



COLLADA – Digital Asset Schema リリース 1.5.0

リリースノート

2008年6月

編集者: Mark Barnes および Ellen Levy Finch, Sony Computer Entertainment Inc.



© 2005-2009 The Khronos Group Inc., Sony Computer Entertainment Inc.

**All Rights Reserved.**

This specification is protected by copyright laws and contains material proprietary to the Khronos Group, Inc. It or any components may not be reproduced, republished, distributed, transmitted, displayed, broadcast, or otherwise exploited in any manner without the express prior written permission of Khronos Group. You may use this specification for implementing the functionality therein, without altering or removing any trademark, copyright, or other notice from the specification, but the receipt or possession of this specification does not convey any rights to reproduce, disclose, or distribute its contents, or to manufacture, use, or sell anything that it may describe, in whole or in part.

Khronos Group grants express permission to any current Promoter, Contributor, or Adopter member of Khronos to copy and redistribute UNMODIFIED versions of this specification in any fashion, provided that NO CHARGE is made for the specification and the latest available update of the specification for any version of the API is used whenever possible. Such distributed specification may be reformatted AS LONG AS the contents of the specification are not changed in any way. The specification may be incorporated into a product that is sold as long as such product includes significant independent work developed by the seller. A link to the current version of this specification on the Khronos Group website should be included whenever possible with specification distributions.

Khronos Group makes no, and expressly disclaims any, representations or warranties, express or implied, regarding this specification, including, without limitation, any implied warranties of merchantability or fitness for a particular purpose or noninfringement of any intellectual property. Khronos Group makes no, and expressly disclaims any, warranties, express or implied, regarding the correctness, accuracy, completeness, timeliness, and reliability of the specification. Under no circumstances will the Khronos Group, or any of its Promoters, Contributors, or Members or their respective partners, officers, directors, employees, agents, or representatives be liable for any damages, whether direct, indirect, special, or consequential damages for lost revenues, lost profits, or otherwise, arising from or in connection with these materials.

Khronos is a trademark of The Khronos Group Inc.

COLLADA is a trademark of Sony Computer Entertainment Inc. used by permission by Khronos.

All other trademarks are the property of their respective owners and/or their licensors.

**Publication date: May 2009**

Khronos Group  
14525 SW Millikan Way #45043  
Beaverton, Oregon 97005-2343 USA

Sony Computer Entertainment Inc.  
2-6-21 Minami-Aoyama, Minato-ku,  
Tokyo 107-0062 Japan

Sony Computer Entertainment America  
919 E. Hillsdale Blvd.  
Foster City, CA 94404, U.S.A.

Sony Computer Entertainment Europe  
30 Golden Square  
London W1F 9LD, U.K.



# 目次

COLLADA 1.5.0 スキーマと仕様についての リリースノート .....	1
このドキュメントについて .....	1
追加のウェブリソース .....	1
本ドキュメントの表記法 .....	1
COLLADA 全般の変更 .....	1
COLLADA 文書のインポータおよびエクスポータは、.zax アーカイブ形式を必ずサポ ートする必要があります。 .....	1
バージョン 1.4.1 以降のスキーマの変更 .....	2
COLLADA 1.4.1 と 1.5.0 .....	2
(全般) インポートされた名前空間 (xml:base 要素用) で使用されるスキーマの場所 が更新できるようになりました。 .....	2
(新規) COLLADA は、キネマティクスをサポートするようになりました。 .....	2
(新規) COLLADA は、境界表現 (B-rep) をサポートするようになりました。 .....	4
(全般) 名前引数は、すべて xs:token 型になりました。 .....	5
(全般) SID および SID 参照用の型が追加されました。 .....	5
(FX) <argument> の unit 属性の名前と型が変更されました。 .....	5
(FX) <array> に、resizable 属性が追加されました。 .....	5
(コア) <asset> の子要素として <extra> が追加されました。 .....	5
(コア) <asset> に地理的範囲の情報が追加されました。 .....	6
(フィジックス) <capsule> の <radius> 子要素は、値を 3 つ持つようになりました。 .....	6
(コア) <COLLADA> の version 属性は、バージョン 1.5.0 をサポートするようにな りました。 .....	6
(コア) <COLLADA> にライブラリが追加されました。 .....	6
(FX) <color_target>、<depth_target>、<stencil_target> は、画像を直接参照するよ うになりました。 .....	7
(FX) <connect_param> は削除されました。 .....	7
(コア) <contributor> に電子メールやウェブサイトに関する子要素が追加されまし た。 .....	7
(FX) 画像テクスチャの初期化をカスタマイズするために、<create_2d>、 <create_2d>、<create_cube> が追加されました。 .....	7
(フィジックス) <instance_rigid_body> および <rigid_body> 中の <dynamic> は、デ フォルト値を持つようになりました。 .....	8
(コア) 拡張性を高めるため、<evaluate_scene> および <render> の設計が変更され ました。 .....	8
(コア) <float_array> の型制約がより厳密になりました。 .....	9
(FX) <image> アセットを記述するための <format> が追加されました。 .....	9
(コア) MathML をサポートするために、<formula>、<instance_formula>、 <library_formulas> が追加されました。 .....	9
(FX) <generator> が削除されました。 .....	9
(FX) <image> が再設計され、<instance_image> が追加されました。 .....	9
(FX) 画像の初期化をよりうまく制御できるようにするために、<init_from> の設計が 変更されました。 .....	10
(コア) <instance_node> には、複数の表現や詳細レベル (LOD) をサポートするた めに、proxy 属性が追加されました。 .....	11
(FX) <pass> は設計が変更され、新しく子要素 <states> と <evaluate> が追加され ました。 .....	11
(FX) <profile_*> および </profile_*> / <technique> 子要素は削除されました。 .....	11

