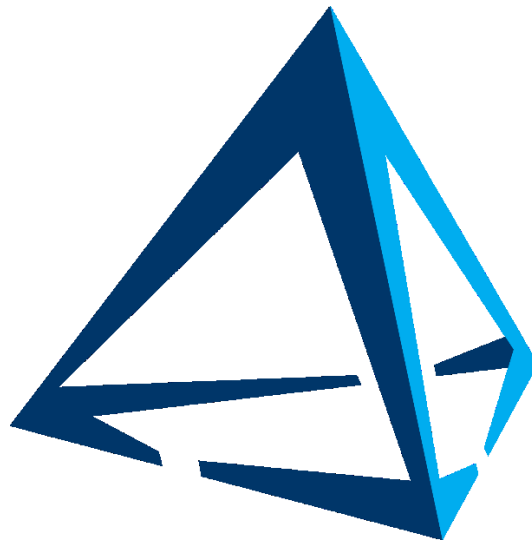


# Lib3MF

---

## Open Source Toolkit for the 3D Manufacturing Format



<b>Version</b>	1.0.1
<b>Status</b>	Published

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.

## Contents

<b>About this Library.....</b>	<b>2</b>
<b>General Architecture .....</b>	<b>3</b>
<b>Examples .....</b>	<b>3</b>
<b>COM Interfaces .....</b>	<b>4</b>
<b>Plain C Interfaces.....</b>	<b>5</b>
<b>Class Reference – Core Specification .....</b>	<b>5</b>
<b>License.....</b>	<b>13</b>
<b>Contributors.....</b>	<b>13</b>

---

## About this Library

Lib3MF is a C++ implementation of the 3D Manufacturing Format file standard.

As 3MF shall become an universal 3D Printing standard, its quick adoption is very important. This library shall lower all barriers of adoption to any possible user, let it be software providers, hardware providers, service providers or middleware tools.

Its aim is to offer an open source way to integrate 3MF reading and writing capabilities, as well as conversion and validation tools for input and output data. The 3MF Library shall provide a clean and easy-to-use API to speed up the development and keep integration costs at a minimum.

While the current code is primarily made for a Microsoft Visual Studio Environment, a lot of energy has been put into keeping it as platform independent as far as possible. For example, it compiles well with the GCC compiler, but there is some work left to recode a few platform specific functionalities, which are now covered by the WinRT platform (like XML parsing and ZIP compression). As described below, we are looking for contributors with extensive experience in this field.

To understand this documentation in full extent, it is important to have taken a look at the 3MF specification 1.0, available for free download at <http://3mf.io/what-is-3mf/3mf-specification/>.

For the source code of the library code, please visit <https://github.com/3MFConsortium/lib3mf>.

Example code can be found at <https://github.com/3MFConsortium/lib3mf-examples>.

For any hints, feedback or contributions, please contact [lib3mf@netfabb.com](mailto:lib3mf@netfabb.com).

## General Architecture

The 3MF library API compiles into three different flavours, which are basically just bindings to the same class model:

- (1) A COM Interface to allow an object-oriented approach in many languages on the Windows platform (e.g. C# or Visual Basic)
- (2) A COM-like DLL Interface, which exposes the same API as the COM Interface, but comes without the need for global CLSID registration. This allows to link the library natively to many unmanaged object-oriented languages (e.g. C++ or Delphi)
- (3) A plain C Interface, which wraps the class structure of the API into pseudo-object-oriented. This works very well for C programs and other low level languages, as well as other operating systems.

In the first release, only (2) is fully implemented, while the others will require some work and a proof of concept implementation. More information can be found in the corresponding headers in

- (1) Include/Model/COM/NMR\_COMInterfaces.h
- (2) Include/Model/COM/NMR\_COMFactory.h
- (3) Include/Model/COM/NMR\_DLLInterfaces.h

---

## Examples

Currently, there are 4 examples which show how to use the library:

- *Cube*: a simple example how to create an empty 3MF document and add custom geometry to it.
- *Converter*: a simple program to convert 3MFs into (binary) STLs and back.
- *Components*: explains component handling in 3MF
- *ExtractInfo*: shows how to import a 3MF and navigate through the in memory representation of the model.

