

# **NVIDIA Material Definition Language 1.5**

## **Appendix E – MDL Internationalization**

Language Specification

Document version 1.5.1  
November 26, 2018

**DRAFT — Pre-release**

## Copyright Information

© 2018 NVIDIA Corporation. All rights reserved.

Document build number 312834

# LICENSE AGREEMENT

## License Agreement for NVIDIA MDL Specification

**IMPORTANT NOTICE – READ CAREFULLY:** This License Agreement (“License”) for the NVIDIA MDL Specification (“the Specification”), is the LICENSE which governs use of the Specification of NVIDIA Corporation and its subsidiaries (“NVIDIA”) as set out below. By copying, or otherwise using the Specification, You (as defined below) agree to be bound by the terms of this LICENSE. If You do not agree to the terms of this LICENSE, do not copy or use the Specification.

## RECITALS

This license permits you to use the Specification, without modification for the purposes of reading, writing and processing of content written in the language as described in the Specification, such content may include, without limitation, applications that author MDL content, edit MDL content including material parameter editors and applications using MDL content for rendering.

### 1. DEFINITIONS.

Licensee. “Licensee,” “You,” or “Your” shall mean the entity or individual that uses the Specification.

### 2. LICENSE GRANT.

- 2.1. NVIDIA hereby grants you the right, without charge, on a perpetual, non- exclusive and worldwide basis, to utilize the Specification for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, products complying with the Specification, in all cases subject to the conditions set forth in this Agreement and any relevant patent (save as set out below) and other intellectual property rights of third parties (which may include NVIDIA). This license grant does not include the right to sublicense, modify or create derivatives of the Specification. For the avoidance of doubt, products implementing this Specification are not deemed to be derivative works of the Specification.
- 2.2. NVIDIA may have patents and/or patent applications that are necessary for you to license in order to make, sell, or distribute software programs that are based on the Specification (“Licensed Implementations”).
- 2.3. Except as provided below, NVIDIA hereby grants you a royalty-free license under NVIDIA’s Necessary Claims to make, use, sell, offer to sell, import, and otherwise distribute Licensed Implementations. The term “Necessary Claims” means claims of a patent or patent application (including continuations, continuations-in-part, or reissues) that are owned or controlled by NVIDIA and that are necessarily infringed by the Licensed Implementation. A claim is necessarily infringed only when it is not possible to avoid infringing when implementing the Specification. Notwithstanding the foregoing, “Necessary Claims” do not include any claims that would require a payment of royalties by NVIDIA to unaffiliated third parties.

### 3. NO WARRANTIES.

- 3.1. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SPECIFICATION IS PROVIDED “AS IS” AND NVIDIA AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ACCURACY, COMPLETENESS AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.
- 3.2. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NVIDIA OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SPECIFICATION, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 4. NO SUPPORT.

NVIDIA has no obligation to support or to provide any updates of the SPECIFICATION.

### 5. FEEDBACK.

In the event Licensee contacts NVIDIA to request Feedback (as defined below) on how to design, implement, or optimize Licensee’s product for use with the SPECIFICATION, the following terms and conditions apply to the Feedback:

- 5.1. Exchange of Feedback. Both parties agree that neither party has an obligation to give the other party any suggestions, comments or other feedback, whether orally or otherwise (“Feedback”), relating to (i) the SPECIFICATION; (ii) Licensee’s products; (iii) Licensee’s use of the SPECIFICATION; or (iv) optimization of Licensee’s product with the SPECIFICATION. In the event either party provides Feedback to the other party, the party receiving the Feedback may use and include any Feedback that the other party voluntarily provides to improve the (i) SPECIFICATION or other related NVIDIA technologies, respectively for the benefit of NVIDIA; or (ii) Licensee’s product or other related Licensee technologies, respectively for the benefit of Licensee. Accordingly, if either party provides Feedback to the other party, both parties agree that the other party and its respective licensees may freely use, reproduce, license, distribute, and otherwise commercialize the Feedback in the (i) SPECIFICATION or other related technologies; or (ii) Licensee’s products or other related technologies, respectively, without the payment of any royalties or fees.