(FX) <profile_BRIDGE>が追加されました。 .....	11
(FX) <profile_GLES2>が追加されました。 .....	12
(FX) <profile_*>要素はすべて、id 属性を持つようになりました。 .....	12
(FX) <profile_GLSL>に、platform 属性が追加されました。 .....	12
(FX) シェーダの定義を改善するために、<program>要素が新しい子要素<compiler>、<linker>、<binary>、<bind_attribute>などを併せて追加されました。 .....	12
(FX) <render>が更新されました。 .....	13
(フィジックス) <rigid_body>に id 属性が追加されました。 .....	13
(コア) <sampler>に、最初のキーの前、および最後のキーの後のサンプル値を指定するための属性が追加されました。 .....	13
(FX) <sampler*>子要素の設計が変更されました。 .....	13
(FX) <sampler_image>が追加されました。 .....	14
(FX) <sampler_state>は、<sampler_states>として設計が変更されました。 .....	14
(FX) いくつかの要素から<setparam>が削除されました。 .....	14
(FX) <shader>の設計が変更されました。 .....	14
(コア) <SIDREF_array>が追加されました。 .....	15
(コア、フィジックス) <source>では、トークンや SIDREF の配列を利用できるようになりました。 .....	15
(FX) シェーダソースコードを結合するために、<sources> が追加されました。 .....	15
(FX) <surface>が削除されました。 .....	15
(コア) <tapered_capsule>および<tapered_cylinder>が削除されました。 .....	15
(コア) <technique>は、COLLADA 要素名との衝突を回避できるようになりました。 .....	16
(FX) <texenv>の unit 属性の名前と型が変更されました。 .....	16
(FX) <texture>の<extra>は、maxOccurs=unbounded を使うようになりました。 .....	16
(FX) <texture_pipeline>が出現できるのは、<states>の中だけになりました。 .....	16
(FX) <texture_unit>が削除されました。 .....	16
(コア) <token_array>が追加されました。 .....	16
(FX) <transparent>は、追加の不透明モードをサポートするようになりました。 .....	16
(FX) <usertype>の設計が変更されました。 .....	17
1.4.1 (第2版) 以降の仕様変更 .....	17
(全般) テキストにさまざまな修正を行いました。 .....	17
(索引) 「COLLADA 要素の索引」に主見出し項目のない要素が追加されました。 .....	17
(フィジックス) <convex_mesh>に出現する子要素の説明が訂正されました。 .....	18
(FX) <color_target>属性の型が訂正されました。 .....	18
(コア) <evaluate_scene>が文書化されました。 .....	18
(コア) <morph>属性の名前が、sid から source に修正されました。 .....	18
(コア) <orthographic>子要素が訂正されました。 .....	18
(コア) <point>子要素が訂正されました。 .....	18
仕様 1.5.0 に残っている問題点 .....	18
(本ドキュメント) 「その他の情報源」の訂正。 .....	18
(コア) <token_array>が文書化されていません。 .....	18
(FX) fx_sampler_type 型が誤って fx_sampler_common として参照されています。 .....	19
(コア、FX) 「パラメータの型要素」セクションへの参照は第 11 章を参照する必要があります。 .....	19

---

# COLLADA 1.5.0 スキーマと仕様についてのリリースノート

---

## このドキュメントについて

以下のリリースノートには、COLLADA Digital Asset Schema リリース 1.5.0 の変更の概要が記載されています。スキーマのバージョン 1.5.0、および「COLLADA – Digital Asset Schema リリース 1.5.0 - 仕様書」は、以下のサイトからダウンロードすることができます。

<http://www.khronos.org/collada/>

## 追加のウェブリソース

COLLADA に関する追加情報は、以下のサイトにあります。

- <http://collada.org>: COLLADA に関する追加の技術情報、COLLADA 拡張、プラグイン、コンディショナーの公開ディレクトリ、COLLADA に関する議論を行うための公開フォーラム
- <http://www.khronos.org/bugzilla/>: COLLADA の公式バグレポートシステム

## 本ドキュメントの表記法

本ドキュメントでは、COLLADA 公開バグトラッキングシステムの Bugzilla レポートを参照する際には、接頭辞「P」を付加し、Khronos メンバー専用 Bugzilla サイトからのレポートには、接頭辞「K」を付加しています。

---

## COLLADA 全般の変更

**COLLADA 文書のインポータおよびエクスポータは、.zae アーカイブ形式を必ずサポートする必要があります。**

*レポート K-2975 を解決します。*

これはスキーマの変更ではありませんが、COLLADA 文書のインポートやエクスポートを行うアプリケーションの要件についての変更です。詳しくは、仕様書の第 2 章「ツールの要件とオプション」の「アーカイブにパッケージ化する」セクションを参照してください。

## バージョン 1.4.1 以降のスキーマの変更

以下のリリースノートでは、ユーザが知っておくべきスキーマに対する変更点をすべてを挙げていますが、機能や実装の細部については説明していません。COLLADA のすべての機能についての詳細は、「COLLADA – Digital Asset Schema リリース 1.5.0 - 仕様書」を参照してください。また実装の詳細については、COLLADA 1.5.0 スキーマを参照してください。

### COLLADA 1.4.1 と 1.5.0

このスキーマリリースにおける変更には、既存の COLLADA 1.4.1 ドキュメントと互換性のないものもあります。ただし、1.4.1 から 1.5.0 への移行は必須ではなく、1.4.1 も依然として有効です。1.5.0 の機能を必要としない場合には、1.4.1 スキーマを使用し続けてください。1.5.0 機能が必要かどうかを判断するには、以下のリリースノートを注意深く読んでください。

### (全般) インポートされた名前空間 (xml:base 要素用) で使用されるスキーマの場所が更新できるようになりました。

レポート K-1885 を解決します。

xs 名前空間用の COLLADA `schemaLocation` は、以下のように継続的に修正されている XML Schema を指すようになりました。

```
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
```

### (新規) COLLADA は、キネマティクスをサポートするようになりました。

レポート K-2964 を解決します (仕様: K-2965)。

キネマティクスを使用すると、コンテンツ制作者は、ビジュアルシーンのオブジェクトに運動学的な性質を添付することが可能になります。

ビジュアルシーンのノードは、運動学的シミュレーションによって制御できます。これは、任意の数のキネマティクスモデルによって実現されます。キネマティクスモデルは、ジョイントとリンクから構成されます。キネマティクスモデルは、1つ以上の関節式システムによって制御できます。関節式システムは、キネマティクスモデルを強化して、運動学的な性質や動力学的な性質を加えます。

COLLADA キネマティクスの最上位の要素は、`<COLLADA>`の子要素 (`<COLLADA>`の注を参照) および `<scene>`の子要素である `<instance_kinematics_scene>`の複数のライブラリです。キネマティクスをサポートする要素には、以下のものがあります。詳しくは仕様書を参照してください。