Please note, that you might need a proper understanding of the 3MF Specification in order to get the most out of the example code.

## COM Interfaces

For the 3MF Core spec, the following interfaces specify an in memory representation of the 3MF Document. For a detailed description, please refer to Include/Model/COM/NMR\_COMInterfaces.h.

Interface	derived from	Description
<a href="#">ILib3MFBase</a>	<a href="#">IUnknown</a>	<a href="#">ILib3MFBase</a> is a base interface, which serves as parent for all interfaces related to the 3MF Library
<a href="#">ILib3MFModelWriter</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelWriter</a> encapsulates an writer class for writing the model into a specific file type.
<a href="#">ILib3MFModelReader</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelReader</a> encapsulates an reader class for reading a model from a specific file type.
<a href="#">ILib3MFModelResource</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelResource</a> is a base interface for all 3MF Resources.
<a href="#">ILib3MFModelResourceIterator</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelResourceIterator</a> is a helper class to iterate through arbitrary lists of 3MF resources
<a href="#">ILib3MFModelObjectResource</a>	<a href="#">ILib3MFModelResource</a>	<a href="#">ILib3MFModelObjectResource</a> is a base interface for all 3MF Object Resources.
<a href="#">ILib3MFModelMeshObject</a>	<a href="#">ILib3MFModelObjectResource</a>	<a href="#">ILib3MFModelMeshObject</a> encapsulates all methods for handling 3MF mesh objects.
<a href="#">ILib3MFModelComponentsObject</a>	<a href="#">ILib3MFModelObjectResource</a>	<a href="#">ILib3MFModelComponentsObject</a> encapsulates all methods for handling 3MF component objects.
<a href="#">ILib3MFModelBaseMaterial</a>	<a href="#">ILib3MFModelResource</a>	<a href="#">ILib3MFModelBaseMaterial</a> describes a basic material, as defined in the 3MF core specification.
<a href="#">ILib3MFModelComponent</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelComponent</a> encapsulates one component node of a 3MF component object.
<a href="#">ILib3MFModelBuildItem</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelBuildItem</a> encapsulates all methods for handling 3MF build items.
<a href="#">ILib3MFModelBuildItemIterator</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelBuildItemIterator</a> is a helper class to iterate through arbitrary lists of 3MF build items.
<a href="#">ILib3MFModelThumbnail</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelThumbnail</a> allows to access package and object thumbnails.
<a href="#">ILib3MFModel</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModel</a> is the basic instance owning all In-Memory elements of a 3MF file.

<a href="#">ILib3MFModelFactory</a>	<a href="#">ILib3MFBase</a>	<a href="#">ILib3MFModelFactory</a> is a factory interface for <a href="#">ILib3MFModel</a>
-------------------------------------	-----------------------------	---

---

## Plain C Interfaces

In order to address a wider user base, the object-oriented interfaces above are also compiled into a DLL for emulating pseudo objects in a procedural language. We do not encourage to use this interface, but there are very valid use cases for it.

Please note, that this wrapper is a lot less type-safe than the other possibilities.

**Example:** The following code in COM

```
pModel->QueryWriter (L„3mf“, &pModelWriter);
```

will be translated into

```
lib3mf_model_querywriter (pModel, L„3mf“, &pModelWriter);
```

All these translations follow the same pattern. For more information, please compare the corresponding header files

- (1) Include/Model/COM/NMR\_COMInterfaces.h
- (2) Include/Model/COM/NMR\_DLLInterfaces.h

---

## Class Reference – Core Specification

The following list shall give a short class overview of the library. The interfaces are all defined centrally in one header file. Please read Include/Model/COM/NMR\_COMInterfaces.h for all details.

