

NINTENDO TWL-System NITRO 中間ファイルフォーマット

Ver 1.6.0

(2009/02/18 版)

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、
厳重な取り扱い、管理を行ってください。

目次

1	はじめに	9
2	中間ファイル	10
2.1	中間ファイルとは	10
2.2	中間ファイルの種類	11
3	中間ファイルの記述規則	12
3.1	中間ファイルの構造	12
3.1.1	XML宣言	12
3.1.2	中間ファイルの要素構成	12
3.2	要素 (Element) について	12
3.2.1	要素の命名規則	12
3.2.2	その他の規則	13
3.3	属性 (Attribute) について	13
3.3.1	属性の命名規則	13
3.3.2	その他の規則	14
3.4	共通ヘッダ部の要素及び属性説明	14
3.4.1	ルート要素	14
3.4.2	<head>要素	14
3.4.3	<create>要素	15
3.4.4	<title>要素	15
3.4.5	<comment>要素	16
3.4.6	<generator>要素	16
3.4.7	<body>要素	17
3.5	その他の規則	17
4	モデルデータ用中間ファイル	18
4.1	モデルデータファイル (imd)	18
4.1.1	imdの概要	18
4.1.2	imdの基本的な要素構成	19
4.1.3	imdの要素及び属性説明	20
4.1.4	imdの補足	49
5	アニメーション用中間ファイル	56
5.1	アニメーションファイルのデータ形式	56
5.1.1	データ形式の種類	56
5.1.2	コマ形式	56
5.1.3	FVキー形式	59
5.2	キャラクターアニメーションファイル (ica)	61
5.2.1	icaの概要	61
5.2.2	icaの基本的な要素構成	61

5.2.3	icaの要素及び属性説明	61
5.2.4	icaの補足	67
5.3	ビジビリティアニメーションファイル (iva)	68
5.3.1	ivaの概要	68
5.3.2	ivaの基本的な要素構成	68
5.3.3	ivaの要素及び属性説明	68
5.3.4	ivaの補足	72
5.4	マテリアルカラーアニメーションファイル (ima)	73
5.4.1	imaの概要	73
5.4.2	imaの基本的な要素構成	73
5.4.3	imaの要素及び属性説明	73
5.4.4	imaの補足	80
5.5	テクスチャパターンアニメーションファイル (itp)	81
5.5.1	itpの概要	81
5.5.2	itpの基本的な要素構成	81
5.5.3	itpの要素及び属性説明	81
5.5.4	itpの補足	87
5.6	テクスチャSRTアニメーションファイル (ita)	88
5.6.1	itaの概要	88
5.6.2	itaの基本的な要素構成	88
5.6.3	itaの要素及び属性説明	88
5.6.4	itaの補足	94
6	グラフィックスシーン用中間ファイル	95
6.1	グラフィックスシーンデータファイル (isd)	95
6.1.1	isdの概要	95
6.1.2	isdの基本的な要素構成	95
6.1.3	isdの要素及び属性説明	96
6.1.4	isdの補足	104
7	データの最適化	105
7.1	ノード削減	105
7.2	マテリアル圧縮	105
7.3	ポリゴンの連結化	106
8	参考情報	107
8.1	行列計算について	107
8.1.1	ノードの行列の計算方法	107
8.1.2	テクスチャ行列の計算方法	107

コード

コード 1	コマ形式再生関数のサンプル (Cソース形式)	58
コード 2	FVキー形式関数のサンプル (Cソース形式)	60

表

表 2-1	中間ファイルの種類	11
表 3-1	ルート要素の属性	14
表 3-2	<create>要素の属性	15
表 3-3	<generator>要素の属性	16
表 4-1	<original_create>要素の属性	20
表 4-2	<original_generator>要素の属性	20
表 4-3	<model_info>要素の属性	21
表 4-4	<box_test>要素の属性	23
表 4-5	<vtx_pos_data>要素の属性	23
表 4-6	<vtx_color_data>要素の属性	24
表 4-7	<tex_image_array>要素の属性	24
表 4-8	<tex_image>要素の属性	25
表 4-9	<bitmap>要素の属性	26
表 4-10	<tex4x4_palette_idx>要素の属性	27
表 4-11	<tex_palette_array>要素の属性	27
表 4-12	<tex_palette>要素の属性	28
表 4-13	<material_array>要素の属性	28
表 4-14	<material>要素の属性	29
表 4-15	<weight>要素の属性	32
表 4-16	<node_idx>要素の属性	32
表 4-17	<matrix_array>要素の属性	33
表 4-18	<matrix>要素の属性	33
表 4-19	<polygon_array>要素の属性	34
表 4-20	<polygon>要素の属性	35
表 4-21	<mtx_prim>要素の属性	35
表 4-22	<mtx_list>要素の属性	36
表 4-23	<primitive_array>要素の属性	37
表 4-24	<primitive>要素の属性	38
表 4-25	<mtx>要素の属性	39
表 4-26	<tex>要素の属性	39
表 4-27	<nrm>要素の属性	40
表 4-28	<clr>要素の属性	40
表 4-29	<clr_idx>要素の属性	41
表 4-30	<pos_s>要素の属性	41
表 4-31	<pos_xy>要素の属性	42
表 4-32	<pos_xz>要素の属性	42
表 4-33	<pos_yz>要素の属性	43
表 4-34	<pos_xyz>要素の属性	43
表 4-35	<pos_diff>要素の属性	44
表 4-36	<pos_idx>要素の属性	44
表 4-37	<node_array>要素の属性	45
表 4-38	<node>要素の属性	45
表 4-39	<display>要素の属性	47
表 4-40	<output_info>要素の属性	47

表 5-1 アニメーションのデータ形式	56
表 5-2 コマ形式で出力されるフレーム	57
表 5-3 3DCGツール上のデータ と FVキー・データ.....	59
表 5-4 <node_anm_info>要素の属性	62
表 5-5 <node_scale_data>要素の属性	63
表 5-6 <node_rotate_data>要素の属性	63
表 5-7 <node_translate_data>要素の属性.....	64
表 5-8 <node_anm_array>要素の属性	64
表 5-9 <node_anm>要素の属性	65
表 5-10 <scale_x>,<scale_y>,<scale_z>要素の属性	65
表 5-11 <rotate_x>,<rotate_y>,<rotate_z>要素の属性.....	66
表 5-12 <translate_x>,<translate_y>,<translate_z>要素の属性	67
表 5-13 <visibility_info>要素の属性	68
表 5-14 <frame_idx>要素の属性	70
表 5-15 <visibility>要素の属性.....	70
表 5-16 <visibility_anm_array>要素の属性.....	71
表 5-17 <visibility_anm>要素の属性	71
表 5-18 <mat_color_info>要素の属性.....	74
表 5-19 <mat_color_data>要素の属性.....	75
表 5-20 <mat_color_anm_array>要素の属性	75
表 5-21 <mat_color_anm>要素の属性.....	76
表 5-22 <diffuse_r>,<diffuse_g>,<diffuse_b>要素の属性	77
表 5-23 <ambient_r>,<ambient_g>,<ambient_b>要素の属性.....	77
表 5-24 <specular_r>,<specular_g>,<specular_b>要素の属性	78
表 5-25 <emission_r>,<emission_g>,<emission_b>要素の属性	79
表 5-26 <polygon_alpha>要素の属性	79
表 5-27 <tex_pattern_info>要素の属性.....	82
表 5-28 <tex_pattern_list_data>要素の属性	82
表 5-29 <image_name>要素の属性	83
表 5-30 <palette_name>要素の属性.....	83
表 5-31 <frame_idx>要素の属性	84
表 5-32 <image_idx>要素の属性.....	85
表 5-33 <palette_idx>要素の属性.....	85
表 5-34 <tex_pattern_anm_array>要素の属性	86
表 5-35 <tex_pattern_anm>要素の属性	86
表 5-36 <tex_srt_info>要素の属性.....	89
表 5-37 <tex_scale_data>要素の属性	90
表 5-38 <tex_rotate_data>要素の属性	90
表 5-39 <tex_translate_data>要素の属性.....	91
表 5-40 <tex_srt_anm_array>要素の属性	91
表 5-41 <tex_srt_anm>要素の属性	92
表 5-42 <tex_scale_s>,<tex_scale_t>要素の属性	92
表 5-43 <tex_rotate>要素の属性	93
表 5-44 <tex_translate_s>,<tex_translate_t>要素の属性	93
表 6-1 <camera>要素の属性	96

表 6-2	<light0>,<light1>,<light2>,<light3>要素の属性.....	98
表 6-3	<toon_highlight>要素の属性	99
表 6-4	<fog>要素の属性.....	100
表 6-5	<alpha_test>要素の属性.....	101
表 6-6	<alpha_blending>要素の属性	101
表 6-7	<y_sorting>要素の属性.....	101
表 6-8	<edge_marking>要素の属性.....	102
表 6-9	<antialias>要素の属性	103
表 6-10	<render_1_pixel_depth>要素の属性.....	103
表 6-11	<clear_color>要素の属性	103
表 7-1	ノード削減の種類	105



図 4-1	box_testの設定値.....	23
図 4-2	フルウェイトエンベロープモデルの例	50
図 4-3	ウェイトエンベロープモデルの例	50
図 4-4	ポリゴンの描画優先度の例	52
図 6-1	<camera>要素の属性	97

改訂履歴

版	改訂日	改訂内容
1.6.0	2009-02-18	・商標表記を修正。
1.6.0	2008-05-30	・NITRO-System の名称変更による修正 (NITRO-System を TWL-System に変更)。
1.6.0	2008-04-08	・改訂履歴の書式を変更。
1.6.0	2007-05-18	・imd の<box_test>の属性の説明を修正 (P.23)。
1.6.0	2005-02-02	<ul style="list-style-type: none"> ・imd に属性を追加/削除。 <ul style="list-style-type: none"> ・<material>要素に属性 tex_effect_mtx を追加 (P.31)。 ・<material>要素の属性 tex_gen_st を削除 (P.53)。 ・imd の説明を加筆。 <ul style="list-style-type: none"> ・<model_info>要素の属性 tool_start_frame の説明を加筆 (P.21)。 ・<material>要素の属性 tex_gen_mode, tex_gen_st_src, の説明を修正、加筆 (P.30～31)。 ・<tex>要素の出力条件を修正 (P.38)。 ・「4.1.4 imd の補足」に”環境/投影マッピングについて”の説明を追加 (P.53)。 ・「6.1.4 isd の補足」の説明を修正 (P.102)。
1.5.0	2004-12-13	<ul style="list-style-type: none"> ・imd に要素,属性を追加 <ul style="list-style-type: none"> ・<original_create>, <original_generator>要素を追加 (P.19～20)。 ・<material>要素に属性 depth_test_decals, render_1_pixel, far_clipping, translucent_update_depth を追加 (P.29～30)。 ・<display>要素に属性 priority を追加、描画優先度に関する説明を加筆 (P.45～46、P.51～52、P.102)。 ・isd に要素,属性を追加 <ul style="list-style-type: none"> ・<y_sorting>要素を追加 (P.92、P.98)。 ・<render_1_pixel_depth>要素を追加 (P.92、P.100)。 ・<camera>要素に属性 depth_buffer, scale_w を追加 (P.93)。 ・「5.2.4 ica の補足」の”G3D ライブラリで再生する際の注意”を修正 (P.64)。
1.4.2	2004-10-25	<ul style="list-style-type: none"> ・「表 4-12 <material>要素の属性」の light0 の説明を修正 (P.27)。 ・ica,ima,ita の属性 interpolation の説明を修正、補足を加筆 (P.54、P.59、P.66、P.72、P.81、P.86)。
1.4.2	2004-08-30	<ul style="list-style-type: none"> ・imd の<node>要素の属性 scale_compensate の出力条件を変更 (P.42)。 ・「NINTENDO NITRO-System でのエンベロープ表記について」の説明を追加 (P.46)。 ・「頂点位置座標を変換する行列について」に法線ベクトルに関する説明を加筆 (P.47)。
1.4.1	2004-08-02	<ul style="list-style-type: none"> ・NITRO 中間ファイルプラグイン for SOFTIMAGE XSI、NITRO 中間ファイルプラグイン for 3ds max 対応に伴い、加筆 (P.9、P.98)。 ・imd,ica の<***_info>要素の属性 scaling_rule の説明を修正 (P.19、P.53)。 ・imd,ita の<***_info>要素の属性 tex_matrix_mode のとる値に xsi,3dsmax を追加 (P.19～20、P.80)。 ・imd の<polygon>要素の属性 name の説明を加筆 (P.31)。

1.4.0	2004-06-30	<ul style="list-style-type: none"> •imd の<polygon>要素の属性 name の説明を修正 (P.31)。 •isd の<camera>要素の属性 twist の説明を修正 (P.88)。 •「6.1.4 isd の補足」に加筆 (P.95)。
1.3.0	2004-05-24	<ul style="list-style-type: none"> •ica の<node_anm_info>要素に属性 scaling_rule を追加 (P.53)。 •「5.5.3.4 <palette_name>要素」の出力条件を訂正 (P.74)。 •「6.1.3.1 <camera>要素」の説明に加筆、図を追加 (P.88)。
1.2.0	2004-04-12	<ul style="list-style-type: none"> •グラフィックスシーン用中間ファイル (isd) を追加 (P.10、P.86～93)。 •「表 3-2 <create>要素の属性」の属性 source の出力条件を変更 (P.14)。 •「3.4.5 <comment>」要素の説明を修正 (P.15)。 •imd の<material>要素の属性 polygon_mode のとる値、toon_hilight を toon_highlight に修正 (P.27)。 •3Dマテリアルエディタ対応に伴い修正、加筆。 <ul style="list-style-type: none"> ・説明文を修正 (P.9)。 ・imd に3Dマテリアルエディタ専用の要素及び属性を追加 (P.18、P.43～44)。
1.1.1	2004-03-25	<ul style="list-style-type: none"> •中間ファイル出力プラグイン for SOFTIMAGE 3D 対応に伴い、加筆 (P.8、P.85)。 •「表 6-1 ノード削減の種類」に加筆 (P.83)。
1.1.0	2004-03-01	<ul style="list-style-type: none"> •imd の<model_info>要素に属性 use_primitive_strip を追加 (P.18)。 •A3I5 半透明テクスチャに対応 (P.21、P.42)。 •imd の<tex_palette>要素の属性 color_size がとる値から 2 を削除 (P.23)。 •マニュアル内で NINTENDO NITRO-System ライブラリを使用する場合の名前の制限が 15 文字以内になっていたのを 16 文字以内に修正 (P.42)。 •「6.3 ポリゴンの連結化」の説明文を修正 (P.84)。 •「7.1.1 ノードの行列の計算方法」の説明文を修正 (P.85)。
1.0.0	2004-01-23	リリース

1 はじめに

本マニュアルは、TWL およびニンテンドーDS で3Dモデルを表示するためのデータファイルである ”NITRO 中間ファイル(以下、中間ファイル)” の内容を把握して、独自のコンバータや表示ルーチンを作成する人向けのマニュアルです。

中間ファイルはXML形式を採用しているため、このマニュアルの内容を正しく把握するにはXMLの知識が必要です。

2 中間ファイル

2.1 中間ファイルとは

中間ファイルは、3Dモデルデータやそのアニメーションデータを TWL およびニンテンドーDS に適したデータ構造で格納するXML形式のテキストファイルです。

中間ファイルは、3DCGツールから **NITRO 中間ファイルプラグイン**を使ってファイル出力することができます。

出力した中間ファイルは、**NINTENDO TWL-System 3Dマテリアルエディタ**(以下、3Dマテリアルエディタ)でマテリアル設定を編集したり、**NINTENDO TWL-System ライブラリ** を使って TWL やニンテンドーDS 上に表示するのに利用できます。 また、中間ファイルの内容を解析して独自の表示ルーチンを構築することもできます。

TWL-System では現在、以下の3DCGツールからの中間ファイル出力をサポートしています。

- NITRO 中間ファイルプラグイン for Maya
- NITRO 中間ファイルプラグイン for SOFTIMAGE|XSI
- NITRO 中間ファイルプラグイン for 3ds max

詳細については、各プラグインマニュアルを参照ください。

※NITRO-System でサポートされていた SOFTIMAGE|3D は、TWL-System ではサポートしていません。

中間ファイルフォーマットのバージョンは、 Ver 1.6.0 です。

2.2 中間ファイルの種類

NITRO 中間ファイルは機能別に7種類あり、それぞれファイルの拡張子名で区別します。

3Dモデルを表示するために必要なデータは imd ファイル内に格納されます。

この imd ファイルをベースに、ポーズを変えて動かしたり(=キャラクタアニメーション)、ノード単位で表示非表示を切り替えたり(=ビジビリティアニメーション)、マテリアルの色を変えたり(=マテリアルカラーアニメーション)、テクスチャを差し替えたり(=テクスチャパターンアニメーション)、テクスチャをスクロール表示したり(=テクスチャSRTアニメーション)といった表現を各アニメーション用中間ファイルを使って行うことができます。

3Dモデルを表示する上で、imd ファイルには格納されないカメラやライト・トゥーンテーブルなど主にフレーム単位で設定するシーンの情報については、isd ファイルに格納されます。

表 2-1 中間ファイルの種類

拡張子	中間ファイルの種類	ファイル内容
imd	モデルデータ	ポリゴン、親子構造、マテリアル、テクスチャを含むモデル情報
ica	キャラクタアニメーションデータ	ノードの行列を操作するアニメーション情報
iva	ビジビリティアニメーションデータ	ノードのビジビリティを操作するアニメーション情報
ima	マテリアルカラーアニメーションデータ	マテリアルの色を操作するアニメーション情報
itp	テクスチャパターンアニメーションデータ	複数のテクスチャを差し替えるアニメーション情報
ita	テクスチャSRTアニメーションデータ	テクスチャ行列を操作するアニメーション情報
isd	グラフィックスシーンデータ	カメラやライト、トゥーンテーブルなどシーンの情報