- ジョイント

<code>&lt;joint&gt;</code>	1つ以上の自由度をもつ単一のジョイントを定義します。
<code>&lt;library_joints&gt;</code>	<code>&lt;joint&gt;</code> 要素のモジュールを宣言します。
<code>&lt;prismatic&gt;</code>	ジョイントに単一の並進自由度を定義します。
<code>&lt;revolute&gt;</code>	ジョイントに単一の回転自由度を定義します。

- キネマティクスモデル

<code>&lt;attachment_end&gt;</code>	アタッチメント中の閉じたループの一方の端を定義します。
<code>&lt;attachment_full&gt;</code>	2つのリンクを連結します。
<code>&lt;attachment_start&gt;</code>	2つのリンクを連結して、閉じたループの一方の端を定義します。
<code>&lt;instance_joint&gt;</code>	COLLADA ジョイントリソースをインスタンス化します。



<code>&lt;instance_kinematics_model&gt;</code>	COLLADA <code>&lt;kinematics_model&gt;</code> リソースをインスタンス化します。
<code>&lt;kinematics_model&gt;</code>	キネマティクスモデルを記述します。
<code>&lt;library_kinematics_models&gt;</code>	<code>&lt;kinematics_model&gt;</code> 要素を保存するライブラリを提供します。
<code>&lt;link&gt;</code>	運動が1つ以上のジョイントによって拘束される質量のない運動学的な剛体を表現します。

- 関節式システム

<code>&lt;articulated_system&gt;</code>	キネマティクスシステムの汎用的な制御情報の宣言をカテゴリ化します。
<code>&lt;axis_info&gt;</code>	関節式モデルのキネマティクス、つまり運動学的な運動の挙動を記述するための軸情報を含みます。
<code>&lt;bind&gt;</code> (キネマティクス)	インスタンス化時に、入力をキネマティクスパラメータにバインドします。
<code>&lt;connect_param&gt;</code>	事前に定義した2つのパラメータの間にシンボリック接続を作成します。
<code>&lt;effector_info&gt;</code>	エフェクタに追加の動力学情報を指定します。
<code>&lt;frame_object&gt;</code>	キネマティクスの計算に使われる座標系の情報が含まれます。
<code>&lt;frame_origin&gt;</code>	キネマティクスの計算に使われる座標系の情報が含まれます。
<code>&lt;frame_tcp&gt;</code>	キネマティクスの計算に使われる座標系の情報が含まれます。
<code>&lt;frame_tip&gt;</code>	キネマティクスの計算に使われる座標系の情報が含まれます。
<code>&lt;instance_articulated_system&gt;</code>	COLLADA <code>&lt;articulated_system&gt;</code> リソースをインスタンス化します。
<code>&lt;kinematics&gt;</code>	関節式モデルの運動学的挙動を記述するための追加情報が含まれます。
<code>&lt;library_articulated_systems&gt;</code>	<code>&lt;articulated_system&gt;</code> 要素を保存するライブラリを提供します。
<code>&lt;motion&gt;</code>	関節式モデルの動力学挙動を記述するための追加情報が含まれます。

- キネマティクスシーン: キネマティクスシーンは、具体的なシーン用にインスタンス化されたモデルです。キネマティクスシーンは使われるリンクや、現在のシーンの設定、特にデフォルトや現在のジョイント値などを定義します。

<code>&lt;bind_joint_axis&gt;</code>	キネマティクスモデルのジョイント軸を特定のノード変換にバインドします。
<code>&lt;bind_kinematics_model&gt;</code>	キネマティクスモデルをノードにバインドします。
<code>&lt;instance_kinematics_scene&gt;</code>	COLLADA <code>&lt;kinematics_scene&gt;</code> リソースをインスタンス化します。
<code>&lt;kinematics_scene&gt;</code>	COLLADA リソースのコンテンツ中の関節式モデルに変換可能な情報セット全体を表します。
<code>&lt;library_kinematics_scenes&gt;</code>	<code>&lt;kinematics_scene&gt;</code> 要素を保存するライブラリを提供します。

**(新規) COLLADA は、境界表現 (B-rep) をサポートするようになりました。**

レポート K-2960 を解決します (仕様: K-2961)。

境界表現 (B-rep) モデルは、位相要素と幾何要素の 2 つの部分から構成されます。位相要素は、対応する制約のないジオメトリを限定するさまざまなエンティティ (頂点、辺など) を指定します。幾何学的エンティティとしては、直線、曲線、曲面などがあります。

最上位の要素は <brep> で、この要素は <geometry> の子要素です。B-rep をサポートする要素には、以下のものがあります。詳しくは仕様書を参照してください。

- ジオメトリ

<a href="#">brep</a>	境界表現 (B-rep) の構造を記述します。
----------------------	-------------------------

- 曲線

<a href="#">circle</a>	3次元空間中の円を記述します。
<a href="#">curve</a>	特定の曲線を記述します。
<a href="#">curves</a>	B-rep 構造で使われるあらゆる曲線が含まれます。
<a href="#">ellipse</a>	3次元空間中の楕円を記述します。
<a href="#">hyperbola</a>	3次元空間中の双曲線を記述します。
<a href="#">line</a>	3次元空間中の直線を記述します。
<a href="#">nurbs</a>	3次元空間中の NURBS 曲線を記述します。
<a href="#">parabola</a>	3次元空間中の放物線を記述します。
<a href="#">surface_curves</a>	B-rep 構造で使われるあらゆるパラメトリック曲線 (pcurve) が含まれます。

- 位相要素

<a href="#">edges</a>	B-rep 構造の辺を記述します。
<a href="#">faces</a>	B-rep 構造の面を記述します。
<a href="#">pcurves</a>	面のパラメータ空間の中で辺を表す方法を指定します。
<a href="#">shells</a>	B-rep 構造のシェルを記述します。
<a href="#">solids</a>	B-rep 構造のソリッドを記述します。
<a href="#">wires</a>	B-rep 構造のワイヤを記述します。