All Methods return a HRESULT defining the success of the operation. A successful operation always returns 0 (S\_OK), and a windows error code otherwise. If no success is returned, the output parameters might be in an undefined state.

### 1. ILib3MFModelWriter

ILib3MFModelWriter encapsulates a writer class for a writing the model into a specific file type. Current implementations include (binary) STL and 3MF.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
WriteToFile	LPCWSTR <b>pwszFilename</b> : Filename to write into	Writes out the model as file. The file type is specified by the Model Writer class
WriteToStream	IStream* <b>pStream</b> : IStream to write into	Writes out the model into a COM IStream. Only available on Windows.

## 2. ILib3MFModelReader

ILib3MFModelReader encapsulates a reader class for reading a model from a specific file type. Current implementations include (binary) STL and 3MF.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
ReadFromFile	LPCWSTR <b>pwszFilename</b> : Filename to read from	Reads a model from a file. The file type is specified by the Model Reader class.
ReadFromStream	IStream* <b>pStream</b> : IStream to read from	Reads a model from a COM IStream. Only available on Windows.
GetWarningCount	DWORD* <b>pnWarningCount</b> : filled with the count of the occurred warnings.	Returns warning and error count of the read process.
GetWarning	DWORD <b>nIndex</b> : Index of the Warning. Valid values are 0 to WarningCount – 1. DWORD* <b>pErrorCode</b> : filled with the error code of the warning LPWSTR <b>pwszBuffer</b> : filled with the error message, may be NULL DWORD <b>cbBufferSize</b> : size of pwszBuffer (including trailing 0). DWORD* <b>pcbNeededChars</b> : filled with the count of the written bytes, or needed buffer size.	Returns warning and error information of the read process

## 3. ILib3MFModelResourceIterator

ILib3MFModelResourceIterator is a helper class to iterate through arbitrary lists of 3MF resources.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
MoveNext	BOOL* <b>pbHasNext</b> : returns, if there is a resource to use	Iterates to the next resource in the list.

MovePrevious	BOOL* <b>pbHasPrevious</b> : returns, if there is a resource to use	Iterates to the previous resource in the list.
GetCurrent	ILib3MFModelResource** <b>ppResourceInstance</b> : returns the resource instance	Returns the resource the iterator points at.
Clone	ILib3MFModelResourceIterator** <b>ppIterator</b> : returns the cloned Iterator instance	Creates a new resource iterator with the same resource list.

#### 4. ILib3MFModelResource

ILib3MFModelResource is a base interface for all 3MF Resources.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
GetResourceID	DWORD* <b>pnResourceID</b> : Filled with the ID of the Resource Instance	Retrieves the ID of a Model Resource Instance

#### 5. ILib3MFModelObjectResource

ILib3MFModelObjectResource is a base interface for all 3MF Object Resources, i.e. Mesh Objects and Component Objects.

Parent interface: *ILib3MFModelResource*

Method	Parameters	Description
GetType	DWORD* <b>pObjectType</b> : returns object type constant. See ModelTypes.h for more information.	Retrieves an object's type.
SetType	DWORD <b>ObjectType</b> : object type constant. See ModelTypes.h for more information	Sets an object's type.
GetName	LPWSTR <b>pwszBuffer</b> : buffer to fill ULONG <b>cbBufferSize</b> : size of buffer to fill. needs to be at least string length + 1 ULONG* <b>pcbNeededChars</b> : returns needed characters in buffer	Retrieves an object's name string.
SetName	LPCWSTR <b>pwszName</b> : new name of the object. (e.g. "Car")	Sets an object's name string.
GetPartNumber	LPWSTR <b>pwszBuffer</b> : buffer to fill ULONG <b>cbBufferSize</b> : size of buffer to fill. needs to be at least string length + 1 ULONG <b>pcbNeededChars</b> : returns needed characters in buffer	Retrieves an object's part number string.

