | • | • | • | • |
| 0905\_10 | • | • | • | • | • | • |
| 0905\_12 | • | • | • | • | • | • |
| 0905\_13 | • | • | • | • | • | • |
| 0906\_01 | • | • | • | • | • | • |
| 0906\_02 | • | • | • | • | • | • |
| 0906\_03 | • | • | • | • | • | • |
| 0906\_04 | • | • | • | • | • | • |
| 0906\_05 | • | • | • | • | • | • |
| 0906\_06 | • | • | • | • | • | • |
| 0907\_01 | • | • | • | • | • | • |
| 0907\_02 | • | • | • | • | • | • |
| 0907\_03 | • | • | • | • | • | • |
| 0907\_04 | • | • | • | • | • | • |
| 0909\_01 | • | • | • | • | • | • |
| 0909\_02 | • |  |  | • |  |  |
| 0909\_03 | • | • | • | • | • | • |
| 0909\_04 | • | • | • | • | • | • |
| 0909\_05 |  | • | • |  | • | • |
| 0909\_06 | • |  |  | • |  |  |
| 0909\_07 |  | • |  |  | • |  |
| 0909\_08 |  | • |  |  | • |  |
| 0910\_01 | • | • | • | • | • | • |
| 0910\_02 | • | • | • | • | • | • |
| 0910\_03 | • | • | • | • | • | • |
| 0910\_04 | • | • | • | • | • | • |
| 0910\_05 | • | • | • | • | • | • |
| 0910\_06 | • | • |  |  | • |  |
| 0911\_01 | • | • | • | • | • | • |
| 0912\_01 | • | • | • | • | • | • |
| 0913\_01 | • | • | • | • | • | • |
| 0914\_01 | • | • |  | • | • | • |
| 0915\_01 | • | • |  |  | • |  |
| Slice |  |  |  |  |  |  |
| 1501\_02 | • |  |  | • |  |  |
| 1501\_03 | • |  |  | • |  |  |
| 1502\_01 | • |  |  | • |  |  |
| 1502\_02 | • |  |  | • |  |  |
| 1502\_03 | • |  |  | • |  |  |
| 1502\_04 | • |  |  | • |  |  |
| 1502\_05 | • |  |  | • |  |  |
| 1503\_01 | • |  |  | • |  |  |
| 1503\_02 | • |  |  | • |  |  |
| 1503\_03 | • |  |  | • |  |  |
| 1504\_01 | • |  |  | • |  |  |
| 1505\_01 | • |  |  | • |  |  |
| 1505\_02 | • |  |  | • |  |  |
| 1505\_03 | • |  |  | • |  |  |
| 1506\_01 | • |  |  | • |  |  |
| 1507\_01 | • |  |  | • |  |  |
| 1508\_01 | • |  |  | • |  |  |
| 1508\_02 | • |  |  | • |  |  |
| 1509\_01 | • |  |  | • |  |  |
| 1509\_02 | • |  |  | • |  |  |
| 1509\_03 | • |  |  | • |  |  |
| 1509\_04 | • |  |  | • |  |  |
| 1510\_01 | • |  |  | • |  |  |
| 1511\_01 | • |  |  | • |  |  |
| 1511\_02 | • |  |  | • |  |  |
| 1511\_03 | • |  |  | • |  |  |
| 1511\_04 | • |  |  | • |  |  |
| 1511\_05 | • |  |  | • |  |  |
| 1511\_06 | • |  |  | • |  |  |
| 1512\_01 | • |  |  | • |  |  |
| 1513\_01 | • |  |  | • |  |  |
| 1513\_02 | • |  |  | • |  |  |
| 1513\_03 | • |  |  | • |  |  |
| 1514\_01 | • |  |  | • |  |  |
| 1514\_02 | • |  |  |  |  |  |
| 1515\_01 | • |  |  | • |  |  |
| 1516\_01 | • |  |  |  |  |  |
| 1516\_02 | • |  |  |  |  |  |
| 1517\_01 | • |  |  | • |  |  |
| 1517\_02 | • |  |  | • |  |  |

**Negative Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case** | **Suites and Extensions Supported** | | | | | |
| **Suite 1**  **SPX** | **Suite 2**  **XPM** | **Suite 3**  **XXX** | **Suite 4**  **SXX** | **Suite 5**  **XPX** | **Suite 6**  **XXM** |
| OPC |  |  |  |  |  |  |
| 0202\_01 | • | • | • | • | • | • |
| 0203\_01 | • | • | • | • | • | • |
| 0204\_01 | • | • | • | • | • | • |
| 0205\_01 | • | • | • | • | • | • |
| 0205\_02 | • | • | • | • | • | • |
| 0206\_01 | • | • | • | • | • | • |
| 0207\_01 | • | • | • | • | • | • |
| 0208\_01 | • | • | • | • | • | • |
| Core |  |  |  |  |  |  |
| 0402\_01 | • | • | • | • | • | • |
| 0402\_02 | • | • | • | • | • | • |
| 0402\_03 | • | • | • | • | • | • |
| 0402\_04 | • | • | • | • | • | • |
| 0403\_01 | • | • | • | • | • | • |
| 0404\_01 | • | • | • | • | • | • |
| 0404\_02 | • | • | • | • | • | • |
| 0404\_03 | • | • | • | • | • | • |