現在、isd ファイルは3Dマテリアルエディタ からのみ出力可能です。

3 中間ファイルの記述規則

3.1 中間ファイルの構造

3.1.1 XML宣言

XML文書であることを示すため、ファイルの1行目に必ず以下のXML宣言が記述されます。

```
<?xml version="1.0" encoding="Shift_JIS"?>
```

3.1.2 中間ファイルの要素構成

全ての中間ファイルは、拡張子と同じ名前の要素をルート要素とし、その下にヘッダ部(<head>要素)とデータ部(<body>要素)の2つの要素が格納されます。<head>要素は全中間ファイル共通のフォーマットでドキュメントそのものに関する情報が格納されます。<body>要素は各中間ファイルごとに異なる要素構成となります。

例) imd の場合

<?xml version="1.0" encoding="Shift_JIS"?>	…XML宣言
<imd version="1.0.0">	…ルート要素
<head>	…ヘッダ部
… (省略: 全ての中間ファイルに共通) …	
</head>	
<body>	…データ部
… (省略: 中間ファイルの種類ごとに異なる) …	
</body>	
</imd>	

3.2 要素(Element)について

3.2.1 要素の命名規則

- 要素名は、全てアルファベット小文字の英単語と '_' (アンダーバー) を組み合わせてつけられます。ローマ字表記やアルファベット大文字は含みません。
- 要素によっては以下のような略語を用いる場合があります(アルファベット順)。

animation	→	anm
color	→	clr
difference	→	diff
index	→	idx
information	→	info
matrix	→	mtx
normal	→	nrm

position → pos
primitive → prim
scale rotate translate → srt
texture → tex
vertex → vtx

- <XXXX_array>のような名前の要素は、必ず **size** という属性を持ち、子要素として<XXXX>要素を **size** 個格納します。
- 全ての中間ファイルで基本的に要素名は重複しません。ほとんどの要素は名前でのどのファイルのどの位置に出力されるかが決まります。例外として、アニメーション用中間ファイルに共通して使われる要素がいくつか存在しますが、属性及び要素内容の形式は統一されています。

3.2.2 その他の規則

- 中間ファイルでは、データが空になる不要な要素は出力しません。ある特定の条件でのみ出力される要素については各要素説明の出力条件欄をご覧ください。
- 属性に **index** (通し番号) を持つ要素は **index** の順番に並びます。
- 基本的に固定長のデータは属性に持ちます。可変長のデータについては必ずデータのサイズを示す数値を属性で持ち、データ配列を要素内容として格納します。
- 要素の出力順序には決まりがあります。
- 要素内容に複数の数値を羅列する場合は、半角スペース・タブ・改行がデリミタ(区切り文字)となります。

3.3 属性(Attribute)について

3.3.1 属性の命名規則

- 属性名は、全てアルファベット小文字の英単語と '_' (アンダーバー) を組み合わせてつけられます。ローマ字表記やアルファベット大文字は含みません。
- 属性によっては以下のような略語を用いる場合があります(アルファベット順)。

color → clr
frame per seconds → fps
index → idx
interpolation → interp
matrix → mtx
maximum → max
minimum → min
normal → nrm
position → pos
previous → prev
primitive → prim
red green blue → rgb
texture → tex
vertex → vtx

width height distance → whd

- 属性名は各中間ファイル間や同一ファイル内の要素間で重複する場合があります。

3.3.2 その他の規則

- 属性の出力順序に決まりはありません。
- 存在が任意もしくは条件付きの属性については、その旨を説明欄に記述しています。特に触れられていない属性については必ず1つ出力されます。

3.4 共通ヘッダ部の要素及び属性説明

3.4.1 ルート要素

説明 全ての中間ファイルは、拡張子と同じ名前のルート要素の中に全ての情報を格納します。
出力条件 必ず1つ出力されます。
出力例 (imd の場合)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<imd version="1.0.0">
<head>
    ... (省略) ...
</head>
<body>
    ... (省略) ...
</body>
</imd>
```

属性

表 3-1 ルート要素の属性

属性名	説明
version	中間ファイルのバージョン。 実数x1

内容 ルート要素には、ドキュメントの作成情報を保持する<head>要素とドキュメントの本体となる<body>要素が1つずつこの順序で格納されます。

3.4.2 <head>要素

説明 作成者や作成時刻、コメントなどドキュメント全体に関する情報を格納する要素です。
この<head>要素以下は全ての中間ファイルに共通の形式で記述されます。

出力条件 必ず1つ出力されます。

出力例

```
<head>
  <create user="nintendo" host="nintendo-pc" date="2003-10-31T14:25:43"
    source="n64_mario.mb" />
  <title>Model Data for NITRO</title>
  <comment>free space</comment>
  <generator name="Maya 5.0.1 NNS_Export" version="1.0.0"/>
</head>
```

属性 なし

内容 <create>,<title>,<comment>,<generator>要素を1つずつこの順序で格納します。
ただし、<comment>要素のみ存在は任意です。

3.4.3 <create>要素

説明 ドキュメントの作成に関する情報を格納します。

出力条件 必ず1つ出力されます。

出力例

```
<head>
  <create user="nintendo" host="nintendo-pc" date="2003-10-31T14:25:43"
    source="n64_mario.mb" />
  <title>Model Data for NITRO</title>
  <comment>free space</comment>
  <generator name="Maya 5.0.1 NNS_Export" version="1.0.0"/>
</head>
```

属性

表 3-2 <create>要素の属性

属性名	説明
user	作成したユーザー名。 文字列x1
host	作成したマシン名。 文字列x1
date	作成日時。 文字列x1 。 書式は、YYYY-MM-DDThh:mm:ss とします。
source ※出力元が 3DCG ツールの時のみ存在	3DCGツール上のシーン名。 文字列x1 シーンが保存されていない場合は「untitled」となります。

内容 なし

3.4.4 <title>要素

説明 要素内容としてドキュメントの内容を説明する文字列を格納します。

出力条件 必ず1つ出力されます。

出力例

```

<head>
  <create user="nintendo" host="nintendo_pc" date="2003-10-31T14:25:43"
    source="n64_mario.mb" />
  <title>Model Data for NITRO</title>
  <comment>free space</comment>
  <generator name="Maya 5.0.1 NNS_Export" version="1.0.0"/>
</head>

```

属性 なし
 内容 文字列

3.4.5 <comment>要素

説明 要素内容としてドキュメントに対するコメントを格納します。
 出力条件 この要素の存在は任意です。
 出力例

```

<head>
  <create user="nintendo" host="nintendo_pc" date="2003-10-31T14:25:43"
    source="n64_mario.mb" />
  <title>Model Data for NITRO</title>
  <comment>free space. 日本語入力も可能です</comment>
  <generator name="Maya 5.0.1 NNS_Export" version="1.0.0"/>
</head>

```

属性 なし
 内容 任意の文字列

半角英数字だけでなく、全角文字や半角カナ文字を格納することもできます。

3.4.6 <generator>要素

説明 ドキュメントのジェネレータに関する情報が格納されます。
 出力条件 必ず1つ出力されます。
 出力例

```

<head>
  <create user="nintendo" host="nintendo_pc" date="2003-10-31T14:25:43"
    source="n64_mario.mb" />
  <title>Model Data for NITRO</title>
  <comment>free space</comment>
  <generator name="Maya 5.0.1 NNS_Export" version="1.0.0"/>
</head>

```

属性

表 3-3 <generator>要素の属性

属性名	説明
name	中間ファイルを出力したプログラム名。 文字列x1
version	中間ファイル出力プログラムのバージョン。 文字列x1

内容 なし

3.4.7 <body>要素

説明 ドキュメントの本体となる情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<?xml version="1.0" encoding="Shift_JIS"?>
<imd version="1.0.0">
<head>
    ...省略...
</head>
<body>
    ...省略...
</body>
</imd>
```

属性 なし

内容 各中間ファイルの種類ごとに異なる構成で複数の要素を格納します。

3.5 その他の規則

- Rotate の値について
Rotate は全て degree で出力されます。
- 浮動小数表記の値について

中間ファイルに出力される浮動小数表記の値は、TWL およびニンテンドーDS の仕様に合った固定小数値を浮動小数値に変換した値になっています。

NITRO ジオメトリコマンド等で使う場合は、それぞれの仕様に従って固定小数値に変換してください。例えば、NITRO ジオメトリコマンド VertexShort に相当する <pos_s>要素の xyz 値は、64 倍して小数第一位を四捨五入した値を使います。同様に、<node>要素の属性 translate の値は、Translate コマンドの仕様に従い、4096 倍して小数第一位を四捨五入した値を使います。

4 モデルデータ用中間ファイル

4.1 モデルデータファイル (imd)

4.1.1 imdの概要

モデルデータファイル(imd: Intermediate Model Data file)には、頂点を持つ情報(位置座標、法線ベクトル、頂点カラー、テクスチャ座標)・ポリゴン情報・親子構造・マテリアル・テクスチャイメージ・テクスチャパレットなど3Dモデルを表示するために必要な情報が格納されます。

imd ファイルには、TWL およびニンテンドーDS に最適な頂点データ形式で出力する **direct** 形式と、頂点位置座標及び頂点カラー値を操作し易い **index** 形式とがあります。頂点座標を操作して形状を変化させるシェイプアニメーションや頂点カラーアニメーションなど特殊な処理を行う場合以外は **direct** 形式の使用を奨励します。

(中間ファイルのシェイプアニメーション及び頂点カラーアニメーション対応は未定です)

imdファイルではTWLおよびニンテンドーDSの仕様に基づき、-8.0 以上 8.0 未満におさまるようにシフトした頂点位置座標を保持します。モデルの形状を正しく表示するには通常のScale,Rotate,Translate行列計算を行った後、位置座標行列にimdファイル内に出力される [pos scale](#) 値で行列の 3x3 部分をシフトするか、スケール行列として乗算する必要があります。

4.1.2 imdの基本的な要素構成

<imd>	…ルート
<head>	…ドキュメント全体に関する情報
… (省略) …	
</head>	
<body>	…データ本体
<original_create />	…最初に中間ファイルが作成された環境情報
<original_generator />	…最初に中間ファイルを作成したツール情報
<model_info />	…モデル情報
<box_test />	…ボックステスト情報
<vtx_pos_data>	…index形式用 頂点位置座標配列
<vtx_color_data>	…index形式用 頂点カラー配列
<tex_image_array>	…テクスチャイメージ配列
<tex_image>	…テクスチャイメージ情報
<bitmap />	…イメージデータ
<tex4x4_palette_idx />	…4x4テクセル圧縮テクスチャ用パレットデータ
</tex_image>	
</tex_image_array>	
<tex_palette_array>	…テクスチャパレット配列
<tex_palette />	…テクスチャパレット情報
</tex_palette_array>	
<material_array>	…マテリアル配列
<material />	…マテリアル情報
</material_array>	
<envelope>	…ウェイトエンベロープ情報
<weight />	…エンベロープウェイトデータ
<node_idx />	…エンベロープノードデータ
</envelope>	
<matrix_array>	…行列配列
<matrix />	…行列情報
</matrix_array>	
<polygon_array>	…ポリゴングループ配列
<polygon />	…ポリゴングループ情報
<mtx_prim>	…行列及びポリゴン情報
<mtx_list />	…行列リスト情報
<primitive_array>	…ポリゴン配列
<primitive>	…ポリゴン情報
<mtx />	…行列設定
<tex />	…テクスチャ座標設定
<nrm />	…法線ベクトル設定
<clr />	…頂点カラー設定
<clr_idx />	…頂点カラー設定 (index形式用)
<pos_s />	…頂点座標設定
<pos_xy />	… "
<pos_xz />	… "
<pos_yz />	… "
<pos_xyz />	… "
<pos_diff />	… "
<pos_idx />	…頂点座標設定 (index形式用)
</primitive>	
</primitive_array>	
</mtx_prim>	
</polygon>	
</polygon_array>	
<node_array>	…ノード配列
<node>	…ノード情報
<display />	
</node>	
</node_array>	
<output_info />	…出力情報
<ex_nns_3dme>	…3Dマテリアルエディタ専用情報
… (省略) …	

```
</ex_nns_3dme>
</body>
</imd>
```

4.1.3 imdの要素及び属性説明

4.1.3.1 <original_create>要素

説明 最初に中間ファイルが作成された時の環境情報を格納します。
3Dマテリアルエディタ等で中間ファイルを保存し直しても、内容は更新されません。

出力条件 必ず1つ出力されます。

出力例

```
<original_create user="nintendo" host="NINTENDO-PC"
                  date="2004-12-13T09:29:20" source="mario_wait"
/>
```

属性

表 4-1 <original_create>要素の属性

属性名	内容
user	作成したユーザー名。 文字列x1 ユーザー名が不明の場合は、「unknown」となります。
host	作成したマシン名。 文字列x1 マシン名が不明の場合は、「unknown」となります。
date	作成日時。 文字列x1。 書式は、YYYY-MM-DDThh:mm:ss とします。 作成日時が不明の場合は、「unknown」となります。
source ※出力元が3DCGツールの時のみ存在	3DCGツール上のシーン名。 文字列x1 シーンが保存されていない場合は「untitled」となります。

内容 なし

4.1.3.2 <original_generator>要素

説明 最初に中間ファイルを作成したツール情報を格納します。
3Dマテリアルエディタ等で中間ファイルを保存し直しても、内容は更新されません。

出力条件 必ず1つ出力されます。

出力例

```
<original_generator name="SoftimageXSI 4.0 NNS_Export"
                    version="1.5.0.20041213"
/>
```

属性

表 4-2 <original_generator>要素の属性

属性名	内容
name	中間ファイルを出力したプログラム名。 文字列x1 プログラム名が不明の場合は、「unknown」となります。
version	中間ファイル出力プログラムのバージョン。 文字列x1 出力プログラムのバージョンが不明の場合は、「unknown」となります。

内容 なし

4.1.3.3 <model_info>要素

説明 モデルデータ全体に関する情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<model_info
  pos_scale="0"
  scaling_rule="maya"
  vertex_style="direct"
  magnify="1.000000"
  tool_start_frame="1"
  tex_matrix_mode="maya"
  compress_node="none" node_size="31 31"
  compress_material="on" material_size="34 7"
  output_texture="used"
  force_full_weight="off"
  use_primitive_strip="on"
/>
```

属性

表 4-3 <model_info>要素の属性

属性名	説明
pos_scale	TWL やニンテンドーDS 上で頂点位置座標にかけるシフト値。 (0以上の整数)×1 CPUで行列計算する時はモデル行列の 3×3 要素をこの値でシフトします。 ジオメトリエンジンで行列計算を行う時は、(1<<pos_scale)倍のスケール行列をモデル行列に乗算します。 imdの概要 の説明も合わせてご覧ください。
scaling_rule	スケールの計算方法。 文字列[standard / maya / si3d]のいずれか ノード構成が親(a)、子(b)、孫(c)とすると、 standard : 一般的な行列計算。(Sc * Rc * Tc * Sb * Rb * Tb * Sa * Ra * Ta) maya : maya の Segment Scale Compensate を考慮した計算。 si3d : Softimage 方式の行列計算。 各計算方法については ノードの行列の計算方法 をご覧ください。
vertex_style	頂点位置座標 / 頂点カラーの出力方法。 文字列[direct / index]のいずれか direct : <polygon>内に NITRO コマンドに相当する要素を格納します。 index : <polygon>内にはインデックス番号を格納し、実データは別要素に格納します。 ※index 形式は将来、中間ファイルが頂点カラー/シェイプアニメーションに対応するための拡張用です。通常は TWL およびニンテンドーDS に最適なデータ形式で出力される direct 形式を指定ください。
magnify	3DCGツールから imd を出力する際にモデル全体にかけた倍率。 実数値×1 中間ファイル出力時の設定を確認するための情報です。 中間ファイル内の頂点位置座標・Translate 値等のデータに反映されていますので、TWL およびニンテンドーDS 上でこの値を使う必要はありません。
tool_start_frame	3DCGツールからファイル出力する際に指定した StartFrame。 整数×1 中間ファイル出力時の設定を確認するための情報です。 imd 内に格納されるキャラのポーズ(行列要素)は、この StartFrame 時のポーズもしくはエンベロープ設定時のポーズが格納されます。また、imd の各ノードに格納される visibility 等も StartFrame 時の状態が格納されます。

tex_matrix_mode	テクスチャ行列の計算方法。 文字列[maya / si3d / xsi / 3dsmax]のいずれか maya: Maya の計算方法。 si3d: SOFTIMAGE 3D の計算方法。 xsi: SOFTIMAGE XSI の計算方法。 3dsmax: 3ds max の計算方法。 各計算方法については テクスチャ行列の計算方法 をご覧ください。
compress_node	出力されたノードの圧縮状態。 文字列 [none / cull / merge / unite / unite_combine]のいずれか none: ツール上と同じノード構成で出力します。 cull: モデルの表示に必要なでないノードを削除して出力します。 merge: cull の処理に加え、合成できるノードの行列を合成して出力します。 unite: 1つのノードにまとめます。(主に地形データ用) unite_combine: unite の処理に加え、さらに同じ<material>要素で表示する<polygon>要素を1つにまとめます。
node_size	ノード未圧縮時 と 圧縮後のノード数。 (1以上の整数)×2 compress_node=none の時は、未圧縮時の数が2つ並びます。
compress_material	マテリアルの圧縮状態。 文字列[off / on]のいずれか off: ツール上と同じマテリアル数を出力します。 on: 設定内容が全て一致するマテリアルを1つにまとめます。
material_size	未圧縮時 と 圧縮後のマテリアル数。 (1以上の整数)×2 compress_material=off の時は、未圧縮時の数が2つ並びます。
output_texture	3DCGツールから imd にテクスチャデータを出力する際の条件。 文字列 [used / all]のいずれか used: imd に出力するモデルに使われているテクスチャだけを出力します。 all: 3DCG ツールのシーン内にある全てのテクスチャを imd に出力します。
force_full_weight	ウェイトエンベロープ設定された頂点を検出した時に(3DCG ツール上で最もウェイトがかかっているノードに対して 100%にして)、強制的にフルウェイトエンベロープにするかどうかのフラグ。 文字列 [off / on]のいずれか off: ウェイトエンベロープデータがあればその情報を出力します。 on: ウェイトエンベロープデータを強制的にフルウェイトとみなして出力します。
use_primitive_strip	連結ポリゴンを使っているかどうかのフラグ。 文字列 [off / on]のいずれか off: 連結ポリゴンは使わず、三角形ポリゴンもしくは四角形ポリゴンのどちらかでポリゴンを出力しています。 on: できるだけ、連結三角形ポリゴンと連結四角形ポリゴンを使い、連結できなかったポリゴンのみ三角形ポリゴンもしくは四角形ポリゴンで出力します。 off 設定に比べ、見た目は変わりませんが処理頂点数を減らすことができます。

内容 なし

4.1.3.4 <box_test>要素

説明 モデルが視体積に入るかテストするための情報が格納されます。

NITRO コマンド BoxTest に相当します。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```
<box_test pos_scale="2" xyz="-1.197510 -0.009277 1.412354"
whd="2.458740 4.608398 2.441406"/>
```