- 曲面

<a href="#">cone</a>	円錐面を記述します。
<a href="#">cylinder</a>	無限の円柱面を記述します。
<a href="#">nurbs_surface</a>	3次元空間中の NURBS 曲面を記述します。
<a href="#">plane</a> (フィジックス)	無限の平面を記述します。
<a href="#">sphere</a> (フィジックス)	中央揃えされた球体プリミティブを記述します。
<a href="#">surface</a>	特定の曲面を記述します。
<a href="#">surfaces</a>	B-rep 構造で使われるあらゆる曲面が含まれます。
<a href="#">swept_surface</a>	曲線の押し出しや回転によって曲面を記述します。
<a href="#">torus</a>	3次元空間中のトーラス (円環面) を記述します。

- 変換

<a href="#">orient</a>	物体座標系の方向を記述します。
<a href="#">origin</a>	物体座標系の原点を記述します。

**(全般) 名前引数は、すべて `xs:token` 型になりました。**

レポート P-9、K-532 を解決します (仕様: P-64)。

出現する要素の名前引数は、すべて `xs:token` 型になりました。これにより、引数値の単語の間にシングルスペースを挿入できるようになりました。

**(全般) SID および SID 参照用の型が追加されました。**

レポート K-1293 を解決します (仕様: K-1886、K-3154)。

新しい型が 2 つ追加されました。

- `sid_type`: スコープ付き識別子。sid 属性は、すべて `sid_type` 型になりました。
- `sidref_type`: スコープ付き識別子への参照。他の要素の SID を参照する要素および属性 (以下のリスト中では `xxx` で表されている) は、すべて `sidref_type` 型になりました。
  - `<channel target="xxx">`
  - `<renderstate_name param="xxx">`
  - `<bind target="xxx">`
  - `<instance_rigid_body body="xxx">`
  - `<instance_rigid_constraint constraint="xxx">`
  - `<param ref="xxx">`
  - `<texenv sampler="xxx">`
  - `<SIDREF>xxx</SIDREF>`
  - `<texture*><param>xxx</param></texture*>`

これに関連した新たな配列要素については、`<SIDREF_array>` の注を参照してください。

有効な SID および SIDREF について詳しくは、仕様書の第 3 章: 「スキーマのコンセプト」の「アドレス構文」を参照してください。

**(FX) `<argument>` の `unit` 属性の名前と型が変更されました。**

解決: レポート未提出。

`<argument>` の `unit` 属性は、名前が `sampler` に変更され、型もサンブラを指す `sidref_type` 型になりました。

**(FX) `<array>` に、`resizable` 属性が追加されました。**

解決: レポート未提出。

Cg スコープ内の `<array>` 要素には、配列のサイズ変更を可能にする Boolean である `resizable` 属性が追加されました。

**(コア) `<asset>` の子要素として `<extra>` が追加されました。**

レポート K-617 を解決します (仕様: K-3272)。

`<asset>` 要素は、ゼロまたは任意の数の `<extra>` 要素を含むことができるようになりました。

## (コア) <asset>に地理的範囲の情報が追加されました。

レポート K-2955 を解決します (仕様: K-2863)。

<coverage>/<geographic\_location>要素は、<asset>の中で定義されるようになりました。詳しくは、仕様書の「コア要素のリファレンス」の<asset>要素および<geographic\_location>要素を参照してください。

例：

```
<asset>
  <coverage>
    <geographic_location>
      <longitude>-105.2830</longitude>
      <latitude>40.0170</latitude>
      <altitude mode="relativeToGround">0</altitude>
    </geographic_location>
  </coverage>
  <created>2008-01-28T20:51:36Z</created/>
  <modified>2008-01-28T20:51:36Z</modified/>
</asset>
```

## (フィジックス) <capsule>の<radius>子要素は、値を3つ持つようになりました。

レポート K-3168 を解決します (仕様: K-3169)。

<capsule>/<radius>には、浮動小数点値が(2でなく)3つ含まれるようになりました。

## (コア) <COLLADA>の version 属性は、バージョン 1.5.0 をサポートするようになりました。

解決: レポート未提出。

スキーマのこのバージョンにおいて、<COLLADA>要素の version 属性は、「1.5.0」という値だけをサポートします。COLLADA 1.5.0 の文書は、すべて以下のように現在の COLLADA スキーマを参照する必要があります。

```
<COLLADA xmlns="http://www.collada.org/2008/03/COLLADASchema" version="1.5.0">
```

## (コア) <COLLADA>にライブラリが追加されました。

解決済み: 新機能の統合。

<COLLADA>要素は、以下の要素と関連を持つようになりました。

- <library\_articulated\_systems> (キネマティックス用)
- <library\_formulas>
- <library\_joints> (キネマティックス用)
- <library\_kinematics\_models> (キネマティックス用)
- <library\_kinematics\_scenes> (キネマティックス用)

(FX) `<color_target>`、`<depth_target>`、`<stencil_target>`は、画像を直接参照するようになりました。

解決：レポート未提出。

`<color_target>`、`<depth_target>`、`<stencil_target>`には、以下の子要素のいずれか1つだけが必要になりました。

<code>&lt;param&gt;</code> (リファレンス)	どの画像を使用すべきかを判断するため、samplerパラメータを参照します。
<code>&lt;instance_image&gt;</code>	レンダリング可能な画像を直接インスタンス化します。

(FX) `<connect_param>`は削除されました。

レポート K-3156 を解決します (仕様: K-459)。

`<connect_param>`は、`<array>`、`<usertype>`、またほとんどの`<setparam>`の中でも無効になりました。キネマティックスの`<setparam>`要素および`<instance_formula>`/`<setparam>`の中でだけ有効です。

(コア) `<contributor>`に電子メールやウェブサイトに関する子要素が追加されました。

レポート K-370 を解決します (仕様: K-3117)。

`<contributor>`要素には、`<author_email>`および`<author_website>`という子要素が追加されました。

例：

```
<asset>
<contributor>
  <author>Bob the Artist</author>
  <author_email>bob@bobartist.com</author_email>
  <author_website>http://www.bobartist.com</author_website>
  <authoring_tool>Super3DmodelMaker3000</authoring_tool>
  <comments>This is a big Tank</comments>
  <copyright>Bob's game shack: all rights reserved</copyright>
  <source_data>c:\models\tank.s3d</source_data>
</contributor>
</asset>
```

(FX) 画像テクスチャの初期化をカスタマイズするために、`<create_2d>`、`<create_2d>`、`<create_cube>`が追加されました。