5.2. Residual Rights. Licensee agrees that NVIDIA shall be free to use any general knowledge, skills and experience, (including, but not limited to, ideas, concepts, know-how, or techniques) (“Residuals”), contained in the (i) Feedback provided by Licensee to NVIDIA; (ii) Licensee’s products shared or disclosed to NVIDIA in connection with the Feedback; or (c) Licensee’s confidential information voluntarily provided to NVIDIA in connection with the Feedback, which are retained in the memories of NVIDIA’s employees, agents, or contractors who have had access to such (i) Feedback provided by Licensee to NVIDIA; (ii) Licensee’s products; or (c) Licensee’s confidential information voluntarily provided to NVIDIA, in connection with the Feedback. Subject to the terms and conditions of this Agreement, NVIDIA’s employees, agents, or contractors shall not be prevented from using Residuals as part of such employee’s, agent’s or contractor’s general knowledge, skills, experience, talent, and/or expertise. NVIDIA shall not have any obligation to limit or restrict the assignment of such employees, agents or contractors or to pay royalties for any work resulting from the use of Residuals. FEEDBACK FROM EITHER PARTY IS PROVIDED FOR THE OTHER PARTY’S USE “AS IS” AND BOTH PARTIES DISCLAIM ALL WARRANTIES, EXPRESS, IMPLIED AND STATUTORY INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. BOTH PARTIES DO NOT REPRESENT OR WARRANT THAT THE FEEDBACK WILL MEET THE OTHER PARTY’S REQUIREMENTS OR THAT THE OPERATION OR IMPLEMENTATION OF THE FEEDBACK WILL BE UNINTERRUPTED OR ERROR-FREE.

6. NO IMPLIED LICENSES.

Under no circumstances should anything in this Agreement be construed as NVIDIA granting by implication, estoppel or otherwise, (i) a license to any NVIDIA product or technology other than the SPECIFICATION; or (ii) any additional license rights for the SPECIFICATION other than the licenses expressly granted in this Agreement.

7. THIRD PARTY RIGHTS.

Without limiting the generality of Section 3 above, NVIDIA ASSUMES NO RESPONSIBILITY TO COMPILE, CONFIRM, UPDATE OR MAKE PUBLIC ANY THIRD PARTY ASSERTIONS OF PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS THAT MIGHT NOW OR IN THE FUTURE BE INFRINGED BY AN IMPLEMENTATION OF THE SPECIFICATION IN ITS CURRENT, OR IN ANY FUTURE FORM. IF ANY SUCH RIGHTS ARE DESCRIBED ON THE SPECIFICATION, NVIDIA TAKES NO POSITION AS TO THE VALIDITY OR INVALIDITY OF SUCH ASSERTIONS, OR THAT ALL SUCH ASSERTIONS THAT HAVE OR MAY BE MADE ARE SO LISTED.

8. TERMINATION OF LICENSE.

This LICENSE will automatically terminate if Licensee fails to comply with any of the terms and conditions hereof. In such event, Licensee must destroy all copies of the SPECIFICATION and all of its component parts.

9. DEFENSIVE SUSPENSION.

If Licensee commences or participates in any legal proceeding against NVIDIA, then NVIDIA may, in its sole discretion, suspend or terminate all license grants and any other rights provided under this LICENSE during the pendency of such legal proceedings.

10. ATTRIBUTION.

Licensee shall, in a manner reasonably acceptable to NVIDIA prominently embed in all products produced in compliance with the Specification a notice stating that such product has been so produced.

11. MISCELLANEOUS.

11.1. Notices. All notices required under this Agreement shall be in writing, and shall be deemed effective five days from deposit in the mails. Notices and correspondence to either party shall be sent to its address as it appears below.

General Counsel, NVIDIA Corporation 2701 San Tomas Expressway  
Santa Clara, CA 95050.

11.2. Export regulations. The Specification, or portions thereof, including technical data, may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Licensee agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain all export, re-export, import or other licenses in connection with its use of the Specification or any product complying with the Specification.