属性

表 4-4 <box_test>要素の属性

属性名	内容
pos_scale	BoxTest 専用シフト値。 (0以上の整数)x1 NITRO コマンド BoxTest を送る前にカレント位置座標行列の3x3要素をこの値でシフトします。
xyz	モデル全体を囲むボックスの基準頂点(左下奥)の x,y,z 座標。 実数x3 NITRO コマンド BoxTest の X,Y,Z 座標に相当します。
whd	モデル全体を囲むボックスの幅・高さ・奥行き。 実数x3 NITRO コマンド BoxTest の幅、高さ、奥行きに相当します。

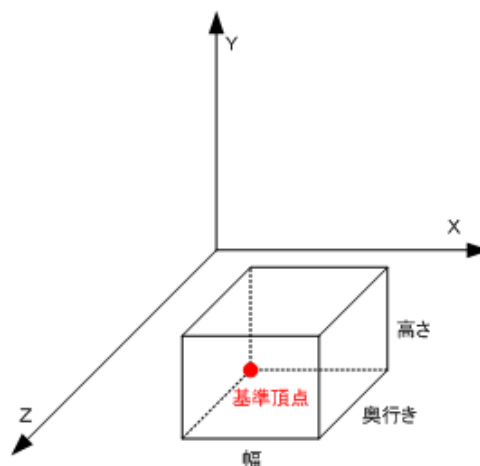


図 4-1 box_test の設定値

内容 なし

4.1.3.5 <vtx_pos_data>要素

説明 頂点の座標データが格納されます。

出力条件 ポリゴンが存在しかつ、<model_info>要素の属性 vertex_style が index の時のみ出力されます。

出力例

```
<vtx_pos_data pos_size="346">
  2.406982 -0.030518 0.998779
  2.897705 0.147705 0.010742
  ... (省略) ...
  1.701172 -0.427734 -2.048096
</vtx_pos_data>
```

属性

表 4-5 <vtx_pos_data>要素の属性

属性名	内容
pos_size	要素内容に格納される3次元座標の数。 (3以上の整数)x1 ※実際に要素内容に格納されるデータは(pos_size x3)個になります。

内容 -8.0 以上 8.0 未満の実数を(pos_size x3)個格納します。

4.1.3.6 <vtx_color_data>要素

説明 頂点カラーデータが格納されます。

出力条件 ポリゴンが存在しかつ、<model_info>要素の属性 vertex_style が index で、かつ頂点カラーが設定されている時だけ出力されます。

出力例

```
<vtx_color_data color_size="4">
  0 0 31
  0 31 1
  31 31 0
  31 0 0
</vtx_color_data>
```

属性

表 4-6 <vtx_color_data>要素の属性

属性名	内容
color_size	要素内容に格納されるRGBセットの数。 (1以上の整数) x1 ※実際に要素内容に格納されるデータは (color_size x3) 個になります。

内容 0以上31以下の整数を (color_size x3) 個格納します。

TWL およびニンテンドーDS の仕様に合わせ、R,G,B 各 0.0～1.0 の濃淡が0～31の整数に正規化されています。

4.1.3.7 <tex_image_array>要素

説明 全てのテクスチャのイメージデータが格納されます。

出力条件 テクスチャがある時のみ出力されます。

出力例

```
<tex_image_array size="2">
  <tex_image index="0" ... (省略) ... >
    ... (省略) ...
  </tex_image>
  <tex_image index="1" ... (省略) ... >
    ... (省略) ...
  </tex_image>
</tex_image_array>
```

属性

表 4-7 <tex_image_array>要素の属性

属性名	内容
size	要素内容として格納される<tex_image>要素の数。 (1以上の整数) x1

内容 <tex_image>要素を size 個格納します。

<tex_image>要素はイメージ名のアルファベット順(a→z,0→9)に格納されます。

4.1.3.8 <tex_image>要素

説明 テクスチャのイメージデータが格納されます。

出力条件 テクスチャがある時のみ出力されます。

出力例

```
<tex_image_array size="1">
  <tex_image index="0" name="p.1"
    width="64" height="64"
    original_width="64" original_height="64"
    format="palette256" color0_mode="color"
    palette_name="p.1_pl"
    path="D:/data/TextureImage/number_pic/p.1.pic"
  >
  <bitmap size="2048">
    ... (省略) ...
  </bitmap>
</tex_image>
</tex_image_array>
```

属性

表 4-8 <tex_image>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)x1
name	イメージの名前 (= テクスチャ画像ファイル名)。 文字列x1
width	イメージの横幅。 整数 [8 / 16 / 32 / 64 / 128 / 256 / 512 / 1024]のいずれか NITRO コマンド TexImageParam のテクスチャサイズに相当します。 テクスチャ画像ファイルの横幅が TWL やニンテンドーDS で使えるサイズ (8, 16, 32, 64, 128, 256, 512, 1024) になっていない場合、右端のテクセルの色を付加して横幅が TWL やニンテンドーDS で使えるサイズになるように合わせます。
height	イメージの縦幅。 整数 [8 / 16 / 32 / 64 / 128 / 256 / 512 / 1024]のいずれか NITRO コマンド TexImageParam のテクスチャサイズに相当します。 テクスチャ画像ファイルの縦幅が TWL やニンテンドーDS で使えるサイズ (8, 16, 32, 64, 128, 256, 512, 1024) になっていない場合、下端のテクセルの色を付加して縦幅が TWL やニンテンドーDS で使えるサイズになるように合わせます。
original_width	テクスチャ画像ファイルの横幅。 (1以上の整数)x1
original_height	テクスチャ画像ファイルの縦幅。 (1以上の整数)x1
format	テクスチャのフォーマット形式。 文字列 [palette4 / palette16 / palette256 / tex4x4 / a3i5 / a5i3 / direct]のいずれか palette4: 4色パレットテクスチャ palette16: 16色パレットテクスチャ palette256: 256色パレットテクスチャ tex4x4: 4x4 テクセル圧縮テクスチャ a3i5: A3I5 半透明テクスチャ a5i3: A5I3 半透明テクスチャ direct: ダイレクトテクスチャ NITRO コマンド TexImageParam のテクスチャフォーマットに相当します。
color0_mode ※format が palette4, palette16, palette256 のいずれかの時のみ存	パレットのカラー0をそのまま使うかヌキとして使うかのフラグ。 文字列 [color / transparency]のいずれか color: 色情報をそのまま使います。 transparency: ヌキ (アルファ=0) にします。

在します	NITRO コマンド TexImageParam のカラー0設定値イネーブルフラグに相当します。
palette_name ※format が direct 以外の時のみ存在します	テクスチャ画像ファイルで設定されていたパレット名 (<tex_palette>要素の名前に相当します)。 文字列x1
path ※存在は任意	PC上でのテクスチャ画像ファイルのパス。 文字列 x1 ※フォルダ間の区切りは"¥"ではなく"/(スラッシュ)"とします。

内容 <bitmap>要素を1つ格納します。
format が tex4x4 の時には、<bitmap>要素の後に<tex4x4_palette_idx>要素を格納します。

4.1.3.9 <bitmap>要素

説明 イメージの実データが格納されます。
出力条件 テクスチャがある時のみ出力されます。
出力例

```
<tex_image_array size="1">
  <tex_image index="0" name="p.1"
    width="64" height="64"
    original_width="64" original_height="64"
    format="palette256" color0_mode="color"
    palette_name="p.1_pl"
    path="D:/texture/p.1.pic"
  >

    <bitmap size="2048">
      1f1f 1f1f 1f1f 1f1f 1f1f 1f1f 1f1f 1f1f
      ... (省略) ...
      1f1f 1f1f 1f1f 1f1f 1f1f 1f1f 1f1f 1f1f
    </bitmap>
  </tex_image>
</tex_image_array>
```

属性

表 4-9 <bitmap>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 整数x1

内容 <tex_image>要素の属性 format が tex4x4 の時は符号なし32ビット16進数が、
tex4x4 以外の時は符号なし16ビット16進数が、size 個並びます。
※16進のアルファベット表記は大文字でも小文字でも構いません。

4.1.3.10 <tex4x4_palette_idx>要素

説明 4x4 テクセル圧縮テクスチャ用のパレットインデックスの実データが格納されます。
出力条件 4x4 テクセル圧縮テクスチャがある時のみ出力されます。
出力例

```

<tex_image index="6" name="tex4x4_cmp2"
    idth="64" height="64"
    original_width="64" original_height="64"
    format="tex4x4" color0_mode="color"
    palette_name="tex4x4_cmp2_pl"
    path="D:/texture/tex4x4_cmp2.tga"
>
<bitmap size="256">
    55aaa855 95ba002d 545e5e5c 2d555555
    ... (省略) ...
    0ff05575 0affd5f5 002bff57 2bbf5555
</bitmap>
<tex4x4_palette_idx size="256">
    c000 c001 c002 c003 c004 c005 c006 c007
    ... (省略) ...
    c010 c011 c012 c013 c014 c015 c016 c017
</tex4x4_palette_idx>
</tex_image>

```

属性

表 4-10 <tex4x4_palette_idx>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 整数x1

内容 符号なし16ビット16進数が、size 個並びます。

※16進のアルファベット表記は大文字でも小文字でも構いません。

4.1.3.11 <tex_palette_array>要素

説明 全てのテクスチャパレットデータが格納されます。

出力条件 パレットを用いるテクスチャがある時のみ出力されます。

出力例

```

<tex_palette_array size="2">
    <tex_palette index="0" name="p.1_pl" color_size="16">
        001f 043f 085f 0c7f 109f 14bf 18df 1cff
        631f 673f 6b5f 6f7f 739f 77bf 7bdf 7fff
    </tex_palette>
    <tex_palette index="1" name="p.2_pl" color_size="32">
        ... (省略) ...
    </tex_palette>
</tex_palette_array>

```

属性

表 4-11 <tex_palette_array>要素の属性

属性名	内容
size	要素内容に格納される<tex_palette>要素の数。 (1以上の整数)x1

内容 <tex_palette>要素を size 個格納します。

<tex_palette>要素はパレット名のアルファベット順(a→z,0→9)に格納されます。

4.1.3.12 <tex_palette>要素

説明 パレットの実データが格納されます。

出力条件 パレットを用いるテクスチャがある時のみ出力されます。

出力例

```
<tex_palette_array size="1">
  <tex_palette index="0" name="p.1_pl" color_size="16">
    001f 043f 085f 0c7f 109f 14bf 18df 1cff
    631f 673f 6b5f 6f7f 739f 77bf 7bdf 7fff
  </tex_palette>
</tex_palette_array>
```

属性

表 4-12 <tex_palette>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)×1
name	パレット名。 文字列×1
color_size	パレット内の色数。 整数 [4/8の倍数]のいずれか

内容 符号なし16ビット16進数を color_size 個格納します。

※16進のアルファベット表記は大文字でも小文字でも構いません。

4.1.3.13 <material_array>要素

説明 全てのマテリアルデータが格納されます。

出力条件 マテリアルが存在する時のみ出力されます。

出力例

```
<material_array size="2">
  <material index="0" ... (省略) ... />
  <material index="1" ... (省略) ... />
</material_array>
```

属性

表 4-13 <material_array>要素の属性

属性名	内容
size	要素内容に格納される<material>要素の数。 (1以上の整数)×1

内容 <material>要素を size 個格納します。

<material>要素はマテリアル名のアルファベット順(a→z,0→9)に格納されます。

4.1.3.14 <material>要素

説明 マテリアルデータが格納されます。

出力条件 マテリアルが存在する時のみ出力されます。

出力例

```

<material_array size="1">
  <material index="0" name="lambert2"
    light0="on" light1="off" light2="off" light3="off"
    face="front"
    alpha="31"
    wire_mode="off"
    polygon_mode="modulate"
    polygon_id="0"
    fog_flag="off"
    depth_test_decals="on"
    translucent_update_depth="on"
    render_1_pixel="on"
    far_clipping="on"
    diffuse="25 25 25"
    ambient="0 0 0"
    specular="0 0 0"
    emission="0 0 0"
    shininess_table_flag="off"
    tex_image_idx="0" tex_palette_idx="0"
    tex_tiling="repeat repeat"
    tex_scale="1.000000 1.000000"
    tex_rotate="0.000000"
    tex_translate="0.000000 0.000000"
    tex_gen_mode="nrm"
    tex_gen_st_src="material"
    tex_effect_mtx="1.000000 0.000000 0.000000 0.000000
                    0.000000 1.000000 0.000000 0.000000
                    0.000000 0.000000 1.000000 0.000000
                    0.000000 0.000000 0.000000 1.000000"
  />
</material_array>

```

属性

表 4-14 <material>要素の属性

属性名	内容
index	通し番号。(0以上の整数)x1
name	マテリアル名。文字列 x1
light0	ライト0のイネーブルフラグ。文字列 [off / on]のいずれか NITRO コマンド PolygonAttr のライトイネーブルフラグ0に相当します。
light1	ライト1のイネーブルフラグ。文字列 [off / on]のいずれか NITRO コマンド PolygonAttr のライトイネーブルフラグ1に相当します。
light2	ライト2のイネーブルフラグ。文字列 [off / on]のいずれか NITRO コマンド PolygonAttr のライトイネーブルフラグ2に相当します。
light3	ライト3のイネーブルフラグ。文字列 [off / on]のいずれか NITRO コマンド PolygonAttr のライトイネーブルフラグ3に相当します。
face	ポリゴンを描画する面。文字列 [front / back / both]のいずれか front: 表面のみ表示します。 back: 裏面のみ表示します。 both: 両面を表示します。 NITRO コマンド PolygonAttr のポリゴン描画面に相当します。
alpha	ポリゴンアルファ。(0 以上 31 以下の整数)x1 NITRO コマンド PolygonAttr のアルファ値に相当します。 この値が0の時は、このマテリアルが割り当てられているポリゴンの描画をスキップし非表示にしてください。
wire_mode	ワイヤーの表示モード。文字列 [off / on]のいずれか off: alpha 値を使います。alpha が0の時は非表示にします。 on: 常にワイヤーフレーム($\alpha = 0$)で表示します。