解決：レポート未提出。

これらの要素は、`<image>`の子要素です。基本的な構文は、以下の通りです。

```
<create_2d>
  <size_exact width="..." height="..." /> or
  <size_ratio width="..." height="...">
  <mips levels="..." auto_generate="true" /> or <unnormalized />
  <format ... />
  <init_from ... />
</create_2d>

<create_3d>
  <size width="..." height="..." depth="..." />
  <mips levels="..." auto_generate="true" />
  <array length="..." ... />
```

```

    <format ... />
    <init_from .../>
  </create_3d>

  <create_cube>
    <size width="..." />
    <mips levels="..." auto_generate="false" />
    <array length="..." .../>
    <format ... />
    <init_from... />
  </create_cube>

```

詳細は、[<image>](#)、[<format>](#)、[<init\\_from>](#)の注と仕様書を参照してください。

**(フィジックス) <instance\_rigid\_body>および<rigid\_body>の中の<dynamic>は、デフォルト値を持つようになりました。**

レポート P-51 を解決します (仕様: P-65)。

[<instance\\_rigid\\_body/>](#)[<technique\\_common>](#)および[<rigid\\_body/>](#)[<technique\\_common>](#)の中の[<dynamic>](#)は、デフォルト値 `true` を持つようになりました。

**(コア) 拡張性を高めるため、<evaluate\_scene>および<render>の設計が変更されました。**

レポート K-366 を解決します。

[<evaluate\\_scene>](#)への変更は以下の通りです。

- オプションの新属性 3 つ - `id`、`sid`、`enable`。
- 新子要素 2 つ - [<asset>](#)と[<extra>](#)。
- [<render>](#)子要素は、オプション (0 または 1 回以上出現可能) になりました ([<render>](#)の注も参照)。

[<render>](#)への変更は以下の通りです。

- オプションの新属性 2 つ - `name`、`sid`。
- 新しい子要素 - [<extra>](#)。
- [<instance\\_effect>](#)子要素は、名前を[<instance\\_material>](#)に変更され、再定義されました (注: これは、[<bind\\_material>/<technique\\_common>/<instance\\_material>](#)とは違います)。

[<instance\\_material>](#)への変更 ([<instance\\_effect>](#)と比較して) は以下の通りです。

- [<instance\\_material>](#)は 1 つしか認められないので、`name` 属性や `sid` 属性は削除されました。
- 子要素[<technique\\_hint>](#)が新要素[<technique\\_override>](#)に置き換えられました。これによりエフェクトテクニックやパスを複数のエフェクトに分割しなくても、マテリアルの中の特定のテクニックやパスをターゲットにすることが可能になりました。
- 子要素[<setparam>](#)が[<bind>](#)に置き換えられました。

基本的な構文は、以下の通りです。

```

<evaluate_scene name="..." id="..." sid="..." enable="...">
  <asset ... />
  <render camera_node="..." name="..." sid="..." >
    <layer .../>
    <instance_material url="...">

```

```

    <technique_override ref="..." pass="..." />
    <bind .../>
    <extra .../>
  </instance_material>
  <extra ...>
</render>
<extra ... />
</evaluate_scene>

```

### (コア) <float\_array>の型制約がより厳密になりました。

レポート P-32 を解決します (仕様: P-37)。

`digits` 属性および `magnitude` 属性は、値の範囲が制約されるようになりました。それらの定義を次に示します。

<code>digits</code>	<code>xs:unsignedByte</code>	配列中に含めることが可能な浮動小数点値の有効桁数 (10 進数)。最小値は 1、最大値は 17 です。デフォルト値は 6 です。オプション。
<code>magnitude</code>	<code>xs:short</code>	配列中に含めることが可能な浮動小数点値の最大指数。最大値は 308、最小値は -324 です。デフォルト値は 38 です。オプション。

### (FX) <image>アセットを記述するための<format>が追加されました。

解決: レポート未提出。

この新要素は、<image>アセットに期待されるフォーマットやメモリレイアウトを記述します。この要素は、<image>の中では、<create\_2d>、<create\_3d>、<create\_cube>の子要素になります。<format>の子要素としては、以下の一方または両方が可能です。

```

<hint channels="..." range="..." precision="..." space="..." />
または、
<exact>string-specifying-textel-format</exact>

```

詳しくは仕様書を参照してください。

### (コア) MathML をサポートするために、<formula>、<instance\_formula>、<library\_formulas>が追加されました。

レポート K-2962 を解決します (仕様: K-2963)。

<formula>は、<library\_formulas>、<animation\_clip>、および新しいキネマティクス要素である<kinematics\_model>/<technique\_common>および<kinematics>/<axis\_info>の中に出現することができます。<formula>は、COLLADA の中で関数を定義する機能を提供します。MathML スキーマにアクセスする COLLADA 文書において、<formula>/<technique\_common>には有効な任意の MathML XML を含むことができます。詳しくは仕様書を参照してください。

### (FX) <generator>が削除されました。

レポート K-464、K-636、K-3157 を解決します。

### (FX) <image>が再設計され、<instance\_image>が追加されました。

解決: レポート未提出。

<image>要素には、以下の変更点があります。詳しくは仕様書を参照してください。

- 属性 `format`、`height`、`width`、`depth` は削除されました。

- 属性 `sid` が追加されました。  
`<image id="..." name="..." sid="...">`
- ソース情報用のこれまでの選択肢 (`<data>`、`<init_from>`) が以下に置き換えられました。  
`<init_from>` または `<create_2d>` または `<create_3d>` または `<create_cube>`  
 詳細は、`<init_from>`、`<create_2d>`、`<create_3d>`、`<create_cube>`の注を参照してください。
- 新しい`<renderable>`子要素は、画像がレンダーターゲットであるかどうかを指定します。この要素には、オプションの Boolean 属性 `share` があります。この属性は、全インスタンスの間でレンダーターゲットを複製せずに共有するかどうかを指定します。  
`<renderable share="..." />`
- `<image>`要素が有効なのは、`<library_images>`の子要素としてだけです。他の場所ではすべて無効です。
- 新しい`<instance_image>`要素が有効なのは、`<sampler_*>`、`<color_target>`、`<depth_target>`、`<stencil_target>`の子要素としてです。