# Table of Contents

25	Appendix E – MDL internationalization — DRAFT .....	1
25.1	Translated annotation parameters .....	1
25.2	File names and locations .....	1
25.2.1	Module XLIFF files .....	1
25.2.2	Package XLIFF files .....	1
25.2.3	XLIFF file name resolution .....	2
25.2.4	XLIFF files in MDL archives .....	2
25.2.5	XLIFF file context .....	2
25.3	XLIFF elements used for MDL internationalization .....	2
25.3.1	Translation units .....	2
25.3.2	Scope of the translation units .....	3
25.3.3	Groups and group context .....	3
25.3.4	Translation lookup order .....	4
26	Bibliography .....	5
27	Changes to this document .....	7
27.1	Changes for version 1.5.1 .....	7



## 25 Appendix E – MDL internationalization — DRAFT

**MDL 1.5** MDL defines a framework for the internationalization of MDL annotation parameters of type string. The translations are stored in the XLIFF 1.2 file format [1]. Following the standard, XLIFF files use the `.xlf` file name extension. XLIFF files are stored in MDL search paths and can be embedded in MDL archives. Each XLIFF file is specific to a *locale*, i.e., a translation language. The locale must be a 2-letter code defined by the ISO 639-1:2002 standard [2].

### 25.1 Translated annotation parameters

All annotation parameters of type string can be translated using this framework. In the standard library, translation is limited to the following annotations only (Section 18):

<code>anno::display_name</code>	<code>anno::description</code>	<code>anno::copyright_notice</code>
<code>anno::author</code>	<code>anno::contributor</code>	<code>anno::key_words</code>
<code>anno::unused</code>	<code>anno::deprecated</code>	<code>anno::in_group</code>

### 25.2 File names and locations

XLIFF files used for MDL translations are located either in a directory or sub-directory of an MDL search path (Section 2.2) or inside an MDL archive file (Section 24). Their file paths are relative to a search root, see Section 25.2.3.

An XLIFF file can be a *module XLIFF file*, which provides translations for one language for an MDL module only, or a *package XLIFF file*, which provides translations for one language for a package with all its modules and sub-packages.

#### 25.2.1 Module XLIFF files

A module XLIFF file is placed in a package or at the root of an MDL search path. Its name is

```
<module>_<locale>.xlf
```

where `<module>` is the MDL module name and `<locale>` is the locale. The file path of the module XLIFF file must be the same as the file path of the module that it applies to.

#### 25.2.2 Package XLIFF files

A package XLIFF file is placed in a package or at the root of an MDL search path. Its name is:

```
<locale>.xlf
```

where `<locale>` is the locale. The file path of the package XLIFF file must be the same as the file path of the package or search root that it applies to.

### 25.2.3 XLIFF file name resolution

The file paths for XLIFF files are not locations in the file system, but only references in relation to the search paths. The translation of a file path to a file system location is the *file resolution*, which follows the same rules as the file path resolutions for the absolute file paths of an MDL module, see Section 2.2.

This implies that XLIFF files for a module or package do not have to be in the same directory as the MDL module or package itself. For example, the XLIFF file for the module file `m.mdl` located under `<search-path-root-1>` could be located under `<search-path-root-2>`, which can be used to install translation packs in locations independent from the MDL modules themselves:

```
<search-path-root-1>
+--- parent
+--- m.mdl

<search-path-root-2>
+--- parent
+--- m_fr.xlf
```

### 25.2.4 XLIFF files in MDL archives

When provided inside MDL archives (Section 24), XLIFF files follow the same conventions as in a regular package structure with the exception that a package translation file is not allowed in the archive root directory but only below that. A module translation file is allowed in the root directory of an archive.

### 25.2.5 XLIFF file context

The *XLIFF file context* is a fully qualified name that controls the scope of strings that are translated by this file. The context is defined by the location of the file in relation to its MDL search path root, and analogously in MDL archive files: A module XLIFF file context is the absolute module name of its respective module, and a package XLIFF file context is the absolute package name of its respective package.

In the following example, the XLIFF file context for `fr.xlf` is `::parent` and the `m_fr.xlf` context is `::parent::m`:

```
<search-path-root>
+--- parent
+--- m.mdl
+--- fr.xlf
+--- m_fr.xlf
```

## 25.3 XLIFF elements used for MDL internationalization

XLIFF is an XML-based standard. This framework makes in particular use of the following tag elements and attributes of the XLIFF standard.