polygon_mode	ポリゴンの描画モード 文字列 [modulate / decal / toon_highlight / shadow]のいずれか NITRO コマンド PolygonAttr のポリゴンモードに相当します。
polygon_id	ポリゴンID。 (0以上63以下の整数)x1 NITRO コマンド PolygonAttr のポリゴンIDに相当します。
fog_flag	フォグイネーブルフラグ。 文字列 [off / on]のいずれか NITRO コマンド PolygonAttr のフォグイネーブルフラグに相当します。
depth_test_decal	デプステスト条件。 文字列 [off / on] のいずれか off: フラグメントのデプス値がデプスバッファのデプス値よりも小さい場合に描画します。 on: フラグメントのデプス値がデプスバッファのデプス値と等しい場合に描画します (デカルポリゴン)。 NITRO コマンド PolygonAttr のデプステスト条件に相当します。
translucent_update_depth	半透明ポリゴンのデプス値更新イネーブルフラグ。 文字列 [off / on] のいずれか off: 半透明ポリゴン描画時にデプスバッファを更新しません。 on: 半透明ポリゴン描画時にデプスバッファを更新します。 NITRO コマンド PolygonAttr の半透明ポリゴンのデプス値更新イネーブルフラグに相当します。
render_1_pixel	1ピクセル(ドット)ポリゴンのレンダリング指定。 文字列 [off / on] のいずれか off: 1ピクセル(ドット)になったらレンダリングしません。 on: 1ピクセル(ドット)になってもレンダリングします。 NITRO コマンド PolygonAttr の1ドットポリゴンの表示指定に相当します。
far_clipping	FAR面交差ポリゴン表示指定。 文字列 [off / on] のいずれか off: FAR面と交差したら消去します。 on: FAR面と交差したらクリッピングします。 NITRO コマンド PolygonAttr のFAR面交差ポリゴン表示指定に相当します。
diffuse	拡散反射色 (R, G, B)。 (0以上31以下の整数)x3 NITRO コマンド MaterialColor0 の diffuse に相当します。
ambient	環境反射色 (R, G, B)。 (0以上31以下の整数)x3 NITRO コマンド MaterialColor0 の ambient に相当します。
specular	鏡面反射色 (R, G, B)。 (0以上31以下の整数)x3 NITRO コマンド MaterialColor1 の specular に相当します。
emission	放射光色 (R, G, B)。 (0以上31以下の整数)x3 NITRO コマンド MaterialColor1 の emission に相当します。
shininess_table_flag	鏡面反射輝度テーブルフラグ。 文字列 [off / on]のいずれか NITRO コマンド MaterialColor1 の鏡面反射輝度テーブル・イネーブルフラグに相当します。
tex_image_idx	テクスチャのイメージ番号。 (-1以上の整数)x1 テクスチャは貼られていない時は-1になります。
tex_palette_idx	テクスチャのパレット番号。 (-1以上の整数)x1 パレットを使うテクスチャが貼られていない時は-1になります。
tex_tiling ※tex_image_idx が0以上の時に存在	テクスチャのタイリング方法。 文字列 [clamp / repeat / flip]のいずれか x2 NITRO コマンド TexImageParam のフリップ・リピートに相当します。 clamp: リピートしない・フリップしない。 repeat: リピートする・フリップしない。 flip: リピートする・フリップする。
tex_scale ※tex_image_idx が0以上の時に存在	テクスチャ行列に設定する ScaleS, ScaleT 値。 実数x2
tex_rotate ※tex_image_idx が0以上の時に存在	テクスチャ行列に設定する Rotate 値。 (-180.0 以上 180.0 未満の実数)x1
tex_translate ※tex_image_idx が0以上の時に存在	テクスチャ行列に設定する TranslateS, TranslateT 値。 実数x2
tex_gen_mode ※tex_image_idx が0以上の時に存在	テクスチャ座標変換モード。 文字列 [none / tex / nrm / pos]のいずれか NITRO コマンド TexImageParam のテクスチャ座標変換モードに相当します。 none: テクスチャ座標変換なし tex: TexCoord ソース

	nrm: Normal ソース pos: Vertex ソース 環境マッピングを行う場合は nrm 、投影マッピングを行う場合は pos を指定します。 ※<polygon>要素内の<nrm>要素は、通常は法線ベクトル方向が変わる毎に格納されますが、 tex_gen_mode="nrm" かつ tex_gen_st_src="polygon" の場合は、テクスチャ座標が変わった後にも格納されます。 (<tex>要素と<pos_***>要素の間に必ず<nrm>要素が格納されます)
tex_gen_st_src ※tex_gen_mode が nrm か pos の時に存在	tex_gen_mode="nrm"(テクスチャ座標変換モードが Normal ソース)の時、もしくは tex_gen_mode="pos"(テクスチャ座標変換モードが Vertex ソース)の時に、対応する<polygon>要素内にテクスチャ座標(<tex>要素)を出力するかどうかのフラグ。 文字列 [polygon / material]のいずれか polygon: 対応する<polygon>要素内に<tex>要素が出力されます。 普通に貼られたテクスチャを法線ベクトルや頂点座標に従って変形するような特殊なマッピング表現が可能です。 material: 対応する<polygon>要素内に<tex>要素が出力されません。 一般的な環境/投影マッピングを行う場合は、こちらを指定してください。
tex_effect_mtx ※tex_gen_mode が nrm か pos の時に存在	テクスチャを環境マッピングや投影マッピングなどで表示する際に影響を与える 4x4 行列の要素。 実数x16 値は、4x4 行列の以下の並びに対応します。 $\begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix}$ ※「 imdの補足 」の "tex_effect_mtxの使い方" も合わせてご覧ください。

内容 なし

4.1.3.15 <envelope>要素

説明 ウェイトエンベロープ用のデータが格納されます。
行列の参照手順については [こちら\(頂点位置座標を変換する行列について\)](#) も併せてご覧ください。

出力条件 ウェイトエンベロープが設定されている時のみ出力されます。

出力例

```

<envelope>
  <weight size="11">
    60 40 50 42 8 50 50 60 40 50
    50
  </weight>
  <node_idx size="11">
    8 9 8 9 6 8 9 5 6 5
    6
  </node_idx>
</envelope>

```

属性 なし

内容 <weight>、<node_idx>要素を1つずつこの順序で格納します。

4.1.3.16 <weight>要素

説明 ウェイトエンベロープの重み情報が格納されます。

出力条件 ウェイトエンベロープが設定されている時のみ出力されます。

出力例

```
<envelope>
  <weight size="11">
    60 40 50 42 8 50 50 60 40 50
    50
  </weight>
  <node_idx size="11">
    8 9 8 9 6 8 9 5 6 5
    6
  </node_idx>
</envelope>
```

属性

表 4-15 <weight>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 1以上99以下の整数を size 個格納します。
(単位は%。必ず足して100になるようにします)

※ここに出力される値は<node_idx>の並びと対になります。

4.1.3.17 <node_idx>要素

説明 ウェイトエンベロープのノード情報が格納されます。

出力条件 ウェイトエンベロープが設定されている時のみ出力されます。

出力例

```
<envelope>
  <weight size="11">
    60 40 50 42 8 50 50 60 40 50
    50
  </weight>
  <node_idx size="11">
    8 9 8 9 6 8 9 5 6 5
    6
  </node_idx>
</envelope>
```

属性

表 4-16 <node_idx>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 0以上の整数(=ノード番号)を size 個格納します。

※ここに出力される値は<weight>の並びと対になります。

4.1.3.18 <matrix_array>要素

説明 描画に必要な全ての行列情報が格納されます。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```
<matrix_array size="3">
  <matrix index="0" mtx_weight="2" envelope_head="0"/>
  <matrix index="1" mtx_weight="1" node_idx="8"/>
  <matrix index="2" mtx_weight="1" node_idx="9"/>
</matrix_array>
```

属性

表 4-17 <matrix_array>要素の属性

属性名	内容
size	要素内容に格納される<matrix>要素の数。 (1以上の整数)x1

内容 <matrix>要素を size 個格納します。

※mtx_weight 値の大きい順、envelope_head の小さい順、node_idx の小さい順に通し番号をつけて格納します。

4.1.3.19 <matrix>要素

説明 描画に使う行列情報が格納されます。

行列の参照手順については [こちら\(頂点位置座標を変換する行列について\)](#)も併せてご覧ください。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```
<matrix_array size="4">
  <matrix index="0" mtx_weight="3" envelope_head="2"/>
  <matrix index="1" mtx_weight="2" envelope_head="0"/>
  <matrix index="2" mtx_weight="1" node_idx="9"/>
  <matrix index="3" mtx_weight="1" node_idx="10"/>
</matrix_array>
```

属性

表 4-18 <matrix>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)x1
mtx_weight	重みのかかるノードの数。 (1以上の整数)x1 1の時: ある1つのノードの行列に依存します。 2以上の時: 複数のノードの行列に依存します(=ウェイトエンベロープ)。
node_idx ※mtx_weight が1の時	この値をインデックスとする<node>要素で計算された行列を頂点頂点座標変換行列として用います。 (0以上の整数)x1
envelope_head ※mtx_weight が2以上の時	<envelope>内のデータの先頭位置。 (0以上の整数)x1 <weight>,<node_idx>要素を参照します。 envelope_head 番目から mtx_weight 個をノード番号、重み値としてエンベロープ計算を行います。

内容 なし

4.1.3.20 <polygon_array>要素

説明 全てのポリゴン情報が格納されます。
出力条件 ポリゴンが存在する時のみ出力されます。
出力例

```
<polygon_array size="2">
  <polygon index="0" name="polygon0" ... (省略) ... >
    ... (省略) ...
  </polygon>
  <polygon index="1" name="polygon1" ... (省略) ... >
    ... (省略) ...
  </polygon>
</polygon_array>
```

属性

表 4-19 <polygon_array>要素の属性

属性名	内容
size	要素内容に格納される<polygon>要素の数。 (1以上の整数)×1

内容 <polygon>要素を size 個格納します。

4.1.3.21 <polygon>要素

説明 ポリゴン情報が格納されます。
出力条件 ポリゴンが存在する時のみ出力されます。
出力例

```
<polygon index="0" name="polygon0"
  vertex_size="4" polygon_size="1"
  triangle_size="0" quad_size="1"
  volume_min="-5.000000 -5.000000 5.000000"
  volume_max="5.000000 5.000000 5.000000"
  volume_r="8.660254"
  mtx_prim_size="1"
  nrm_flag="off" clr_flag="off" tex_flag="on"
>
  <mtx_prim index="0">
    <mtx_list size="1">0</mtx_list>
    <primitive_array size="5">
      ... (省略) ...
    </primitive_array>
  </mtx_prim>
</polygon>
```

属性

表 4-20 <polygon>要素の属性

属性名	内容
index	通し番号。(0以上の整数)x1
name	このポリゴン群の名前。 文字列x1 ※ Maya 、 SOFTIMAGE XSI から出力した場合、 "polygon + 通し番号"になります。 3ds max から出力した場合、 3ds max 上のポリゴン名になります。ただし、同じ 名前のポリゴン名が複数定義されている場合、語尾に"_通し番号"が付加されま す。
vertex_size	処理頂点数。(3以上の整数)x1
polygon_size	総ポリゴン数(=三角形ポリゴン数+四角形ポリゴン数)。(1以上の整数)x1
triangle_size	三角形ポリゴン数。(0以上の整数)x1
quad_size	四角形ポリゴン数。(0以上の整数)x1
volume_min	このポリゴン群に外接する格子の左下奥のローカル座標値(X, Y, Z)。 実数x3 当たり判定などに利用できます。
volume_max	このポリゴン群に外接する格子の右上手前のローカル座標値(X, Y, Z)。 実数x3 当たり判定などに利用できます。
volume_r	このポリゴン群に外接する球の半径。 実数x1 当たり判定などに利用できます。
mtx_prim_size	このポリゴン群に格納される<mtx_prim>要素の数。(1以上の整数)x1
nrm_flag	<nrm>要素が格納されているかどうかのフラグ。 文字列 [off/on]のいずれか
clr_flag	<clr>要素が格納されているかどうかのフラグ。 文字列 [off/on]のいずれか
tex_flag	<tex>要素が格納されているかどうかのフラグ。 文字列 [off/on]のいずれか

内容 <mtx_prim>要素を mtx_prim_size 個格納します。

4.1.3.22 <mtx_prim>要素

説明 行列リストとポリゴンデータが格納されます。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```
<mtx_prim index="0">
  <mtx_list size="2">0 1</mtx_list>
  <primitive_array size="1">
    ... (省略) ...
  </primitive_array>
</mtx_prim>
```

属性

表 4-21 <mtx_prim>要素の属性

属性名	内容
index	通し番号。(0以上の整数)x1

内容 <mtx_list>,<primitive_array>要素を1つずつこの順序で格納します。

4.1.3.23 <mtx_list>要素

説明 描画に必要な行列情報が格納されます。
行列の参照手順については [こちら\(頂点位置座標を変換する行列について\)](#) も併せてご覧ください。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```
<mtx_prim index="0">
  <mtx_list size="2">0 1</mtx_list>
  <primitive_array size="1">
    ... (省略) ...
  </primitive_array>
</mtx_prim>
```

属性

表 4-22 <mtx_list>要素の属性

属性名	内容
size	要素内容に含まれるデータの数。 (1以上31以下の整数)×1

内容 次に続く<primitive_array>要素の中のポリゴンを描画するのに必要な行列番号(<matrix>要素へのインデックス番号)を size 個格納します。

エンベロープを使っていない場合は、属性 size は必ず1で、内容には(0以上の整数)を1つ格納します。

エンベロープを使っている場合は、内容に(-1以上の整数)を(size)個格納します。

内容に格納される0以上の値は<matrix>要素へのインデックス番号を意味しますが、<polygon>要素の属性 mtx_prim_size が2以上の時にのみ、2つ目以降の<mtx_list>要素の内容に-1 が格納される場合があります。-1 は一つ前の<mtx_list>の同じ箇所で指定された行列がそのまま使えるため、行列スタックを更新する必要がないことを意味します。

4.1.3.24 <primitive_array>要素

説明 ポリゴン情報が格納されます。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```

<mtx_prim index="0">
  <mtx_list size="2">0 1</mtx_list>
  <primitive_array size="5">
    <primitive index="0" type="quad_strip" vertex_size="6">
      ... (省略) ...
    </primitive>
    <primitive index="1" type="triangle_strip" vertex_size="8">
      ... (省略) ...
    </primitive>
    <primitive index="2" type="triangle_strip" vertex_size="7">
      ... (省略) ...
    </primitive>
    <primitive index="3" type="quads" vertex_size="4">
      ... (省略) ...
    </primitive>
    <primitive index="4" type="triangles" vertex_size="3">
      ... (省略) ...
    </primitive>
  </primitive_array>
</mtx_prim>

```

属性

表 4-23 <primitive_array>要素の属性

属性名	内容
size	要素内容に格納される<prim>要素の数。(1以上の整数)×1

内容

<primitive>要素を size 個格納します。

※quad_strip、triangle_strip、quads、triangles の順に、
またそれぞれ処理頂点数が大きい順に並べて通し番号をつけて格納します。

4.1.3.25 <primitive>要素

説明

ポリゴンデータが格納されます。

NITRO コマンド Begin / End に相当します。

出力条件

ポリゴンが存在する時のみ出力されます。

出力例

```

<primitive_array size="2">
  <primitive index="0" type="triangle_strip" vertex_size="9">
    ... (省略) ...
  </primitive>
  <primitive index="1" type="triangle_strip" vertex_size="6">
    <pos_xy xy="2.406982 -0.030518"/>
    <nrm xyz="0.492188 0.726563 0.478516"/>
    <pos_xyz xyz="1.970703 1.489746 0.651611"/>
    <nrm xyz="-0.521484 0.636719 0.568359"/>
    <pos_xyz xyz="0.433350 1.601318 0.760986"/>
    <nrm xyz="-0.544922 0.605469 -0.582031"/>
    <pos_xyz xyz="0.415771 1.597168 -0.793457"/>
    <nrm xyz="-0.949219 0.042969 -0.310547"/>
    <pos_xyz xyz="0.037354 0.813232 -0.375732"/>
    <nrm xyz="-0.441406 -0.326172 -0.835938"/>
    <pos_xyz xyz="0.677246 -0.232422 -1.155518"/>
  </primitive>
</primitive_array>

```

属性

表 4-24 <primitive>要素の属性

属性名	内容
index	通し番号。(0以上の整数)x1
type	ポリゴンの描画方法 文字列 [triangles / quads / triangle_strip / quad_strip]のいずれか triangles : 三角形ポリゴンで表示します。 quads : 四角形ポリゴンで表示します。 triangle_strip : 連結三角形ポリゴンで表示します。 quad_strip : 連結四角形ポリゴンで表示します。 NITRO コマンド Begin のプリミティブタイプに相当します。
vertex_size	処理頂点数。(3以上の整数)x1

内容 <mtx>,<tex>,<nrm>,<clr>,<clr_idx>,<pos_s>,<pos_xy>,<pos_yz>,<pos_xz>,<pos_xyz>,<pos_diff>,<pos_idx>要素を頂点リストの順番に合わせて複数格納します。

※<pos_*>要素は vertex_size 個出力されます。

※<model_info>要素の vertex_type が index の時は、<clr>要素が<clr_idx>要素に、<pos_*>要素が<pos_idx>要素に置き換わります

4.1.3.26 <mtx>要素

説明 参照する行列の行列リスト番号(=<mtx_list>要素内容への番号)が格納されます。
CPUで行列計算をして(フル)ウェイトエンベロープモデルを表示する場合、
NITRO コマンド RestoreMatrix に相当します。
行列の参照手順については [こちら\(頂点位置座標を変換する行列について\)](#)も併せてご覧ください。

出力条件 ポリゴンが存在する時のみ出力されます。

出力例

```

<mtx_list size="2">0 2</mtx_list>
<primitive_array size="1">
  <primitive index="0" type="triangles" vertex_size="3">
    <mtx idx="0"/>
    <nrm xyz="0.000000 0.998047 0.000000"/>
    <pos_xyz xyz="-7.899902 0.000000 7.899902"/>
    <pos_xy xy="7.899902 0.000000"/>
    <pos_xz xz="7.899902 -7.899902"/>
  </primitive>
</primitive_array>

```

属性

表 4-25 <mtx>要素の属性

属性名	内容
idx	描画に使用する行列番号。 (0以上の整数)x1 <mtx_list>を参照します。

内容 なし

4.1.3.27 <tex>要素

説明 1組のテクスチャ座標データ(s,t)が格納されます。

NITRO コマンド TexCoord に相当します。

出力条件 テクスチャが貼られたポリゴンがある時のみ出力されます。

ただし、対応する<material>要素の tex_gen_st_src="material"の場合は出力されません。

出力例

```

<primitive index="1" type="triangles" vertex_size="3">
  <tex st="0.000000 0.000000"/>
  <nrm xyz="0.000000 0.000000 0.998047"/>
  <pos_s xyz="-2.500000 -2.500000 0.000000"/>
  <tex st="64.000000 0.000000"/>
  <pos_xy xy="2.500000 -2.500000"/>
  <tex st="0.000000 64.000000"/>
  <pos_xy xy="-2.500000 2.500000"/>
</primitive>

```

属性

表 4-26 <tex>要素の属性

属性名	内容
st	テクスチャ座標の s,t 成分。 実数x2

内容 なし

4.1.3.28 <nrm>要素

説明 1組の法線ベクトルデータ(nx,ny,nz)が格納されます。

NITRO コマンド Normal に相当します。

出力条件 ライト計算を行う時のみ出力されます。

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <tex st="0.000000 0.000000"/>
  <nrm xyz="0.000000 0.000000 0.998047"/>
  <pos_s xyz="-2.500000 -2.500000 0.000000"/>
  <tex st="64.000000 0.000000"/>
  <pos_xy xy="2.500000 -2.500000"/>
  <tex st="0.000000 64.000000"/>
  <pos_xy xy="-2.500000 2.500000"/>
</primitive>
```