| 0404\_04 | • | • | • | • | • | • |
| 0405\_01 | • | • | • | • | • | • |
| 0405\_02 | • | • | • | • | • | • |
| 0405\_03 | • | • |  |  | • |  |
| 0405\_05 | • | • | • | • | • | • |
| 0406\_01 | • | • | • | • | • | • |
| 0406\_02 | • | • |  |  | • |  |
| 0407\_01 | • | • |  |  | • |  |
| 0407\_02 | • | • | • | • | • | • |
| 0409\_01 | • | • | • | • | • | • |
| 0410\_01 | • | • | • | • | • | • |
| 0410\_02 | • |  |  | • |  |  |
| 0410\_03 | • | • | • | • | • | • |
| 0410\_04 | • |  |  | • |  |  |
| 0411\_01 | • | • | • | • | • | • |
| 0412\_01 | • | • | • | • | • | • |
| 0412\_02 | • |  |  | • |  |  |
| 0412\_03 | • |  |  | • |  |  |
| 0412\_04 | • |  |  | • |  |  |
| 0413\_01 | • | • |  |  | • |  |
| 0413\_02 | • | • | • | • | • | • |
| 0415\_01 | • | • |  |  | • |  |
| 0415\_02 | • | • |  |  | • |  |
| 0415\_03 | • | • |  |  | • |  |
| 0415\_04 |  | • |  |  | • |  |
| 0416\_01 | • | • | • | • | • | • |
| 0416\_02 | • | • | • | • | • | • |
| 0417\_01 | • |  |  | • |  |  |
| 0418\_01 | • | • | • | • | • | • |
| 0419\_01 | • | • | • | • | • | • |
| 0420\_01 | • | • | • | • | • | • |
| 0421\_01 | • | • | • | • | • | • |
| 0421\_02 | • | • | • | • | • | • |
| 0422\_01 | • | • | • | • | • | • |
| 0424\_01 | • | • | (1) | • | (1) | • |
| 0426\_01 | • | • | • | • | • | • |
| 0427\_01 | • | • | • | • | • | • |
| Material |  |  |  |  |  |  |
| 0601\_01 |  | • |  |  |  | • |
| 0601\_02 |  | • |  |  |  |  |
| 0602\_01 |  | • |  |  |  | • |
| 0602\_02 |  | • |  |  |  | • |
| 0602\_03 |  | • |  |  |  | • |
| 0602\_04 |  | • |  |  |  | • |
| 0603\_01 |  | • |  |  |  | • |
| 0604\_01 |  | • |  |  |  | • |
| 0604\_02 |  | • |  |  |  | • |
| 0604\_03 |  | • |  |  |  | • |
| 0605\_01 |  | • |  |  |  | • |
| 0605\_02 |  | • |  |  |  | • |
| 0606\_01 |  | • |  |  |  | • |
| 0606\_02 |  | • |  |  |  | • |
| 0606\_03 |  | • |  |  |  | • |
| 0607\_01 |  | • |  |  |  | • |
| 0608\_01 |  | • |  |  |  | • |
| 0609\_01 |  | • |  |  |  | • |
| 0609\_02 |  | • |  |  |  | • |
| 0609\_03 |  | • |  |  |  | • |
| 0609\_04 |  | • |  |  |  | • |
| 0609\_05 |  | • |  |  |  | • |
| 0609\_06 |  | • |  |  |  | • |
| 0609\_07 |  | • |  |  |  | • |
| 0609\_08 |  | • |  |  |  | • |
| 0609\_09 |  | • |  |  |  | • |
| 0609\_10 |  | • |  |  |  | • |
| 0609\_11 |  | • |  |  |  | • |
| 0610\_01 |  | • |  |  |  | • |
| 0610\_02 |  | • |  |  |  | • |
| 0610\_03 |  | • |  |  |  | • |
| Production |  |  |  |  |  |  |
| 0801\_01 | • | • |  |  | • |  |
| 0801\_02 | • | • |  |  | • |  |
| 0801\_03 | • | • |  |  | • |  |
| 0801\_04 | • | • |  |  | • |  |
| 0801\_05 | • | • |  |  | • |  |
| 0801\_06 | • | • |  |  | • |  |
| 0801\_07 | • |  |  |  |  |  |
| 0801\_08 | • |  |  |  |  |  |
| 0801\_09 | • |  |  |  |  |  |
| 0802\_01 | • | • |  |  | • |  |
| 0802\_02 | • | • |  |  | • |  |
| 0802\_03 | • | • |  |  | • |  |
| 0802\_04 | • | • |  |  | • |  |
| 0802\_05 | • | • |  |  | • |  |
| 0803\_01 | • | • |  |  | • |  |
| Slice |  |  |  |  |  |  |
| 1601\_01 | • |  |  | • |  |  |
| 1601\_02 | • |  |  | • |  |  |
| 1601\_03 | • |  |  | • |  |  |
| 1601\_04 | • |  |  | • |  |  |
| 1601\_05 | • |  |  | • |  |  |
| 1604\_01 | • |  |  | • |  |  |
| 1605\_01 | • |  |  | • |  |  |
| 1606\_01 | • |  |  | • |  |  |
| 1607\_01 | • |  |  | • |  |  |
| 1608\_01 | • |  |  | • |  |  |
| 1609\_01 | • |  |  | • |  |  |
| 1609\_02 | • |  |  | • |  |  |
| 1610\_01 | • |  |  | • |  |  |
| 1612\_01 | • |  |  | • |  |  |