SetPartNumber	LPCWSTR <b>pwszPartNumber</b> : new part number string for referencing parts from an outside context.	Sets an object's part number string.
---------------	---	--------------------------------------

## 6. ILib3MFModelMeshObject

ILib3MFModelMeshObject encapsulates all methods for handling 3MF mesh objects.

Parent interface: *ILib3MFModelObjectResource*

Method	Parameters	Description
GetVertexCount	DWORD* <b>pnVertexCount</b> : filled with the vertex count	Returns the vertex count of a mesh object.
GetTriangleCount	DWORD* <b>pnTriangleCount</b> : filled with the triangle count	Returns the triangle count of a mesh object.
GetVertex	DWORD <b>nIndex</b> : Index of the vertex (0 to vertexcount - 1) MODELMEHVEX* <b>pVertex</b> : filled with the vertex coordinates	Returns coordinates of a single vertex of a mesh object.
SetVertex	DWORD <b>nIndex</b> : Index of the vertex (0 to vertexcount - 1) MODELMEHVEX* <b>pVertex</b> : contains the vertex coordinates	Sets the coordinates of a single vertex of a mesh object.
AddVertex	MODELMEHVEX* <b>pVertex</b> : contains the vertex coordinates DWORD* <b>pnIndex</b> : filled with the new Index of the vertex	Adds a single vertex to a mesh object.
GetTriangle	DWORD <b>nIndex</b> : Index of the triangle (0 to trianglecount - 1) MODELMEHTRIANGLE* <b>pTriangle</b> : filled with the triangle indices	Returns indices of a single triangle of a mesh object.
SetTriangle	DWORD <b>nIndex</b> : Index of the triangle (0 to trianglecount - 1) MODELMEHTRIANGLE* <b>pTriangle</b> : contains the triangle indices	Sets the indices of a single triangle of a mesh object
AddTriangle	MODELMEHTRIANGLE* <b>pTriangle</b> : contains the triangle indices DWORD* <b>pnIndex</b> : filled with the new Index of the vertex	Adds a single triangle to a mesh object.
GetVertices	MODELMEHVEX* <b>pVertices</b> : buffer filled with the vertex coordinates DWORD <b>nBufferSize</b> : size of the buffer in elements, must be at least vertexcount DWORD* <b>pnVertexCount</b> : returns how many vertices have been written	Retrieves all vertex coordinates of a mesh object.



GetTriangleIndices	MODELMEHTRIANGLE* <b>pIndices</b> : buffer filled with the triangle indices DWORD <b>nBufferSize</b> : size of the buffer in elements, must be at least trianglecount DWORD* <b>pnTriangleCount</b> : returns how many triangles have been written	Retrieves all triangle indices of a mesh object.
SetGeometry	MODELMEHVERTEX* <b>pVertices</b> : Array of vertex coordinates DWORD <b>nVertexCount</b> : Size of the vertex array MODELMEHTRIANGLE* <b>pTriangles</b> : Array of triangle indices DWORD <b>nTriangleCount</b> : Size of the triangle array	Sets the whole geometry of a mesh object.

## 7. ILib3MFModelComponentsObject

ILib3MFModelComponentsObject encapsulates all methods for handling 3MF component objects.

Parent interface: *ILib3MFModelObjectResource*

Method	Parameters	Description
AddComponent	ILib3MFModelObjectResource* <b>pObject</b> : object to add as component. May not lead to circular references! MODELTRANSFORM* <b>pmTransform</b> : optional transform matrix for the component ILib3MFModelComponent** <b>ppComponent</b> : returns new component instance	Adds a new component to a component object.
GetComponent	DWORD <b>nIndex</b> : index of the component to retrieve (0 to componentcount - 1) ILib3MFModelComponent** <b>ppComponent</b> : retrieves component instance	Retrieves a component from a component object.
GetComponentCount	DWORD* <b>pComponentCount</b> : returns the component count	Retrieves the component count of a component object.