属性

表 4-27 <nrm>要素の属性

属性名	内容
xyz	法線ベクトルの x,y,z 成分。 (-1.0 以上 0.998047 以下の実数)×3

内容 なし

4.1.3.29 <clr>要素

説明 1組の頂点カラーデータ(r,g,b)が格納されます。

NITRO コマンド Color もしくは MaterialColor0 に相当します。

出力条件 <model_info>要素の vertex_style が direct でかつ、頂点カラーが使われている時に出力されます。

出力例

```
<primitive index="2" type="triangles" vertex_size="3">
  <clr rgb="0 1 31"/>
  <pos_xyz xyz="-7.899902 0.000000 7.899902"/>
  <clr rgb="0 31 0"/>
  <pos_xy xy="7.899902 0.000000"/>
  <clr rgb="31 30 0"/>
  <pos_xz xz="7.899902 -7.899902"/>
</primitive>
```

属性

表 4-28 <clr>要素の属性

属性名	内容
rgb	頂点カラーの r,g,b 成分。 (0以上31以下の整数)×3

内容 なし

4.1.3.30 <clr_idx>要素

説明 頂点カラーデータ(=<vtx_color_data>要素)へのインデックス番号が格納されます。

出力条件 <model_info>要素の vertex_style が index でかつ、頂点カラーが使われている時に出力されます。

出力例


```

<primitive index="2" type="triangles" vertex_size="3">
  <tex st="0.000000 0.000000"/>
  <clr_idx idx="0"/>
  <pos_idx idx="6"/>
  <tex st="16.000000 0.000000"/>
  <clr_idx idx="1"/>
  <pos_idx idx="9"/>
  <tex st="16.000000 16.000000"/>
  <clr_idx idx="2"/>
  <pos_idx idx="7"/>
</primitive>

```

属性

表 4-29 <clr_idx>要素の属性

属性名	内容
idx	頂点カラーデータへのインデックス番号。 (0以上の整数)x1 <vtx_color_data>要素の (idx x3) 番目から3の値を r,g,b として参照します。

内容 なし

4.1.3.31 <pos_s>要素

説明 1組の頂点位置座標(x,y,z)が格納されます。
NITRO コマンド VertexShort に相当します。

出力条件 以下の条件を全て満たす時に出力されます。

- ポリゴンが存在してかつ vertex_style が direct
- <pos_xy>,<pos_xz>,<pos_yz>,<pos_diff>の条件に該当しない
- x,y,z それぞれ固定小数値に変換した際に小数 12 ビットのうち下位 6 ビットが全て0の時

出力例

```

<primitive index="1" type="triangles" vertex_size="3">
  <nrm xyz="0.000000 0.000000 0.998047"/>
  <pos_s xyz="-2.500000 -2.500000 0.000000"/>
  <pos_xy xy="2.500000 -2.500000"/>
  <pos_yz yz="2.500000 2.500000"/>
</primitive>

```

属性

表 4-30 <pos_s>要素の属性

属性名	内容
xyz	頂点位置座標。 (-8.0 以上 8.0 未満の実数)x3

内容 なし

4.1.3.32 <pos_xy>要素

説明 1組の頂点位置座標の(x,y)座標が格納されます。
NITRO コマンド VertexXY に相当します。

出力条件 以下の条件を全て満たす時に出力されます。

- ポリゴンが存在してかつ `vertex_style` が `direct`
- `z`座標が直前に設定した`z`座標値と同じ時

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <pos_xyz xyz="2.500000 0.100000 -2.500000"/>
  <pos_xy xy="-1.500000 -1.500000"/>
  <pos_yz yz="3.500000 3.500000"/>
</primitive>
```

属性

表 4-31 <pos_xy>要素の属性

属性名	内容
xy	頂点位置座標のx、y成分。 (-8.0 以上 8.0 未満の実数)x2

内容 なし

4.1.3.33 <pos_xz>要素

説明 1組の頂点位置座標の(x,z)座標が格納されます。

NITRO コマンド `VertexXZ` に相当します。

出力条件 以下の条件を全て満たす時に出力されます。

- ポリゴンが存在してかつ `vertex_style` が `direct`
- `y`座標が直前に設定した`y`座標値と同じ時

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <pos_xy xy="-1.500000 -1.500000"/>
  <pos_xz xz="2.500000 -2.500000"/>
  <pos_yz yz="3.500000 3.500000"/>
</primitive>
```

属性

表 4-32 <pos_xz>要素の属性

属性名	内容
xz	頂点位置座標のx、z成分。 (-8.0 以上 8.0 未満の実数)x2

内容 なし

4.1.3.34 <pos_yz>要素

説明 1組の頂点位置座標の(y, z)座標が格納されます。

NITRO コマンド `VertexYZ` に相当します。

出力条件 以下の条件を全て満たす時に出力されます。

- ポリゴンが存在してかつ `vertex_style` が `direct`
- `x`座標が直前に設定した`x`座標値と同じ時

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <pos_xy xy="-1.500000 -1.500000"/>
  <pos_xz xz="2.500000 -2.500000"/>
  <pos_yz yz="3.500000 3.500000"/>
</primitive>
```

属性

表 4-33 <pos_yz>要素の属性

属性名	内容
yz	頂点位置座標の y、z 成分。 (-8.0 以上 8.0 未満の実数)×2

内容 なし

4.1.3.35 <pos_xyz>要素

説明 1組の頂点位置座標の(x,y,z)座標が格納されます。

NITRO コマンド Vertex に相当します。

出力条件 以下の条件を全て満たす時に出力されます。

- ポリゴンが存在してかつ vertex_style が direct
- <pos_xy>,<pos_xz>,<pos_yz>,<pos_diff>,<pos_s>の条件に該当しない

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <pos_xyz xyz="1.500000 1.500000 1.500000"/>
  <pos_xz xz="2.500000 -2.500000"/>
  <pos_yz yz="3.500000 3.500000"/>
</primitive>
```

属性

表 4-34 <pos_xyz>要素の属性

属性名	内容
xyz	頂点位置座標の x、y、z 成分。 (-8.0 以上 8.0 未満の実数)×3

内容 なし

4.1.3.36 <pos_diff>要素

説明 1組の頂点位置座標の(x,y,z)座標が格納されます。

NITRO コマンド VertexDiff に相当します。

出力条件 以下の条件を全て満たす時に出力されます。

- ポリゴンが存在してかつ vertex_style が direct
- 直前に設定した頂点位置座標と x,y,z それぞれの誤差が -0.125 以上 0.125 未満になる時

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <pos_xyz xyz="1.500000 1.500000 1.500000"/>
  <pos_diff xzz="0.000010 0.000007 0.000008"/>
  <pos_xyz xyz="0.000000 3.500000 3.500000"/>
</primitive>
```

属性

表 4-35 <pos_diff>要素の属性

属性名	内容
xyz	最後に設定した頂点位置座標からの差分。 (-0.125 以上 0.125 未満の実数) ×3

内容 なし

4.1.3.37 <pos_idx>要素

説明 頂点位置座標データ(=<vtx_pos_data>要素)へのインデックス番号が格納されます。

出力条件 ポリゴンが存在してかつ、vertex_style が index の時に出力されます。

出力例

```
<primitive index="1" type="triangles" vertex_size="3">
  <pos_idx idx="5"/>
  <pos_idx idx="7"/>
  <pos_idx idx="10"/>
</primitive>
```

属性

表 4-36 <pos_idx>要素の属性

属性名	内容
idx	頂点位置座標データへのインデックス番号。 (0以上の整数) ×1 <vtx_pos_data>要素の (idx ×3) 番目から3つの値を x,y,z として参照します。

内容 なし

4.1.3.38 <node_array>要素

説明 全てのノード情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<node_array size="2">
  <node index="0" name="null" ... (省略) ... >
    ... (省略) ...
  </node>
  <node index="1" name="pPlane1" ... (省略) ... >
    ... (省略) ...
  </node>
</node_array>
```

属性

表 4-37 <node_array>要素の属性

属性名	内容
size	要素内容に格納される<node>要素の数。 (1以上の整数)x1

内容 <node>要素を size 個格納します。
3DCGツール上の親子構造に従い、子を優先、兄弟はノード名のアルファベット順(a→z,0→9)
に格納されます。モデルのルートノードは必ず<node>要素の index0番目に出力されます。

4.1.3.39 <node>要素

説明 ノード情報が格納されます。

出力条件 必ず1つ以上出力されます。

出力例

```
<node_array size="1">
  <node index="0" name="pPlane1" kind="mesh"
    parent="-1" child="-1"
    brother_next="-1" brother_prev="-1"
    draw_mtx="on"
    scale_compensate="off"
    billboard="off"
    scale="1.000000 1.000000 1.000000"
    rotate="0.000000 0.000000 0.000000"
    translate="0.000000 0.000000 0.000000"
    visibility="on"
    display_size="1"
    vertex_size="4" polygon_size="1"
    triangle_size="0" quad_size="1"
    volume_min="-7.899902 0.000000 -7.899902"
    volume_max="7.899902 0.000000 7.899902"
    volume_r="11.172363"
  >
  <display index="0" material="0" polygon="0"/>
</node>
</node_array>
```

属性

表 4-38 <node>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)x1
name	ノード名。 文字列x1
kind	ノードタイプ。 文字列 [null / mesh / joint / chain / effector]のいずれか null : スルノード(Maya の Locator に相当)。 mesh : ポリゴンを持つノード。 joint : ジョイント(骨)ノード。 chain : ジョイントの一番親となるノード。 effector : ジョイントの一番末端となるノード。
parent	親ノードの番号。 (-1以上の整数)x1 -1の時は親を持たず、自身がルートノードを意味します。
child	一番下の子ノード(=自分の子供でかつ、名前が a からアルファベット順に並べて一番始めにくるノード)の番号。 (-1以上の整数)x1 -1の時は子ノードを持たず、自身が末端ノードを意味します。

brother_next	兄ノード(=同じ親ノードを持つ兄弟ノードを、名前で a からアルファベット順に並べて自分の次にくるノード)の番号。 (−1以上の整数)×1 −1の時は兄ノードを持ちません。
brother_prev	弟ノード(=同じ親ノードを持つ兄弟ノードを、名前で a からアルファベット順に並べて自分の一つ前にくるノード)の番号。 (−1以上の整数)×1 −1の時は弟ノードを持ちません。
draw_mtx	このノードの行列がポリゴンを描画するのに使われているかどうかのフラグ。 この値が off のノードは、ノード圧縮する時に削減される対象になります。 文字列 [off/on] のいずれか
scale_compensate ※scaling_rule が maya の時のみ出力	Maya で設定できるスケール計算のフラグ。 文字列 [off/on]のいずれか off: Maya の segment_scale_compensate 計算をしません。 on: Maya の segment_scale_compensate 計算をします。
billboard	ビルボード表示設定。 文字列 [off/on/y_on]のいずれか off: 通常表示。 on: 常にカメラの方を向くように表示します。 y_on: グローバルY軸のみカメラの方を向くようにして表示します。
scale	ノードの ScaleX,ScaleY,ScaleZ 値。 実数×3
rotate	ノードの RotateX,RotateY,RotateZ 値。 (−180.0 以上 180.0 未満の実数)×3
translate	ノードの TranslateX,TranslateY,TranslateZ 値。 実数×3
visibility	表示非表示の設定。 文字列 [off/on]のいずれか
display_size	子要素として出力される<display>要素の数。 (0以上の整数)×1
vertex_size ※ポリゴンを持つ(kind=mesh)時のみ存在	処理頂点数。 (3以上の整数)×1
polygon_size ※ポリゴンを持つ(kind=mesh)時のみ存在	ポリゴン数(=三角形ポリゴン数+四角形ポリゴン数)。 (1以上の整数)×1
triangle_size ※ポリゴンを持つ(kind=mesh)時のみ存在	三角形ポリゴン数。 (0以上の整数)×1
quad_size ※ポリゴンを持つ(kind=mesh)時のみ存在	四角形ポリゴン数。 (0以上の整数)×1
volume_min ※ポリゴンを持つ(kind=mesh)時のみ存在	このノードに属する全ポリゴンに外接する格子の左下奥のローカル座標値(x,y,z)。 実数×3 当たり判定などに利用できます。
volume_max ※ポリゴンを持つ(kind=mesh)時のみ存在	このノードに属する全ポリゴンに外接する格子の右上手前のローカル座標値(x,y,z)。 実数×3 当たり判定などに利用できます。
volume_r ※ポリゴンを持つ(kind=mesh)時のみ存在	このノードに属する全ポリゴンに外接する球の半径。 実数×1 当たり判定などに利用できます。

内容 <display>要素を display_size 個格納します。

4.1.3.40 <display>要素

説明 描画するマテリアル番号とポリゴン番号、及び描画順序を指定する描画優先度が格納されます。

出力条件 この要素が格納される<node>がポリゴンを持つ(kind=mesh)時のみ出力されます。

出力例

```

<node index="0" name="pPlane1" ... (省略) ...>
    <display index="0" material="0" polygon="0" priority="1"/>
    <display index="1" material="1" polygon="2" priority="0"/>
</node>

```

属性

表 4-39 <display>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)×1
material	描画に使用するマテリアル番号。 (0以上の整数)×1。 <material>要素を参照。
polygon	描画するポリゴン番号。 (0以上の整数)×1。 <polygon>要素を参照。
priority	<p>どの<display>要素から順に描画していくかを定める描画優先度。 (0以上 255 以下の整数)×1</p> <p>この値を使って、半透明ポリゴンやデカルポリゴンを表示する際の描画順序を制御することができます。</p> <p>1以上の場合、値の小さい<display>要素から順に描画します。</p> <p>0の場合、描画順序は指定せず、描画するタイミングは描画ルーチンに依存します。</p> <p>同じ値が複数ある場合、どちらを先に描画するかは描画ルーチンに依存します。</p> <p>※描画する順序を、各<node>内でのみ制御するか、モデル全体で制御するかは、描画ルーチンに依存します。詳しくは、「imd の補足」の”ポリゴンの描画優先度”を参照ください。</p>

内容 なし

4.1.3.41 <output_info>要素

説明 3DCGツールからファイル出力した際の出力情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```

<output_info vertex_size="962" polygon_size="629"
    triangle_size="620" quad_size="9"/>

```

属性

表 4-40 <output_info>要素の属性

属性名	内容
vertex_size	NITRO ジオメトリコマンドで出力する総処理頂点数。 (0以上の整数)×1
polygon_size	総ポリゴン数 (= 三角形ポリゴン数 + 四角形ポリゴン数)。 (0以上の整数)×1
triangle_size	三角形ポリゴン数。 (0以上の整数)×1
quad_size	四角形ポリゴン数。 (0以上の整数)×1

内容 なし

4.1.3.42 <ex_nns_3dme>要素

説明 3Dマテリアルエディタ専用の情報が格納されます。

出力条件 3Dマテリアルエディタからファイル出力した場合に1つ存在します。

出力例