基本的な構文は、以下のようになりました。

```
<image id="..." name="..." sid="...">
  <asset .../>
  <renderable share="..." />
  <init_from.../> or <create_2d.../> or <create_3d.../> or <create_cube.../>
  <extra .../>
</image>
```

**(FX) 画像の初期化をよりうまく制御できるようにするために、`<init_from>`の設計が変更されました。**

*解決: レポート未提出。*

`<init_from>`は、`<image>`、`<create_2d>`、`<create_3d>`、`<create_cube>`の子要素になれるようになりました。利用できる属性子要素は、親によって異なりますが、基本的な構文は以下の通りです。

```
<image>
  <init_from mips_generate="...">
    <ref> url </ref>
  または、
    <hex format="...">embedded_image_data</hex>
  </init_from>
</image>

または、

<create_*>
  <init_from array_index="..." mip_index="..." depth="..." face="...">
    <ref> url </ref>
  または、
    <hex format="...">embedded_image_data</hex>
  </init_from>
</create_*>
```

詳細は、`<image>`、`<create_*>`の注と仕様書を参照してください。



(コア) `<instance_node>`には、複数の表現や詳細レベル (LOD) をサポートするために、`proxy` 属性が追加されました。

レポート K-3166 を解決します (仕様: K-2974)。

`proxy` 属性は、オプションの `xs:anyURI` 型属性です。アプリケーションは、`url` 属性の URL と、`proxy` 属性の URL のどちらを解決するかを決定することができます。どちらも `<node>` 要素に解決されます。アプリケーションが追跡するパスを決定する方法に関する情報は、`<instance_node>` の `<extra>` 要素に格納する必要があります。詳しくは仕様書を参照してください。

(FX) `<pass>` は設計が変更され、新しく子要素 `<states>` と `<evaluate>` が追加されました。

解決: レポート未提出。

`<pass>` の新しい基本構文は、以下の通りです。

```
<pass sid="..." >
  <annotate />
  <states>
    render_states_list
  </states>
  <program .../> <!-- CG, GLES2, GLSL スコープ内のみ -->
  <evaluate>
    <color_target .../>
    <depth_target .../>
    <stencil_target .../>
    <color_clear .../>
    <depth_clear .../>
    <stencil_clear .../>
    <draw .../>
    <extra .../>
  </pass>
```

相違点は、以下の通りです。

- `<pass>` のすぐ下にあった子要素のほとんどは、新しい子要素 `<states>`、`<evaluate>` の下に移動されました。
- これまでの `<shader>` 子要素は、`<program>` 要素によって置き換えられました。詳しくは、`<program>` の注を参照してください。

(FX) `<profile_*>` および `</profile_*>` / `<technique>` 子要素は削除されました。

解決: レポート未提出。

以下の子要素は、ほとんどの場合、`<profile_*>` 要素のすぐ下に出現することはなくなりました。

- `<image>`、`<newparam>`、`<setparam>`、`<code>`、`<include>`

以下の子要素は、`<profile_*>` / `<technique>` 要素のすぐ下に出現することはなくなりました。

- `<array>`、`cg_value_type`、`<connect_param>`、`<usertype>`

(FX) `<profile_BRIDGE>` が追加されました。

解決: レポート未提出。

このプロファイルは、他の `<profile_*>` 要素では扱われていない外部規格によって書き込まれたエフェクトプロファイルの参照に、サポートを提供します。`<effect>` の下で有効です。基本的な構文は、以下の通りです。

```

<profile_BRIDGE id="..." platform="..." url="...">
  <asset .../>
  <extra .../>
</profile_BRIDGE>

```

### (FX) <profile\_GLES2>が追加されました。

解決: レポート未提出。

<effect>の下で有効です。基本的な構文は、以下の通りです。

```

<profile_GLES2 id="..." language="..." platforms="...">
  <asset .../>
  <code .../>
  <include .../>
  <newparam .../>
<technique>
  <asset .../>
  <annotate .../>
  <pass .../>
  <extra .../>
</technique>
  <extra .../>
</profile_GLES2>

```

### (FX) <profile\_\*>要素はすべて、id 属性を持つようになりました。

レポート K-322、P-28 を解決します (仕様: K-329)。

### (FX) <profile\_GLSL>に、platform 属性が追加されました。

レポート K-322 を解決します (仕様: K-329)。

これにより、既存の<profile\_\*>要素の機能に対応するようになりました。

### (FX) シェーダの定義を改善するために、<program>要素が新しい子要素<compiler>、<linker>、<binary>、<bind\_attribute>などを伴って追加されました。

解決: レポート未提出。

新しい<program>要素は、<pass>の子要素で、複数のシェーダをリンクして、ジオメトリ処理用のパイプラインを生成する機能を提供します。有効な要素はプロファイルによって異なりますが、可能な要素の一覧は以下の通りです。

```

<program>
  <shader stage="...">
    <sources .../>
    <compiler platform="..." target="..." options="...">
      <binary .../>
    </compiler>
    <bind_uniform .../>
    <extra .../>
  </shader>
  <linker platform="..." target="..." options="...">
    <binary .../>
  </linker>
  <bind_attribute symbol="...">
    <semantic .../>
  </bind_attribute>

```

```
<bind_uniform .../>
</program>
```

**<bind\_attribute>**は、シェーダの頂点属性入力に、セマンティクスをバインドします。詳しくは仕様書を参照してください。

また、**<shader>**の注も参照してください。

## (FX) <render>が更新されました。

解決: レポート未提出。

**<render>**要素への変更は以下の通り。

- **name** 属性、および **sid** 属性を追加しました。
- 子要素として**<extra>**が利用できるようになりました。
- **<instance\_effect>**は子要素として有効ではなくなりました。

## (フィジックス) <rigid\_body>に id 属性が追加されました。

レポート K-1820 を解決します (仕様: K-1821)。

## (コア) <sampler>に、最初のキーの前、および最後のキーの後のサンプル値を指定するための属性が追加されました。

レポート K-2973 を解決します (仕様: K-3171)。

**<sampler>**に、最初のキーの前、および最後のキーの後のサンプル値の、あるべき挙動を表す属性 **pre\_behavior** と **post\_behavior** が追加されました。詳しくは仕様書を参照してください。

例 :

```
<sampler pre_behavior="CONSTANT" post_behavior="GRADIENT">
```

## (FX) <sampler\*>子要素の設計が変更されました。

解決: レポート未提出。

子要素**<sampler1D>**、**<sampler2D>**、**<sampler3D>**、**<samplerCUBE>**、**<samplerDEPTH>**、**<samplerRECT>**に対して、以下の変更が行われました。