## 8. ILib3MFModelComponent

ILib3MFModelComponent encapsulates one component node of a 3MF component object. It links to other object resources of the same model.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
--------	------------	-------------

GetObjectResourceID	DWORD* <b>pnResourceID</b> : returns the associated resource ID	Returns the associated resource ID of the component.
GetObjectResource	ILib3MFModelObjectResource** <b>ppResource</b> : returns the associated resource instance	Returns the associated resource Instance of the component.
GetTransform	MODELTRANSFORM* <b>pmTransformation</b> : filled with the component transformation matrix.	Returns the transformation matrix of the component.
SetTransform	MODELTRANSFORM* <b>pmTransformation</b> : new transformation matrix	Sets the transformation matrix of the component.
HasTransform	BOOL* <b>pbHasTransform</b> : if true is returned, the transformation is not equal to the identity	Returns, if the component has a different transformation than the identity matrix.

## 9. ILib3MFModelBuildItemIterator

ILib3MFModelBuildItemIterator is a helper class to iterate through arbitrary lists of 3MF build items.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
MoveNext	BOOL* <b>pbHasNext</b> : returns, if there is a build item to use	Iterates to the next build item in the list.
MovePrevious	BOOL* <b>pbHasPrevious</b> : returns, if there is a build item to use	Iterates to the previous build item in the list.
GetCurrent	ILib3MFModelBuildItem** <b>ppBuildItemInstance</b> : returns the build item instance	Returns the build item the iterator points at.
Clone	ILib3MFModelBuildItemIterator** <b>ppIterator</b> : returns the cloned Iterator instance	Creates a new build item iterator with the same build item list.

## 10. ILib3MFModelBuildItem

ILib3MFModelBuildItem encapsulates all methods for handling 3MF build items.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
GetObjectResourceID	DWORD* <b>pnID</b> : returns the associated resource ID	Retrieves the object resource associated to a build item.

GetObjectResource	ILib3MFModelObjectResource** <b>ppResource</b> : returns the associated resource instance	Returns the associated resource Instance of the build item.
GetObjectTransform	MODELTRANSFORM* <b>pmTransformation</b> : filled with the component transformation matrix.	Returns the transformation matrix of the build item.
SetObjectTransform	MODELTRANSFORM* <b>pmTransformation</b> : new transformation matrix	Sets the transformation matrix of the build item.
HasObjectTransform	BOOL* <b>pbHasTransform</b> : if true is returned, the transformation is not equal to the identity	Returns, if the build item has a different transformation than the identity matrix.
GetPartNumber	LPWSTR <b>pwszBuffer</b> : buffer to fill ULONG <b>cbBufferSize</b> : size of buffer to fill. needs to be at least string length + 1. ULONG* <b>pcbNeededChars</b> : returns needed characters in buffer	Retrieves a build item's part number string.
SetPartNumber	LPCWSTR <b>pwszPartNumber</b> : new part number string for referencing parts from the outside world.	Sets a build item's part number string

## 11. ILib3MFModel

ILib3MFModel is the basic instance owning all In-Memory elements of a 3MF file.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
SetUnit	DWORD <b>Unit</b> : enum value for the model unit (see NMR_ModelTypes.h for details)	sets the units of a model
GetUnit	DWORD* <b>pUnit</b> : enum value for the model unit (see NMR_ModelTypes.h for details)	retrieves the units of a model
SetLanguage	LPCWSTR <b>pwszLanguage</b> : Language string identifier	sets the language of a model
GetLanguage	LPWSTR <b>pwszBuffer</b> : buffer to fill ULONG <b>cbBufferSize</b> : size of buffer to fill. needs to be at least string length + 1. ULONG* <b>pcbNeededChars</b> : returns needed characters in buffer	retrieves the language of a model
QueryWriter	LPCWSTR <b>pwszWriterClass</b> : string identifier for the file (currently "stl" and "3mf") ILib3MFModelWriter** <b>ppWriter</b> : returns the writer instance	creates a model writer instance for a specific file type
QueryReader	LPCWSTR <b>pwszReaderClass</b> : string identifier for the file (currently "stl" and "3mf") ILib3MFModelReader ** <b>ppReader</b> : returns the reader instance	creates a model reader instance for a specific file type