```
<ex_nns_3dme>
  <ex_model_info
    scene_name="test_scene"
    ... (省略) ...
  />
  ... (省略) ...
  <ex_***>
    ... (省略) ...
  </ex_***>
</ex_nns_3dme>
```

属性	なし
内容	複数の子要素を格納します(<ex_nns_3dme>要素以下の子要素の名前の先頭は必ず”ex_”で始まります)。 ただし、<ex_nns_3dme>要素以下に格納される全ての子要素及び属性は3Dマテリアルエディタが利用するもので、TWL およびニンテンドーDS で3Dモデルを表示するのに必要な情報は格納されません。

<ex_nns_3dme>要素以下に格納される子要素及び属性については、TWL およびニンテンドーDS で3Dモデルを表示するのに関係する情報ではないため、本マニュアルでは記載しません。

4.1.4 imdの補足

- ノード構成について

出力されるノードの数は、3DCGツールから出力時に設定する「ノード圧縮設定」によって変化します。
ica や iva などのアニメーションを正常に再生するには imd と同じノード構成で同じノード圧縮設定で
ファイル出力する必要があります。

- ポリゴンとマテリアルの関係について

<polygon>要素に格納されるポリゴンをどの<material>で描画するかは、<display>要素にインデックス
番号で格納されます。

- 名前の文字数制限について

imd ファイルにはノード名、マテリアル名、テクスチャイメージ名、テクスチャパレット名、ポリゴングループ
名が半角英数字で格納されますが、その文字数に制限はありません。

ただし、NINTENDO TWL-System ライブラリを使う場合は、いずれの名前も16文字以内にお
く必要があります。

- テクスチャのファイル名について

テクスチャイメージ名にはテクスチャ画像ファイルの名前が格納されます。3DCG ツール上で同じテクス
チャ画像ファイルを複数のマテリアルから参照しても imd ファイル内では1つのイメージデータとして格
納されますが、ファイル名が同じでも別のパスにあるテクスチャ画像ファイルを用いると、同名のテクスチャ
イメージデータが複数作成され正常に表示できない場合があります。

- テクスチャのフォーマットについて

以下のテクスチャフォーマットに対応しています。

- ・4色パレットテクスチャ
- ・16色パレットテクスチャ
- ・256色パレットテクスチャ
- ・4x4テクセル圧縮テクスチャ
- ・A3I5 半透明テクスチャ
- ・A5I3 半透明テクスチャ
- ・ダイレクトテクスチャ

- 頂点カラーについて

- 1) ライト計算オフで表示する場合

頂点カラーが設定されたポリゴンを TWL およびニンテンドーDS 上でライト計算せずそのままの頂点カ
ラーで表示する場合は、3DCGツール上で対応するマテリアルをライトオフ設定にしてください。

この場合、<clr>要素は NITRO ジオメトリコマンド"Color"に相当します。

- 2) 頂点カラーとライト計算を併用する場合

<clr>要素を NITRO ジオメトリコマンド"Color"ではなく、"MaterialColor0"の Diffuse カラー値として
扱うことで、頂点毎に異なる色を設定し、かつライト計算を行う表現が可能です。

- NINTENDO TWL-System でのエンベロープ表記について

ポリゴンメッシュに複数のノード(スケルトンや骨ともいいます)を関連付け、ノードの動きに合わせて形状を変形させる手法を”スキニング(Skinning)”または”エンベロープ(Envelope)”などと呼びますが(3DCG ツール毎に呼び方が異なります)、NINTENDO TWL-System では **エンベロープ** と表記します。

NINTENDO TWL-System のエンベロープには、**フルウェイトエンベロープ** と **ウェイトエンベロープ** の2つがあります。

フルウェイトエンベロープ

フルウェイトエンベロープとは、頂点がいずれか1つのノードに対して100%の重み付けがされている状態を指します。ポリゴンメッシュを構成する全ての頂点がフルウェイトエンベロープの場合、そのモデルをフルウェイトエンベロープモデルと呼びます。

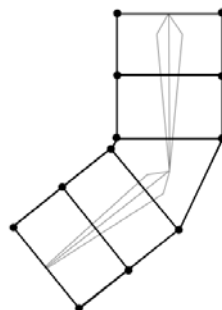


図 4-2 フルウェイトエンベロープモデルの例

ウェイトエンベロープ

ウェイトエンベロープとは、頂点が2つ以上のノードに対して合計100%の重み付けがされている状態を指します。ポリゴンメッシュを構成する頂点の中に、ウェイトエンベロープの頂点が1つでも存在する場合、ウェイトエンベロープモデルと呼びます。

ウェイトエンベロープモデルは、1つの頂点に対して重み付けをするノードの数を増やしたり、重み付けの値の種類を増やすことで、より滑らかな曲面を表現することができますが、それらの数を増やすほど描画時の計算量は増加します。

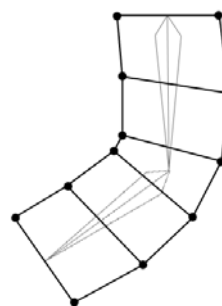


図 4-3 ウェイトエンベロープモデルの例

NINTENDO TWL-System で提供されるG3Dライブラリのウェイトエンベロープモデルへの対応状況については、「G3Dライブラリ リリースノート」を参照してください。

- 頂点位置座標を変換する行列について

頂点位置座標を変換する行列は、<mtx> → <mtx_list> → <matrix>要素の順にインデックスを参照します。

imd ファイルでは、ジオメトリコマンド PushMatrix / PopMatrix を使った行列計算方法だけでなく、CPU で行列計算を行い Store コマンドで行列スタックに書き込む計算方法にも対応しています。モデル全体を描画するのに TWL やニンテンドーDS の制限である行列スタック 31 段に収まらない場合にも対応するために、<mtx_list>要素を用意し、<mtx_prim>ごとに行列スタック 31 段を設定することもできる仕組みにしています。

ウェイトエンベロープ設定がされていない(=いずれか1つの行列に依存する)頂点は、参照する<matrix>要素内の属性 mtx_weight が1になります。このような頂点は、<matrix>要素内の属性 (node_idx) 番目のノードの行列を使って表示します。

ウェイトエンベロープ設定されている(=複数の行列に重みがかかっている)頂点は、参照する<matrix>要素内の属性 mtx_weight が重みのかかっているノードの数(2以上)になります。このような頂点は、さらに<matrix>から<weight>,<node_idx>要素を参照し、<weight>,<node_idx>要素の各 (envelope_head)番目から(mtx_weight)個の情報を使ってエンベロープ計算を行います。

例えば、envelope_head=2, mtx_weight=2 で、以下のような場合

```
<weight size="6">
```

```
    50  50  40  60  40  60
```

```
</weight>
```

```
<node_idx size="6">
```

```
    8  9  8  9  10  11
```

```
</node_idx>
```

index 番号 8 のノードに 40%、index 番号9のノードに 60%の割合で重み付けされたエンベロープ計算を行うことを意味します。

なお、エンベロープモデルの頂点位置座標は、単一のノードに 100% のウェイトが設定されている頂点(=フルウェイトエンベロープ)ならそのノードの空間でのローカル座標が出力され、複数のノードにウェイトが分散している頂点(=ウェイトエンベロープ)ならグローバル座標が出力されます。

法線ベクトルも同様に、フルウェイトエンベロープの時はそのローカル空間での方向ベクトルが、ウェイトエンベロープの時はグローバル空間での方向ベクトルが出力されます。

- 値の精度について

imd ファイルに出力される頂点位置座標値、法線ベクトル値、テクスチャ座標値はそれぞれ対応する NITRO ジオメトリコマンドの固定小数精度に合うように計算されています。

- ポリゴンの描画優先度

3DCGツール上のある一つのメッシュモデル(=imd の<node>要素に相当)に複数のマテリアルを関連付けると、そのメッシュモデルを構成するポリゴンは複数のポリゴン群(=imd の<polygon>要素に相当)に分かれます。そのような場合、3DCGツール上の各マテリアルに対して**描画優先度**(=imd ファイルの <display> 要素の属性 **priority** に相当)を設定することで、ポリゴン群を描画する順序を制御することができます。描画優先度は、半透明ポリゴン同士を重ね合わせて表示したり、デカルポリゴンを表示したりする際に利用します。

ここでいう「描画する順序」とは、TWL およびニンテンドーDS のジオメトリエンジンに描画コマンドを送る順番を意味します。TWL およびニンテンドーDS のハードウェアの仕様として、必ず不透明ポリゴンを描画した後に半透明ポリゴンの描画処理が行われるので、不透明ポリゴンより先に半透明ポリゴンを描画するように描画優先度を設定したとしても、TWL やニンテンドーDS 上では常に不透明ポリゴンから先に描画されるので注意してください。

描画する順序を管理したいポリゴン群に対して1以上の描画優先度を設定して順序付けを行ってください(値が小さいほど先に描画します)。

描画する順序を指定する必要があるポリゴン群に対しては、描画優先度を0に設定します。描画優先度が0のポリゴン群を描画するタイミングは描画ルーチンに依存します。また、描画優先度が同じポリゴン群が複数ある場合、どのポリゴン群から順に描画していくかは描画ルーチンに依存します。

<描画優先度>

- 0 : 描画順序を指定しない(=どのタイミングで描画するかは不定)。
1以上 : 値の小さいものから順に描画します。

描画する順序を各ノード内で制御するか、モデル全体で制御するかは、描画ルーチンに依存します。

例えば下図のようなモデルの場合、各ノード内で制御するか、モデル全体で制御するかによって描画順序が変わります。

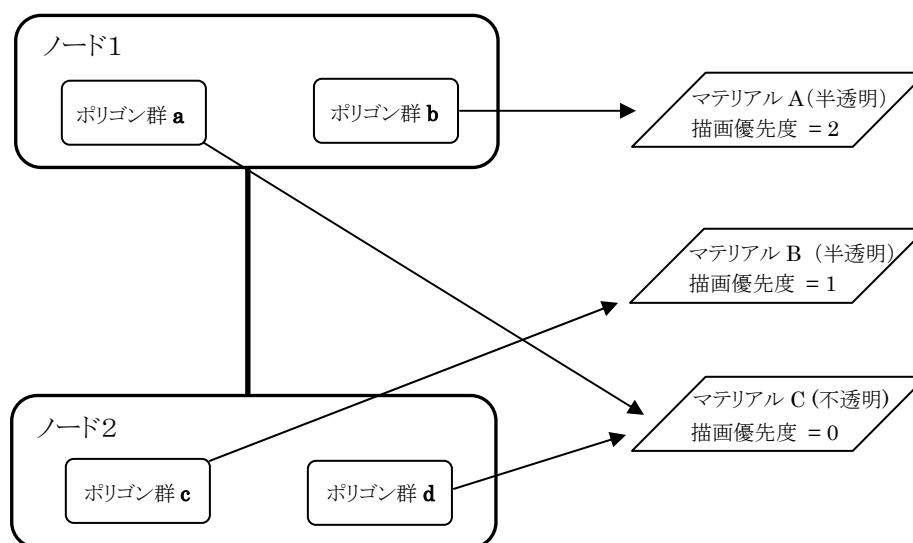


図 4-4 ポリゴンの描画優先度の例

各ノード(imd ファイルの<node>要素に相当)内で描画順序を制御する場合:

一般的な Push/Pop 方式の行列計算を用いた描画ルーチンを用いて、親のノードから順に描画する場合、ノード毎にポリゴン群を描画していくため、モデル全体としては

「(ポリゴン群 **a**) → **b** → (**a**) → (**d**) → **c** → (**d**)」(**a,d** はそれぞれいずれか一つ)の順に描画コマンドを送ることになります。

モデル(imd ファイル)全体で描画順序を制御する場合:

必要な行列計算を行った後にまとめて描画コマンドを送るような描画ルーチンの場合、描画順序をノードの親子関係に依存する必要がないため、モデル内の全てのポリゴン群を描画優先度に従って描画することができます。前述のようなノード構成の場合、

「(ポリゴン群 **a,d**) → **c** → (**a,d**) → **b** → (**a,d**)」(**a,d** はそれぞれいずれか一つ)

の順に描画コマンドを送ることになります。この時、ポリゴン群 **a** と **d** は描画優先度が共に0なので、どちらもどのタイミングで描画するかは描画ルーチンに依存します。

NINTENDO TWL-System で提供されるG3Dライブラリでは、後者の**モデル全体で描画順序を制御する描画ルーチン**になっています。

マテリアル圧縮機能とポリゴン群について

同一ノード内で、マテリアルの内容も描画優先度も同じ二つの異なるマテリアルが使われている場合、中間ファイル出力プラグインのマテリアル圧縮機能を使うと、マテリアルもポリゴン群もそれぞれ一つにまとめられます。

内容は同じで描画優先度が異なる二つのマテリアルの場合、マテリアルは一つにまとめられますが、ポリゴン群は描画優先度が異なるので一つにまとめられません。

また、異なるノード間では、マテリアルの内容と描画優先度が同じで、ノード削減に Unite and Combine Polygon が指定されている時のみ、ポリゴン群が一つにまとめられます。

- 環境/投影マッピングについて

TWL およびニンテンドーDS ではテクスチャ座標変換モードを **Normal** か **Vertex** ソースにすることで、環境マッピングや投影マッピングといった表現を行うことができます。

中間ファイルフォーマットでは、環境/投影マッピング時にどこからどのようにマッピングするかを指定するために、**<material>**要素内に属性 **tex_effect_mtx** を用意しています。

中間ファイルフォーマットでは、**tex_effect_mtx** の厳密な使い方は定義致しません。

どのように用いるかは描画ルーチン側で取り決めてください。

NINTENDO TWL-System が提供するG3Dライブラリでは、以下のように利用しています。

利用例1) 環境マッピング

テクスチャ座標変換モードを **Normal** ソースに設定し、以下の行列計算の結果をテクスチャ行列にセットしてジオメトリコマンドで法線ベクトルを送ることで、TWL およびニンテンドーDS 上で環境マッピングを表現します。

(ビュー座標系の法線ベクトルに変換する行列) *
(**tex_effect_mtx**) *
(マッピングの中心や大きさを合わせる行列(定数)) *
(マテリアルで指定されているテクスチャ **Scale & Rotate**)

tex_effect_mtx に回転行列やスケーリング行列を設定することで、テクスチャをある一定の方向から映りこませるような表現が可能です。

利用例2) 投影マッピング

テクスチャ座標変換モードを **Vertex** ソースに設定し、以下の行列計算の結果をテクスチャ行列にセットしてジオメトリコマンドで頂点座標を送ることで、TWL およびニンテンドーDS 上で投影マッピングを表現します。

(ワールド座標系の頂点座標に変換する行列) *
(**tex_effect_mtx**) *
(マッピングの中心や大きさを合わせる行列(定数))

tex_effect_mtx に回転行列やスケーリング行列を設定することで、テクスチャをある一定の方向から照射するような表現が可能です。

テクスチャ行列を工夫することによって、上記以外にも特殊なマッピング表現が可能です。

※<material>要素の属性 **tex_gen_st** について

テクスチャの環境/投影マッピングを表現するための情報として、中間ファイルフォーマット Ver1.5.0 まで定義しておりましたが、Ver1.6.0 より新たに追加した属性 **tex_effect_mtx** と内容が重複し、今後使用する予定もないため Ver1.6.0 より廃止しました。

5 アニメーション用中間ファイル

5.1 アニメーションファイルのデータ形式

5.1.1 データ形式の種類

NITRO 中間ファイルでは、5種類のアニメーション(キャラクターアニメーション、ビジュアルアニメーション、マテリアルカラーアニメーション、テクスチャパターンアニメーション、テクスチャSRTアニメーション)をサポートしています。

アニメーションのデータ形式としては**コマ形式**と**FVキー形式**の2種類の形式があり、各アニメーションファイルはそれぞれ最適な形式を採用しています。

表 5-1 アニメーションのデータ形式

データ形式	特徴	NITRO 中間ファイル
コマ形式	3DCGツール上のフレーム単位でアニメーションデータを出力する形式です。フレーム数が大きくなるほどデータサイズが大きくなりますが、データを等間隔に間引くことでデータサイズを縮小することもできます。(データを間引くほど精度は落ちますが、データ間を線形補間することで誤差を抑えることができます)	ica ima ita
FVキー形式	(Frame, Value)のセットを1つのキー・データとみなし、複数のキー・データの中から該当するキー・データを検索し、その Value 値を用いる形式です。データサイズはフレーム数とは関係なく、値の変化する度合いに比例します。 ※一般的な 3DCG ツールで用いられる (Frame, Value, Slope) をセットとしたエルミート補間のキー・データとは異なります。 将来的にエルミート形式のファイル出力を対応するかどうかは未定です。	itp iva

5.1.2 コマ形式

5.1.2.1 コマ形式のデータ構造

3DCGツール上の全フレームのデータを出力する形式(ここではコマ形式と呼びます)をベースに、さらに等間隔にデータを間引いてデータサイズを削減することもできる形式です。データサイズを小さくするために、データの間引き具合を3段階から選ぶこともできます。この時の選択肢を **frame_step** と呼びます。

frame_step=1 の時は間引きせず、3DCG ツールのフレーム数をそのまま出力します。

frame_step=2 では2フレーム毎のデータ出力します。データサイズは約 1/2 になります。

frame_step=4 では4フレーム毎のデータ出力します。データサイズは約 1/4 になります。

frame_step を大きくするほどアニメーションの品質が低下しますが、その分データサイズが小さくなります。

(TWL およびニンテンドーDS 上で除算でなくシフトで済ませるため、frame_step=3はありません)

データとしては、frame_step の倍数番目のフレームと、最後の余りの全フレームが出力されます。

例えば、3DCGツール上で16フレームのアニメーションをファイル出力する時は `frame_step` の設定によってそれぞれ以下のフレームデータが出力されます。

表 5-2 コマ形式で出力されるフレーム

フレーム	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
frame_step=1の時	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
frame_step=2の時	0		2		4		6		8		10		12		14	15
frame_step=4の時	0				4				8				12	13	14	15

データを間引いた部分、及び再生速度を変えて小数フレームを計算する場合は、前後のフレームデータを使い線形補間で値を求めます。

3DCGツールから中間ファイルを出力する時には、`frame_step=1 / 2 / 4 / auto` の中から選択します。

`1 / 2 / 4` の時は全てのアニメーションカーブを指定した `frame_step` に従ってデータ出力します。

`auto` の時はプラグイン内部でアニメーションカーブ単位で別途設定する許容誤差の範囲内で最適な `frame_step` を自動計算します

5.1.2.2 コマ形式の再生アルゴリズム

コマ形式データの再生アルゴリズムをC言語の関数形式で記述すると以下のようになります。

ここでは分かり易くするため、小数フレーム再生及び End フレームと Start フレームを補間する処理は考慮しません。

また Rotate データに 4096 のサインテーブルを適用した場合を想定しています。

コード 1 コマ形式再生関数のサンプル (Cソース形式)

```
const int frame_size;          // 総フレーム数
int      now_frame;            // 現在のフレーム
int getLinearInterpValue(const int frame_step, // 間引き具合
                        const int data_size,   // データサイズ
                        const int* value,      // 実データ配列
                        const bool angle_flag) // Rotate要素かどうかのフラグ
{
    int shift, ival, iframe_last_interp;
    int val0, val1, weight0, weight1, result;

    if(data_size == 1) // データが1つの時
        return value[0];

    if(frame_step == 1) // frame_stepが1の時
        return value[now_frame];

    shift = (frame_step==2) ? 1 : 2; // シフト値
    ival = now_frame >> shift;      // 現在のフレームに対応する実データ位置

    // 間引き部分 と 全フレーム部分のしきいとなるフレーム
    iframe_last_interp = ((frame_size - 1) >> shift) << shift;

    // しきいとなるフレーム以降の場合、実データ配列から直接返す
    if (now_frame >= iframe_last_interp)
        return value[ival + (now_frame - iframe_last_interp)];