- サンプラが消費するデフォルト画像をインスタンス化する**<instance\_image>**が追加されました。
- **<mipmap\_maxlevel>**の名前が**<mip\_max\_level>**に変更されました。
- **<mipmap\_bias>**の名前が**<mip\_bias>**に変更されました。
- 評価を開始する最小プログレッシブレベルを表す**<mip\_min\_level>**が追加されました。
- 異方性フィルタリングの際に利用できるサンプル数を表す**<max\_anisotropy>**が追加されました。
- 子要素のほとんどは、どの**<sampler\*>**要素の中でも有効になりました。ただし、要素によっては unnecessary なものもあります。
- **<wrap\_\*>**の NONE モードは、削除されました。そして、テクスチャ座標の絶対値をとって、最大値を超える部分を切り捨てる MIRROR\_ONCE モードが追加されました。
- **<\*filter>** (**<minfilter>**、**<magfilter>**、**<mipfilter>**)、NEAREST\_MIPMAP\_NEAREST、LINEAR\_MIPMAP\_NEAREST、NEAREST\_MIPMAP\_LINEAR、LINEAR\_MIPMAP\_LINEAR が削除されました。

- ポリゴンと画面平面の間との角度の違いに起因する歪みを補償するために、`<*filter>`値 ANISOTROPIC が追加されました (`<minfilter>`専用)。
- `<*filter>`値 NONE が有効なのは、`<mipfilter>`だけになりました。

### (FX) `<sampler_image>`が追加されました。

解決: レポート未提出。

サンプラの対象となる画像をインスタンス化します。これはサンプラの型ではなく、むしろ既存のサンプリングを変更するのに使われる要素です。親の`<setparam>`によって特定されるサンプリング`<newparam>`は、インスタンス化された画像を受信します。この型は、`<instance_image>`の派生型で、具体的な拡張はされていませんが、このような状況を明確にするために名前が変更されています。

### (FX) `<sampler_state>`は、`<sampler_states>`として設計が変更されました。

解決: レポート未提出。

変更点は以下の通りです。

- `<newparam>`から`<sampler_state>`が削除されました。
- `<sampler_state>`は、`<setparam>`の下の`<sampler_states>`になりました。子要素としては、再設計された`<sampler*>`要素と同じものを利用できます (`<instance_image>`を除く)。詳細は`<sampler*>`の注を参照してください。

### (FX) いくつかの要素から`<setparam>`が削除されました。

解決: レポート未提出。

`<setparam>`は、`<effect>`、`<profile_*>`要素のいずれか、また`<profile_*>/<technique>`の下では無効になりました。

### (FX) `<shader>`の設計が変更されました。

解決: レポート未提出。

`<shader>`要素には、以下の変更点があります。詳しくは仕様書を参照してください。

- `stage` 属性の値として有効なのは、FRAGMENT、GEOMETRY、TESSELLATION、VERTEX になりました。これらはすべて、`<shader>`が出現するいかなるコンテキストでも有効です。FRAGMENTPROGRAM と VERTEXPROGRAM は、削除されました。
- `<annotate>`子要素は削除されました。
- `<compiler_target>`および`<compiler_options>`子要素は、削除されて、`<sources>`および`<compiler>`に置き換えられました。詳しくは、`<sources>`および`<program>`の注を参照してください。
- `<name>`子要素は削除されました。代わりとして、`<sources>`要素の `entry` 属性を利用してください。
- `<bind>`子要素の名前は`<bind_uniform>`に変更されました。
- 子要素として`<extra>`が有効になりました。

基本的な構文は、以下のようになりました。

```
<shader stage="VERTEX">
  <sources entry="main">
<inline .../>
```

```

    <import ref="..." />
    </sources>
    <compiler platform="..." target = "..." />
    <bind_uniform symbol="...">
      <param ref="..." />
    </bind_uniform>
    <extra ... />
  </shader>

```

## (コア) <SIDREF\_array>が追加されました。

レポート K-3282 の解決 (仕様: K-3283)。

<SIDREF\_array>要素は、<source>の子要素になりました。この要素には、`sidref_type` 型の値が 1 つ以上含まれており、`id`、`name`、`count` という標準の配列属性を保持しています。

例：

```

<SIDREF_array id="SIDREFlist" name="My SIDREF list" count="3">
  item1 item2 item3
</SIDREF_array>

```

## (コア、フィジックス) <source>では、トークンや SIDREF の配列を利用できるようになりました。

レポート K-3282 を解決します (仕様: K-3283)。

<source>の子要素として、<SIDREF\_array>や<token\_array>を利用できるようになりました。詳しくは<SIDREF\_array>および<token\_array>の注を参照してください。

## (FX) シェーダソースコードを結合するために、<sources> が追加されました。

解決: レポート未提出。

<shader>/<sources>では、インラインコード (<inline>) とインクルードファイル (<code>や<include>要素を参照する<import>) と組み合わせることにより、シェーダコード用のソースを複数指定することができます。<sources>の基本的な構文は以下の通りです。

```

    <sources entry="main">
    <inline .../>
    <import ref="..." />
    </sources>

```

また、<shader>の注も参照してください。

## (FX) <surface>が削除されました。

レポート K-338 の解決。

## (コア) <tapered\_capsule>および<tapered\_cylinder>が削除されました。

レポート K-3170 を解決します (仕様: K-3202)。

**(コア) <technique>は、COLLADA 要素名との衝突を回避できるようになりました。**

レポート K-213 を解決します。

<technique>子要素は、バリデーションの際に COLLADA の要素名と衝突する危険を冒すことなく、<COLLADA>以外の任意の要素名を安全に利用できるようになりました。