GetResourceByID	DWORD <b>nResourceID</b> : Resource ID ILib3MFModelResource** <b>ppResource</b> : returns the resource instance	finds a model resource by its id
GetBuildItems	ILib3MFModelBuildItemIterator** <b>ppIterator</b> : returns the iterator instance	creates a build item iterator instance with all build items
GetResources	ILib3MFModelResourceIterator** <b>ppIterator</b> : returns the iterator instance	creates a resource iterator instance with all resources
GetObjects	ILib3MFModelResourceIterator** <b>ppIterator</b> : returns the iterator instance	creates a resource iterator instance with all object resources
GetMeshObjects	ILib3MFModelResourceIterator** <b>ppIterator</b> : returns the iterator instance	creates a resource iterator instance with all mesh object resources
GetComponentsObjects	ILib3MFModelResourceIterator** <b>ppIterator</b> : returns the iterator instance	creates a resource iterator instance with all component object resources
MergeToModel	ILib3MFModel** <b>ppMergedModel</b> : returns the merged model instance	merges all components and objects which are referenced by a build item. The memory is duplicated and a new model is created.
AddMeshObject	ILib3MFModelMeshObject** <b>ppMeshObject</b> : returns the mesh object instance	adds an empty mesh object to the model
AddComponentsObject	ILib3MFModelComponentsObject** <b>ppComponentsObject</b> : ppComponentsObject returns the component object instance	adds an empty component object to the model
AddBuildItem	ILib3MFModelObjectResource** <b>pObject</b> : Object instance associated with the build item MODELTRANSFORM* <b>pTransform</b> : Transformation matrix to be used ILib3MFModelBuildItem** <b>ppBuildItem</b> : returns the build item instance	adds a build item to the model

## 12. ILib3MFModelFactory

ILib3MFModelFactory is the global factory class for model instances.

Parent interface: *ILib3MFBase*

Method	Parameters	Description
CreateModel	ILib3MFModel** <b>ppModel</b> : returns created model instance	creates an empty model instance
GetSpecVersion	DWORD* <b>pMajorVersion</b> : returns the major version of the specification	retrieves the current version of the 3MF implementation and specification

	DWORD* <b>pMinorVersion</b> : returns the major version of the specification	
GetInterfaceVersion	DWORD* <b>pInterfaceVersion</b> : returns the interface version of the DLL	retrieves the current interface version of the library (build version). this version will increment with each official release of the library, and may be used to ensure API compatibility.

---

## License

The 3MF Library is released under the simplified BSD License (<http://opensource.org/licenses/BSD-2-Clause>). In short, this means it is allowed to use this code in closed source applications, open source applications and web services at no charge.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---

## Contributors

The current code base is maintained by the 3MF consortium, and is assembled with contributions from the following consortium members:

- Microsoft Corporation, Redmond, WA
- netfabb GmbH, Lupburg, Germany

The development has just begun. We invite everyone interested to contribute test results, bug reports, suggestions or code contributions under the simplified BSD License. We are actively looking for testers on all different platforms and in all different programming languages.

If you are making language bindings for your favourite language, we plead with you to release it public into the Lib3MF repository, as this will enable others to spread the format in an easy way.

The current version of this document and the library code can be obtained from github at <https://github.com/3MFConsortium/lib3mf>.

For more information, contact the 3MF Working Group at <http://3mf.io> or send a mail to [lib3mf@netfabb.com](mailto:lib3mf@netfabb.com).