    // しきいとなるフレームより前の場合は補間計算で求める
    val0 = value[ival ];
    val1 = value[ival + 1];
    weight1 = now_frame - (ival << shift);
    weight0 = frame_step - weight1;

    // ちょうどフレーム上にある時
    if (weight1 == 0)
        return val0;

    // 対象がRotate要素の時の特別処理
    else if( angle_flag && abs(val0 - val1) >= 2048){
        if(val0 > val1) val1 += 4096;
        else          val0 += 4096;
        result = ( (val0 * weight0 + val1 * weight1) >> shift );
        if(result > 4095) result -= 4096;
        return result;
    }

    // 通常の補間処理 -----
    else {
        return ((val0 * weight0 + val1 * weight1) >> shift);
    }
}
```

5.1.2.3 EndフレームからStartフレームへ補間する処理

ループ再生するようなアニメーションをゆっくり補間再生する時、最後のフレームから最初のフレームの状態にスムーズにつなげる必要があります。

コマ形式の中間ファイルの中には、End フレームから Start フレームへ補間するかどうかのフラグがあります。

例えば、100フレームのアニメーションデータがあるとします。このアニメーションデータは再生時に0～99.9999フレームまでを指定することができます(100フレーム目＝0フレーム目となります)。再生速度を1フレーム単位にするなら0, 1, 2・・・98, 99となりますが、再生速度を半分の0.5フレーム単位にすると0.0, 0.5, 1.0, 1.5・・・98.5, 99.0, 99.5となり、最終フレーム(99.0)を越える場合があります。仮に99.5フレーム目を計算する場合、End フレームから Start フレームへの補間をしない場合は99フレーム目(つまり最終フレーム)と同じ状態を返します。End フレームから Start フレームへ補間する場合は99フレーム目と0フレーム目を補間してつなげる必要があります。

5.1.3 FVキー形式

5.1.3.1 キー形式のデータ構造

(Frame, Value)のセットを1つのキー・データとみなし、複数のキー・データの中から該当するフレームを検索し、そのValue 値を用いる形式です。

例えば、3DCGツール上で16フレームの以下のようなアニメーションデータがある場合、中間ファイルにはこのようなデータ構成で出力されます。

表 5-3 3DCGツール上のデータ と FV キー・データ

3DCG ツール上のデータ

フレーム	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
値	0	0	0	0	1	1	1	2	2	2	2	1	1	1	0	0

FVキー・データ

Frame データ	0	4	7	11	14
Value データ	0	1	2	1	0

5.1.3.2 FVキー形式の再生アルゴリズム

Frame データ列から現在のフレームと同じもしくは直前となるフレームを検索し、それに対応する Value 値を使います。FVキー形式では補間処理及び End フレームと Start フレームを補間する処理は発生しません。

コード 2 FVキー形式関数のサンプル (Cソース形式)

```
int getFVkeyValue(const int* frames,          // フレーム番号列
                  const int* values,         // 実データ配列
                  const int data_size,       // データ数
                  const int iframe)         // 現在のフレーム
{
    // データが1つしかない時 -----
    if(data_size==1)
        return values[0];

    // データが2つ以上ある時 -----
    else {
        int LeftKey=0, RightKey=data_size-1, CenterKey;

        // 現在のフレームが1つ目のキー以前の時 -----
        if(iframe <= frames[LeftKey])
            return values[LeftKey];

        // 現在のフレームが最後のキー以降の時 -----
        if(iframe >= frames[RightKey])
            return values[RightKey];

        // いずれかのキーの間にある時 -----
        while(1){
            if((RightKey-LeftKey)==1 || RightKey==LeftKey){
                if(iframe < frames[RightKey])
                    return values[LeftKey];
                else
                    return values[RightKey];
            }
            CenterKey = (LeftKey + RightKey)/2;
            if(iframe <= frames[CenterKey])
                RightKey = CenterKey;
            else if(iframe >= frames[CenterKey])
                LeftKey = CenterKey;
        }
    }
}
```

5.2 キャラクタアニメーションファイル (ica)

5.2.1 icaの概要

キャラクタアニメーションファイル(ica: Intermediate Character Animation file)には、各ノードが持つ行列に Scale, Rotate, Translate 行列を乗算してモデルを動かすためのデータが出力されます。キャラクタアニメーションデータは、**コマ形式**で出力されます。

一般的な3DCGツールで使われているエルミート補間によるキー形式出力への対応は未定です。

5.2.2 icaの基本的な要素構成

<ica>	…ルート
<head>	…ドキュメント全体に関する情報
… (省略) …	
</head>	
<body>	…データ本体
<node_anm_info />	…アニメーションの設定情報
<node_scale_data />	…Scaleデータ
<node_rotate_data />	…Rotateデータ
<node_translate_data />	…Translateデータ
<node_anm_array>	…ノード配列
<node_anm />	…アニメーション参照情報
<scale_x />	…Scale X軸の参照情報
<scale_y />	…Scale Y軸の参照情報
<scale_z />	…Scale Z軸の参照情報
<rotate_x />	…Rotate X軸の参照情報
<rotate_y />	…Rotate Y軸の参照情報
<rotate_z />	…Rotate Z軸の参照情報
<translate_x />	…Translate X軸の参照情報
<translate_y />	…Translate Y軸の参照情報
<translate_z />	…Translate Z軸の参照情報
<node_anm />	
</node_anm_array>	
</body>	
</ica>	

5.2.3 icaの要素及び属性説明

5.2.3.1 <node_anm_info>要素

説明 キャラクタアニメーション全体に関する情報が格納されます。

出力条件 必ず出力されます。

出力例

```
<node_anm_info
    frame_size="90"
    scaling_rule="standard"
    magnify="1.000000"
    tool_start_frame="1"
    tool_end_frame="90"
    interpolation="linear"
    interp_end_to_start="off"
    compress_node="none" node_size="31 31"
    frame_step_mode="auto"
    tolerance_scale="0.100000"
    tolerance_rotate="0.100000"
    tolerance_translate="0.100000"
/>
```

属性

表 5-4 <node_anm_info>要素の属性

属性名	内容
frame_size	フレーム数。 (1以上の整数)x1
scaling_rule	スケールの計算方法。 文字列[standard / maya / si3d]のいずれか ノード構成が親(a)、子(b)、孫(c)とすると、 standard : 一般的な行列計算。(Sc * Rc * Tc * Sb * Rb * Tb * Sa * Ra * Ta) maya : maya の Segment Scale Compensate を考慮した計算。 si3d : Softimage 方式の行列計算。 各計算方法については ノードの行列の計算方法 をご覧ください。
magnify	3DCGツールから imd を出力する際にモデル全体にかけた倍率。 実数値x1 中間ファイル出力時の設定を確認するための情報です。 中間ファイル内の頂点位置座標・Translate 値等のデータに反映されていますので、TWL およびニンテンドーDS 上でこの値を使う必要はありません。
tool_start_frame	3DCGツールからファイル出力する際に指定した StartFrame。 整数x1 (ica 出力状態確認用)
tool_end_frame	3DCGツールからファイル出力する際に指定した EndFrame。 整数x1 (ica 出力状態確認用)
interpolation	アニメーションの再生で、小数フレームでの値を求めて再生するときの方法を指定します。 文字列 [frame / linear]のいずれか frame : 小数フレームを切り捨てて整数フレームとして扱って再生します。 linear : 小数フレームを切り捨てずにそのまま扱い、小数フレームでの値を線形補間を用いて求めて再生します。 「G3Dライブラリで再生する際の注意」 も合わせてご覧ください。
interp_end_to_start ※ interpolation が linear の時のみ出力	最後のフレームから最初のフレームの間で線形補間を行うかどうかのフラグ。 文字列 [off / on]のいずれか off : 最後のフレーム以降は変化しない。 on : 線形補間を行う。
compress_node	出力されたノードの圧縮状態。 文字列 [none / cull / merge / unite / unite_combine]のいずれか none : ツール上と同じノード構成で出力します。 cull : モデルの表示に必要なでないノードを削除して出力します。 merge : cull の処理に加え、合成できるノードの行列を合成して出力します。 unite : 1つのノードにまとめます (主に地形データ用)。 unite_combine : unite の処理に加え、さらに同じ<material>要素で表示する<polygon>要素を1つにまとめます。 (ica 出力状態確認用)
node_size	未圧縮時と圧縮後のノード数。 (1以上の整数)x2 compress_node=none の時は、未圧縮時の数が2つ並びます。

frame_step_mode	データの間引き指定。 文字列 [1 / 2 / 4 / auto]のいずれか 1: 3DCGツール上の全フレームを出力。 2: 2フレーム毎に出力 (=アニメーションデータサイズ約 1/2)。 4: 4フレーム毎に出力 (=アニメーションデータサイズ約 1/4)。 auto: アニメーションカーブごとに最適な間引き具合を自動判断。
tolerance_scale	Scale データを間引く際の許容誤差。 (0.0 以上の実数)×1
tolerance_rotate	Rotate データを間引く際の許容誤差。 (0.0 以上の実数)×1
tolerance_translate	Translate データを間引く際の許容誤差。 (0.0 以上の実数)×1

内容 なし

5.2.3.2 <node_scale_data>要素

説明 Scale アニメーションデータが格納されます。

出力条件 必ず出力されます。

出力例

```
<node_scale_data size="3">
    1.000000 1.500000 2.300000
</node_scale_data>
```

属性

表 5-5 <node_scale_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 実数 (=Scale 値) を size 個格納します。

5.2.3.3 <node_rotate_data>要素

説明 Rotate アニメーションデータが格納されます。

出力条件 必ず出力されます。

出力例

```
<node_rotate_data size="1722">
    0.000000 124.377441 0.366699 0.363525 0.354492
    ... (省略) ...
    0.135742 0.162354
</node_rotate_data>
```

属性

表 5-6 <node_rotate_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 実数(=Rotate 値:degree)を size 個格納します。

5.2.3.4 <node_translate_data>要素

説明 Translate アニメーションデータが格納されます。

出力条件 必ず出力されます。

出力例

```
<node_translate_data size="110">
  0.027100 0.037598 0.049072 0.061035 0.073730
  ... (省略) ...
  -0.003662 -0.033691 4.086426 -0.246338 -1.791992
</node_translate_data>
```

属性

表 5-7 <node_translate_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 実数(=Translate 値)を size 個格納します。

5.2.3.5 <node_anm_array>要素

説明 アニメーションの参照情報が配列で格納されます。

出力条件 必ず出力されます。

出力例

```
<node_anm_array size="2">
  <node_anm index="0">
    ... (省略) ...
  </node_anm>
  <node_anm index="1">
    ... (省略) ...
  </node_anm>
</node_anm_array>
```

属性

表 5-8 <node_anm_array>要素の属性

属性名	内容
size	要素内容に格納される<node_anm>要素の数。 (1以上の整数)×1

内容 <node_anm>要素を size 個格納します。

5.2.3.6 <node_anm>要素

説明 アニメーションの参照情報が格納されます。

出力条件 必ず1つ以上出力されます。

出力例

```
<node_anm_array size="2">
  <node_anm index="0">
    <scale_x frame_step="1" data_size="1" data_head="0"/>
    <scale_y frame_step="1" data_size="1" data_head="0"/>
    <scale_z frame_step="1" data_size="1" data_head="0"/>
    <rotate_x frame_step="1" data_size="1" data_head="0"/>
    <rotate_y frame_step="1" data_size="1" data_head="0"/>
    <rotate_z frame_step="1" data_size="1" data_head="0"/>
    <translate_x frame_step="1" data_size="90" data_head="0"/>
    <translate_y frame_step="1" data_size="1" data_head="90"/>
    <translate_z frame_step="1" data_size="1" data_head="91"/>
  </node_anm>
  <node_anm index="1">
    ... (省略) ...
  </node_anm>
</node_anm_array>
```

属性

表 5-9 <node_anm>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)x1

内容 <scale_x>,<scale_y>,<scale_z>,<rotate_x>,<rotate_y>,<rotate_z>,
<translate_x>,<translate_y>,<translate_z>を1つずつこの順序で格納します。

5.2.3.7 <scale_x>,<scale_y>,<scale_z>要素

説明 Scale アニメーションの参照情報が格納されます。

出力条件 必ず出力されます。

出力例

```
<node_anm index="0">
  <scale_x frame_step="1" data_size="1" data_head="0"/>
  <scale_y frame_step="1" data_size="1" data_head="0"/>
  <scale_z frame_step="2" data_size="45" data_head="0"/>
  <rotate_x frame_step="1" data_size="1" data_head="0"/>
  <rotate_y frame_step="1" data_size="1" data_head="1"/>
  <rotate_z frame_step="1" data_size="1" data_head="2"/>
  <translate_x frame_step="1" data_size="90" data_head="0"/>
  <translate_y frame_step="1" data_size="1" data_head="90"/>
  <translate_z frame_step="1" data_size="1" data_head="91"/>
</node_anm>
```

属性

表 5-10 <scale_x>,<scale_y>,<scale_z>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数 [1 / 2 / 4]のいずれか 1: 全フレームをデータ出力。 2: 2フレーム毎にデータ出力(=アニメーションデータサイズ約 1/2)。 4: 4フレーム毎にデータ出力(=アニメーションデータサイズ約 1/4)。

data_size	アニメーションのデータサイズ。(1以上の整数)×1 <node_scale_data>を参照します。
data_head	データの先頭位置。(0以上の整数)×1 <node_scale_data>を参照します。

内容 なし

5.2.3.8 <rotate_x>,<rotate_y>,<rotate_z>要素

説明 Rotate アニメーションの参照情報が格納されます。

出力条件 必ず出力されます。

出力例

```
<node_anm index="0">
  <scale_x frame_step="1" data_size="1" data_head="0"/>
  <scale_y frame_step="1" data_size="1" data_head="0"/>
  <scale_z frame_step="2" data_size="45" data_head="0"/>
  <rotate_x frame_step="1" data_size="1" data_head="0"/>
  <rotate_y frame_step="1" data_size="1" data_head="1"/>
  <rotate_z frame_step="1" data_size="1" data_head="2"/>
  <translate_x frame_step="1" data_size="90" data_head="0"/>
  <translate_y frame_step="1" data_size="1" data_head="90"/>
  <translate_z frame_step="1" data_size="1" data_head="91"/>
</node_anm>
```

属性

表 5-11 <rotate_x>,<rotate_y>,<rotate_z>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数 [1 / 2 / 4]のいずれか 1: 全フレームをデータ出力。 2: 2フレーム毎にデータ出力(=アニメーションデータサイズ約 1/2)。 4: 4フレーム毎にデータ出力(=アニメーションデータサイズ約 1/4)。
data_size	アニメーションのデータサイズ。(1以上の整数)×1 <node_rotate_data>を参照します。
data_head	データの先頭位置。(0以上の整数)×1 <node_rotate_data>を参照します。

内容 なし

5.2.3.9 <translate_x>,<translate_y>,<translate_z>要素

説明 Translate アニメーションの参照情報が格納されます。

出力条件 必ず出力されます。

出力例

```

<node_anm index="0">
  <scale_x frame_step="1" data_size="1" data_head="0"/>
  <scale_y frame_step="1" data_size="1" data_head="0"/>
  <scale_z frame_step="2" data_size="45" data_head="0"/>
  <rotate_x frame_step="1" data_size="1" data_head="0"/>
  <rotate_y frame_step="1" data_size="1" data_head="0"/>
  <rotate_z frame_step="1" data_size="1" data_head="0"/>
  <translate_x frame_step="1" data_size="90" data_head="0"/>
  <translate_y frame_step="1" data_size="1" data_head="90"/>
  <translate_z frame_step="1" data_size="1" data_head="91"/>
</node_anm>

```

属性

表 5-12 <translate_x>,<translate_y>,<translate_z>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数 [1 / 2 / 4]のいずれか 1: 全フレームをデータ出力。 2: 2フレーム毎にデータ出力(=アニメーションデータサイズ約 1/2)。 4: 4フレーム毎にデータ出力(=アニメーションデータサイズ約 1/4)。
data_size	アニメーションのデータサイズ。(1以上の整数)×1 <node_translate_data>を参照します。
data_head	データの先頭位置。(0以上の整数)×1 <node_translate_data>を参照します。

内容 なし

5.2.4 icaの補足

- imd との関連付けについて
ica を正しく再生するには、同じノード構成の imd ファイルが必要です。
imd と ica は、imd の<node>要素と ica の<node_anm>要素のインデックスで対応付けます。
同じモデルでも、3DCGツールからファイル出力する際にノード圧縮設定の違いなどによってノード構成が変わった imd には正しくアニメーションを反映できません。
- G3Dライブラリで再生する際の注意

現在 TWL-System からリリースされている G3D ライブラリでは、interpolation を linear にする事で小数フレームに対する再生が可能ですが、interpolation が frame の時と比べ CPU の計算処理が多少増加します。

interpolation を linear に設定する必要がないようなアニメーションデータについては、CPUの負荷軽減のため interpolation を frame にして出力するように注意してください。

5.3 ビジビリティアニメーションファイル (iva)

5.3.1 ivaの概要

ビジビリティアニメーションファイル(iva: Intermediate Visibility Animation file)には、各ノード単位で表示非表示を切り替えるアニメーションのためのデータが出力されます。ビジビリティアニメーションデータは、「フレーム番号」と「表示非表示の状態」の2つを1組とした**FVキー形式**で出力されます。

5.3.2 ivaの基本的な要素構成

<iva>	…ルート
<head>	…ドキュメント全体に関する情報
… (省略) …	
</head>	
<body>	…データ本体
<visibility_info />	…アニメーションの設定情報
<visibility_data>	…アニメーションデータ
<frame_idx />	…フレームデータ
<visibility />	…表示非表示データ
</visibility_data>	
<visibility_anm_array>	…アニメーション配列
<visibility_anm />	…アニメーションの参照情報
</visibility_anm_array>	
</body>	
</iva>	

5.3.3 ivaの要素及び属性説明

5.3.3.1 <visibility_info>要素

説明 ビジビリティアニメーション全体に関する情報が格納されます。

出力条件 必ず出力されます。

出力例