**(FX) <texenv>の unit 属性の名前と型が変更されました。**

解決: レポート未提出。

<texenv>の unit 属性は、名前が sampler に変更され、型もサンブラを指す sidref\_type 型になりました。

**(FX) <texture>の<extra>は、maxOccurs=unbounded を使うようになりました。**

レポート K-448 を解決します (仕様: K-2890)。

<texture>/<extra>は、1 回以上出現することが可能になりました。

**(FX) <texture\_pipeline>が出現できるのは、<states>の中だけになりました。**

解決: レポート未提出。

<texture\_pipeline>を、<newparam>として生成することはできなくなりました。この要素は、レンダーステートとしてしか利用できません。つまり、<pass>、<newparam>、<setparam>の下では有効ではなくなりました。

**(FX) <texture\_unit>が削除されました。**

解決: レポート未提出。

<texture\_unit>は存在しなくなりました。その主な 3 つの子要素は以下です。

- <surface>は存在しなくなりました。<surface>の注を参照してください。
- <texcoord>は、<newparam>/<sampler2D>の下の GLES スコープでだけ有効になりました。
- <sampler\_state>は、パラメータの参照でした。

**(コア) <token\_array>が追加されました。**

レポート K-3282 を解決します。

<token\_array>要素は、<source>の子要素になりました。この要素には、xs:token 型の値が 1 つ以上含まれており、id、name、count という標準の配列属性を保持しています。例:

```
<token_array id="tokenlist" name="My token list" count="3">
  item1 item2 item3
</token_array>
```

**(FX) <transparent>は、追加の不透明モードをサポートするようになりました。**

レポート K-622 を解決します (仕様: K-3118)。

<transparent>要素の opaque 属性として、A\_ONE、RGB\_ZERO に加えて、以下の値が可能になりました。

- **A\_ZERO** (デフォルト)：透明度情報はカラーのアルファチャンネルから取得します。ここで値 0.0 は不透明を表します。
- **RGB\_ONE**：各チャンネルが独立して調節される、カラーの R (赤)、G (緑)、B (青) チャンネルから透明度情報を取り出します。ここで値 1.0 は不透明を表します。

## (FX) <usertype>の設計が変更されました。

解決：レポート未提出。

以下の変更が行われました。

- **name** 属性の名前は、**typename** に変更されました。
- 有効な子要素は、<setparam>だけになりました。他の子要素はすべて削除されました。

---

## 1.4.1 (第2版) 以降の仕様変更

### (全般) テキストにさまざまな修正を行いました。

レポート未提出。

- 1.5.0 スキーマと一致するようにテキストを更新 (前セクションを参照)。
- 1.4.1 仕様書 (第2版) のバグや脱落を修正 (このセクションで詳述)。
- テキスト、特に「コア要素のリファレンス」章の<controller>の部分を、より明快にしました。
- 「その他の情報源」のリストを拡大して、リンクを最新に訂正しました。
- COLLADA の歴史の章から、「開発方法論」のセクションを削除しました。
- 「スキーマのコンセプト」章の中の、「アドレス構文」のセクションを更新し、拡大しました。
- 「スキーマのコンセプト」章の中の、「テクニックの例と議論」セクションを拡張しました。
- 「プログラミングガイド」の章に「パラメータ」の議論を追加しました。
- GLSL サンプルの付録を更新して、付録に GLES2 のサンプルを追加しました。
- 該当する機能によりふさわしい記述にするため、「コア要素のリファレンス」の章の「<param> (コア)」のラベルを「<param> (データフロー)」に、「<input> (インデックス付き)」というラベルを「<input> (共有)」に、「<input> (インデックスなし)」というラベルを「<input> (非共有)」に、それぞれ変更しました
- 「フィジックスリファレンス」章の概要を書き直して、フィジックス要素の多く、特に、<morph>、<physics\_model>に関する文章をより明快にして拡大しました。
- <setparam>、<newparam>、<param> (参照) 要素が、FX 以外の場所でも使われるようになったことを反映して、これらの要素を「コア要素のリファレンス」章に移動しました。

### (索引) 「COLLADA 要素の索引」に主見出し項目のない要素が追加されました。

レポート P-18、P-19 を解決します。

完全な参照項目があるかどうかにかかわらず、全要素 (<h>、<v>など) が索引に追加されました。

**(フィジックス) <convex\_mesh>に出現する子要素の説明が訂正されました。**

レポート K-3155 を解決します。

<convex\_mesh>の「子要素」セクションを訂正して、この要素に子要素を指定する必要はないが、どれか一つでも指定した場合には、<source>は1回以上出現できるが、<vertices>はちょうど1回しか出現できないという説明にしました。

**(FX) <color\_target>属性の型が訂正されました。**

解決: レポート未提出。

<color\_target>の face 属性が、列挙型として正しく文書化されました。

**(コア) <evaluate\_scene>が文書化されました。**

レポート K-366 を解決します。

<visual\_scene>/<evaluate\_scene>が文書化されました。

**(コア) <morph>属性の名前が、sid から source に修正されました。**

レポート K-3352 を解決します。

<morph source=... >が正しく文書化されました。

**(コア) <orthographic>子要素が訂正されました。**

解決: レポート未提出。

誤って<orthographic>の子要素として記載されていた<extra>が仕様書から削除されました。

**(コア) <point>子要素が訂正されました。**

解決: レポート未提出。

誤って<point>の子要素として記載されていた<zfar>が仕様書から削除されました。

---

## 仕様 1.5.0 に残っている問題点

**(本ドキュメント) 「その他の情報源」の訂正。**

未解決レポート: K-3500。

「Softimage dotXSI FTK」を「Softimage Crosswalk SDK」に訂正する必要があります。

**(コア) <token\_array>が文書化されていません。**

未解決レポート: K-3473。

前述の「バージョン 1.4.1 以降のスキーマの変更」セクションの<token\_array>の説明を参照してください。



**(FX) `fx_sampler_type` 型が誤って `fx_sampler_common` として参照されています。**

未解決レポート: *K-3490*。

この型は、仕様書中で便宜的に使われているだけです。COLLADA 文書やアプリケーション が、この型名を明示的に使うことは想定されていません。

**(コア、FX) 「パラメータの型要素」セクションへの参照は第 11 章を参照する必要があります。**

未解決レポート: *K-3491*。

「パラメータの型要素」セクションへの相互参照は第 11 章を参照している必要がありますが、現在は、以下の 3 箇所です「章末で」と記述されています。

- `<array>`
- `<bind_uniform>`
- `<setparam>`