### 25.3.1 Translation units

A translation unit element (`<trans-unit>`) contains a source (`<source>`) and a target (`<target>`) element.

A source element defines the string which needs to be translated. A target element represents the result of the translation of the source element in the target language. The required `id` attribute is used to uniquely identify the translation unit within all translation units. The optional `note` (`<note>`) element can be used to add comments about the translation unit.

### 25.3.2 Scope of the translation units

The *scope* of a translation unit is a fully qualified MDL name. The scope determines for which MDL elements this translation unit is used for. The scope is either implicit, in which case it is the XLIFF file context, or explicit, in which case it is the group context specified by the `resname` attribute of a group element.

### 25.3.3 Groups and group context

The optional `group` (`<group>`) element specifies a set of elements that should be processed together as a translation unit.

Groups can specify a *group context* with the optional `resname` attribute, which restricts the scope of the translation in this group to a specific MDL element. The group context is a *relative qualified identifier name*, which is the identifier of the MDL element, with module and package names preceding it, all separated by the scope operator `‘:.’`. It does not start with a scope operator. The group context is given relative to the XLIFF file position such that the concatenation of the XLIFF file context and the group context, separated by the scope operator `‘:.’`, form the fully qualified identifier of the MDL element.

The following example MDL file structure illustrates the use of contexts:

```
<search-path-root>
+--- parent
    +--- example_material.mdl      // contains 'example' material
    +--- example_material_fr.xlf
    +--- fr.xlf
```

The package XLIFF file `fr.xlf` for a French translation uses the `resname` attribute to set the group context and restrict the translation to a material `example` in the `example_material` module:

```
<group resname="example_material::example">
  <trans-unit id="anno_0">
    <source>Example material</source>
    <target>Exemple de matériel</target>
    <note>Description for example_material::example</note>
  </trans-unit>
</group>
```

Setting the group context for the same material in the module XLIFF file `example_material_fr.xlf` for a French translation is shown in the following example. Note that the group context in the module XLIFF file does not mention the `example_material` module, which is now part of the XLIFF file context.

```
<group resname="example">
  <trans-unit id="anno_0">
```

```
<source>Example material</source>
<target>Exemple de matériel</target>
<note>Description for example_material::example</note>
</trans-unit>
</group>
```

### 25.3.4 Translation lookup order

When an annotation text in a certain module needs to be translated, the corresponding XLIFF file is looked for in this order:

1. The module XLIFF file for the module,
2. the package XLIFF file for the modules package, and
3. the package XLIFF file for any of the modules package parents up to MDL search path root.

In the following example with a module `::parent::m`:

```
<search-path-root>
+--- parent
  +--- m.mdl
  +--- m_fr.xlf // first lookup
  +--- fr.xlf // second lookup
  +--- fr.xlf // third lookup
```

the translation of an annotation from `m.mdl` will be looked up in:

1. `parent/m_fr.xlf` (context: `::parent::m`)
2. `parent/fr.xlf` (context: `::parent`)
3. `fr.xlf` (context empty)

**Note:** A function variant (Section 12.6) or material variant (Section 13.6) has no place to define own annotations on parameters. Their parameters are defined by the function or material definition on which this variant is based on. This implies that the respective module for parameter translations of annotations is not the module of a variant definition, but the module of the non-variant definition for those parameters.

## 26 Bibliography

- [1] OASIS Standard. *XML Localisation Interchange File Format (XLIFF)*, Version 1.2, February 1, 2008. (<https://docs.oasis-open.org/xliff/xliff-core/xliff-core.html>)
- [2] International Standard ISO 639-1:2002. *Codes for the representation of names of languages ^Ö Part 1: Alpha-2 code*. 2002. ([https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes))



## 27 Changes to this document

Main changes for MDL 1.5 since the MDL 1.4 specification document version 1.4.4 from June 4, 2018.

### 27.1 Changes for version 1.5.1

- Updated version to 1.5.
- Added new Appendix E for MDL internationalization. (Page 1)