```
<visibility_info
  frame_size="55"
  tool_start_frame="1"
  tool_end_frame="55"
  compress_node="none" node_size="10 10"
/>
```

属性

表 5-13 <visibility_info>要素の属性

属性名	内容
frame_size	フレーム数。 (1以上の整数)×1

tool_start_frame	3DCGツールからファイル出力する際に指定した StartFrame。 整数×1 (iva 出力状態確認用)
tool_end_frame	3DCGツールからファイル出力する際に指定した EndFrame。 整数×1 (iva 出力状態確認用)
compress_node	出力されたノードの圧縮状態。 文字列 [none / cull / merge / unite / unite_combine]のいずれか none : ツール上と同じノード構成で出力します。 cull : モデルの表示に必要なでないノードを削除して出力します。 merge : cull の処理に加え、合成できるノードの行列を合成して出力します。 unite : 1つのノードにまとめます(主に地形データ用)。 unite_combine : unite の処理に加え、さらに同じ<material>要素で表示する<polygon>要素を1つにまとめます。 (iva 出力状態確認用)
node_size	ノード未圧縮時 と 圧縮後のノード数。 (1以上の整数)×2 compress_node=none の時は、未圧縮時の数が2つ並びます。 (iva 出力状態確認用)

内容 なし

5.3.3.2 <visibility_data>要素

説明 フレームデータとビジビリティデータを格納します。

出力条件 必ず出力されます。

出力例

```
<visibility_data>
  <frame_idx size="28">
    0 0 4 9 0 9 14 0 14 19
    0 19 24 0 24 29 0 29 34 0
    34 39 0 39 44 0 44 49
  </frame_idx>
  <visibility size="28">
    1 1 0 1 1 0 1 1 0 1
    1 0 1 1 0 1 1 0 1 1
    0 1 1 0 1 1 0 1
  </visibility>
</visibility_data>
```

属性 なし

内容 <frame_idx>,<visibility>要素を1つずつこの順序で格納します。

5.3.3.3 <frame_idx>要素

説明 フレームデータが格納されます。

データの数と並びは<visibility>要素と対になります。

出力条件 必ず出力されます。

出力例

```

<visibility_data>
  <frame_idx size="28">
    0 0 4 9 0 9 14 0 14 19
    0 19 24 0 24 29 0 29 34 0
    34 39 0 39 44 0 44 49
  </frame_idx>
  <visibility size="28">
    1 1 0 1 1 0 1 1 0 1
    1 0 1 1 0 1 1 0 1 1
    0 1 1 0 1 1 0 1
  </visibility>
</visibility_data>

```

属性

表 5-14 <frame_idx>要素の属性

属性名	内容
size	要素内容に格納するデータの数。 (1以上の整数)x1

内容 0以上の整数(=フレーム番号)を size 個格納します。

5.3.3.4 <visibility>要素

説明 ビジビリティデータ(表示非表示の状態。0=非表示、1=表示)が格納されます。
データの数と並びは<frame_idx>要素と対になります。

出力条件 必ず出力されます。

出力例

```

<visibility_data>
  <frame_idx size="28">
    0 0 4 9 0 9 14 0 14 19
    0 19 24 0 24 29 0 29 34 0
    34 39 0 39 44 0 44 49
  </frame_idx>
  <visibility size="28">
    1 1 0 1 1 0 1 1 0 1
    1 0 1 1 0 1 1 0 1 1
    0 1 1 0 1 1 0 1
  </visibility>
</visibility_data>

```

属性

表 5-15 <visibility>要素の属性

属性名	内容
size	要素内容に格納するデータの数。 (1以上の整数)x1

内容 0(=非表示)か1(=表示)の整数を size 個格納します。

5.3.3.5 <visibility_anm_array>要素

説明 ノードの数だけビジビリティアニメーションの参照情報を格納します。

出力条件 必ず出力されます。

出力例

```
<visibility_anm_array size="10">
  <visibility_anm index="0" data_size="1" data_head="0"/>
  ... (省略) ...
  <visibility_anm index="9" data_size="3" data_head="25"/>
</visibility_anm_array>
```

属性

表 5-16 <visibility_anm_array>要素の属性

属性名	内容
size	要素内容に格納する<visibility_anm>要素の数。 (1以上の整数)×1

内容 <visibility_anm>要素を size 個格納します。

※<visibility_anm>要素の順番と数は、反映する imd ファイルの<node>要素に対応します。

5.3.3.6 <visibility_anm>要素

説明 ノードごとのビジビリティアニメーションの参照情報を格納します。

出力条件 必ず出力されます。

出力例

```
<visibility_anm_array size="10">
  <visibility_anm index="0" data_size="1" data_head="0"/>
  <visibility_anm index="1" data_size="3" data_head="1"/>
  <visibility_anm index="2" data_size="3" data_head="4"/>
  <visibility_anm index="3" data_size="3" data_head="7"/>
  <visibility_anm index="4" data_size="3" data_head="10"/>
  <visibility_anm index="5" data_size="3" data_head="13"/>
  <visibility_anm index="6" data_size="3" data_head="16"/>
  <visibility_anm index="7" data_size="3" data_head="19"/>
  <visibility_anm index="8" data_size="3" data_head="22"/>
  <visibility_anm index="9" data_size="3" data_head="25"/>
</visibility_anm_array>
```

属性

表 5-17 <visibility_anm>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)×1 imd のノード番号に対応します。
data_size	アニメーションのデータサイズ。 (1以上の整数)×1 <frame_idx>及び<visibility>要素を参照します。
data_head	データの先頭位置。 (0以上の整数)×1 <frame_idx>及び<visibility>要素を参照します。

内容 なし

5.3.4 ivaの補足

- imd との関連付けについて

iva を正しく再生するには、同じノード構成の imd ファイルが必要です。

imd と iva は、imd の<node>要素と ica の<visibility_anm>要素のインデックスで対応付けます。

同じモデルでも、3DCGツールからファイル出力する際にノード圧縮設定の違いなどによってノード構成が変わった imd には正しくアニメーションを反映できません。

5.4 マテリアルカラーアニメーションファイル (ima)

5.4.1 imaの概要

マテリアルカラーアニメーションファイル(ima: Intermediate Material Color Animation file)には、各マテリアルの diffuse、ambient、specular、emission の色、及びポリゴンアルファを変化させるアニメーションのデータが出力されます。

マテリアルカラーアニメーションデータは**コマ形式**で出力されます。

一般的な3DCGツールで使われているエルミート補間によるキー形式出力への対応は未定です。

5.4.2 imaの基本的な要素構成

<ima>	…ルート
<head>	…ドキュメント全体に関する情報
… (省略) …	
</head>	
<body>	…データ本体
<mat_color_info />	…アニメーションの設定情報
<mat_color_data />	…カラーデータ
<mat_color_anm_array>	…アニメーション配列
<mat_color_anm>	…アニメーションの参照情報
<diffuse_r>	…ディフューズ赤成分の参照情報
<diffuse_g>	…ディフューズ緑成分の参照情報
<diffuse_b>	…ディフューズ青成分の参照情報
<ambient_r>	…アンビエント赤成分の参照情報
<ambient_g>	…アンビエント緑成分の参照情報
<ambient_b>	…アンビエント青成分の参照情報
<specular_r>	…スペキュラ赤成分の参照情報
<specular_g>	…スペキュラ緑成分の参照情報
<specular_b>	…スペキュラ青成分の参照情報
<emission_r>	…エミッション赤成分の参照情報
<emission_g>	…エミッション緑成分の参照情報
<emission_b>	…エミッション青成分の参照情報
<polygon_alpha>	…ポリゴンアルファの参照情報
</mat_color_anm>	
</mat_color_anm_array>	
</body>	
</ima>	

5.4.3 imaの要素及び属性説明

5.4.3.1 <mat_color_info>要素

説明 マテリアルカラーアニメーション全体に関する情報が出力されます。

出力条件 必ず出力されます。。

出力例

```
<mat_color_info
    frame_size="100"
    tool_start_frame="1"
    tool_end_frame="100"
    interpolation="linear"
    interp_end_to_start="off"
    compress_material="on" material_size="1 1"
    frame_step_mode="auto"
    tolerance_color="1"
/>
```

属性

表 5-18 <mat_color_info>要素の属性

属性名	内容
frame_size	フレーム数。 (1以上の整数)×1
tool_start_frame	3DCGツールからファイル出力する際に指定した StartFrame。 整数×1 (ima 出力状態確認用)
tool_end_frame	3DCGツールからファイル出力する際に指定した EndFrame。 整数×1 (ima 出力状態確認用)
interpolation	アニメーションの再生で、小数フレームでの値を求めて再生するときの方法を指定します。 文字列 [frame / linear]のいずれか frame : 小数フレームを切り捨てて整数フレームとして扱って再生します。 linear : 小数フレームを切り捨てずにそのまま扱い、小数フレームでの値を線形補間を用いて求めて再生します。 「G3Dライブラリで再生する際の注意」 も合わせてご覧ください。
interp_end_to_start ※ interpolation が linear の時のみ出力	最後のフレームから最初のフレームの間で線形補間を行うかどうかのフラグ。 文字列 [off / on]のいずれか off : 最後のフレーム以降は変化しない。 on : 線形補間を行う。
compress_material	マテリアルの圧縮状態。 文字列[off / on]のいずれか off : ツール上と同じマテリアル数を出力。 on : 設定内容が全て一致するマテリアルがあれば名前の若い方のみを出力。 (ima 出力状態確認用)
material_size	未圧縮時と圧縮時のマテリアル数。 (0以上の整数)×2 compress_material=off の時は、未圧縮時の数が2つ並びます。 (ima 出力状態確認用)
frame_step_mode	データの間引き指定。 文字列[1 / 2 / 4 / auto]のいずれか 1 : 3DCGツール上の全フレームを出力。 2 : 2フレーム毎に出力(=データサイズ約 1/2)。 4 : 4フレーム毎に出力(=データサイズ約 1/4)。 auto : アニメーションカーブごとに最適な間引きを自動で計算します。
tolerance_color	カラーデータを間引く際の許容誤差。 (0以上 31以下の整数)×1

内容 なし

5.4.3.2 <mat_color_data>要素

説明 カラーデータが格納されます。

出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。

出力例

```
<mat_color_data size="85">
  25 23 21 18 16 14 11 9 7 5
  3 2 1 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 2
  5 7 10 13 15 18 20 22 23 24
  25 25 24 23 21 19 17 14 12 9
  7 4 2 1 1 0 1 0 0 0
  0 0 0 0 0 0 0 0 0 1
  2 4 6 8 10 12 14 17 19 21
  23 24 24 25 31
</mat_color_data>
```

属性

表 5-19 <mat_color_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 0以上31以下の整数値を size 個格納します。

※TWL およびニンテンドーDS の仕様に従い、R, G, B, A 各濃度を0以上31以下の整数に正規化して出力します。

5.4.3.3 <mat_color_anm_array>要素

説明 マテリアルカラーアニメーションの数だけ参照情報が格納されます。

出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。

出力例

```
<mat_color_anm_array size="2">
  <mat_color_anm index="0" material_name="lambert2">
    ... (省略) ...
  </mat_color_anm>
  <mat_color_anm index="1" material_name="lambert3">
    ... (省略) ...
  </mat_color_anm>
</mat_color_anm_array>
```

属性

表 5-20 <mat_color_anm_array>要素の属性

属性名	内容
size	要素内容に格納される<mat_color_anm>要素の数。 (1以上の整数)×1

内容 <mat_color_anm>要素を size 個格納します。

<mat_color_anm>は、material_name のアルファベット順(a→z,0→9)に出力します。

5.4.3.4 <mat_color_anm>要素

- 説明 マテリアルカラーアニメーションの参照情報が格納されます。
- 出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。
- 出力例

```
<mat_color_anm_array size="1">
  <mat_color_anm index="0" material_name="lambert2">
    <diffuse_r frame_step="4" data_size="28" data_head="0"/>
    <diffuse_g frame_step="4" data_size="28" data_head="28"/>
    <diffuse_b frame_step="4" data_size="28" data_head="56"/>
    <ambient_r frame_step="1" data_size="1" data_head="13"/>
    <ambient_g frame_step="1" data_size="1" data_head="13"/>
    <ambient_b frame_step="1" data_size="1" data_head="13"/>
    <specular_r frame_step="1" data_size="1" data_head="13"/>
    <specular_g frame_step="1" data_size="1" data_head="13"/>
    <specular_b frame_step="1" data_size="1" data_head="13"/>
    <emission_r frame_step="1" data_size="1" data_head="13"/>
    <emission_g frame_step="1" data_size="1" data_head="13"/>
    <emission_b frame_step="1" data_size="1" data_head="13"/>
    <polygon_alpha frame_step="1" data_size="1" data_head="84"/>
  </mat_color_anm>
</mat_color_anm_array>
```

属性

表 5-21 <mat_color_anm>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)x1
material_name	反映するマテリアル名。 文字列x1

- 内容 <diffuse_r>,<diffuse_g>,<diffuse_b>,<ambient_r>,<ambient_g>,<ambient_b>,
 <specular_r>,<specular_g>,<specular_b>,<emission_r>,<emission_g>,<emission_b>,
 <polygon_alpha>要素を1つずつこの順序で格納します。

5.4.3.5 <diffuse_r>,<diffuse_g>,<diffuse_b>要素

- 説明 ディフューズ Red,Green,Blue 成分の参照情報が格納されます。
- 出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。
- 出力例

```
<mat_color_anm index="0" material_name="lambert2">
  <diffuse_r frame_step="4" data_size="28" data_head="0"/>
  <diffuse_g frame_step="4" data_size="28" data_head="28"/>
  <diffuse_b frame_step="4" data_size="28" data_head="56"/>
  <ambient_r frame_step="1" data_size="1" data_head="13"/>
  <ambient_g frame_step="1" data_size="1" data_head="13"/>
  <ambient_b frame_step="1" data_size="1" data_head="13"/>
  <specular_r frame_step="1" data_size="1" data_head="13"/>
  <specular_g frame_step="1" data_size="1" data_head="13"/>
  <specular_b frame_step="1" data_size="1" data_head="13"/>
  <emission_r frame_step="1" data_size="1" data_head="13"/>
  <emission_g frame_step="1" data_size="1" data_head="13"/>
  <emission_b frame_step="1" data_size="1" data_head="13"/>
  <polygon_alpha frame_step="1" data_size="1" data_head="84"/>
</mat_color_anm>
```

属性

表 5-22 <diffuse_r>,<diffuse_g>,<diffuse_b>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。(1以上の整数)×1 <mat_color_data>を参照します。
data_head	データの先頭位置。(0以上の整数)×1 <mat_color_data>を参照します。

内容 なし

5.4.3.6 <ambient_r>,<ambient_g>,<ambient_b>要素

説明 アンビエント Red,Green,Blue 成分の参照情報が格納されます。

出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。

出力例

```
<mat_color_anm index="0" material_name="lambert2">
  <diffuse_r frame_step="4" data_size="28" data_head="0" />
  <diffuse_g frame_step="4" data_size="28" data_head="28" />
  <diffuse_b frame_step="4" data_size="28" data_head="56" />
  <ambient_r frame_step="1" data_size="1" data_head="13" />
  <ambient_g frame_step="1" data_size="1" data_head="13" />
  <ambient_b frame_step="1" data_size="1" data_head="13" />
  <specular_r frame_step="1" data_size="1" data_head="13" />
  <specular_g frame_step="1" data_size="1" data_head="13" />
  <specular_b frame_step="1" data_size="1" data_head="13" />
  <emission_r frame_step="1" data_size="1" data_head="13" />
  <emission_g frame_step="1" data_size="1" data_head="13" />
  <emission_b frame_step="1" data_size="1" data_head="13" />
  <polygon_alpha frame_step="1" data_size="1" data_head="84" />
</mat_color_anm>
```

属性

表 5-23 <ambient_r>,<ambient_g>,<ambient_b>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。(1以上の整数)×1 <mat_color_data>を参照します。
data_head	データの先頭位置。(0以上の整数)×1 <mat_color_data>を参照します。

内容 なし

5.4.3.7 <specular_r>,<specular_g>,<specular_b>要素

説明 スペキュラ Red,Green,Blue 成分の参照情報が格納されます。

出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。

出力例

```
<mat_color_anm index="0" material_name="lambert2">
  <diffuse_r frame_step="4" data_size="28" data_head="0"/>
  <diffuse_g frame_step="4" data_size="28" data_head="28"/>
  <diffuse_b frame_step="4" data_size="28" data_head="56"/>
  <ambient_r frame_step="1" data_size="1" data_head="13"/>
  <ambient_g frame_step="1" data_size="1" data_head="13"/>
  <ambient_b frame_step="1" data_size="1" data_head="13"/>
  <specular_r frame_step="1" data_size="1" data_head="13"/>
  <specular_g frame_step="1" data_size="1" data_head="13"/>
  <specular_b frame_step="1" data_size="1" data_head="13"/>
  <emission_r frame_step="1" data_size="1" data_head="13"/>
  <emission_g frame_step="1" data_size="1" data_head="13"/>
  <emission_b frame_step="1" data_size="1" data_head="13"/>
  <polygon_alpha frame_step="1" data_size="1" data_head="84"/>
</mat_color_anm>
```

属性

表 5-24 <specular_r>,<specular_g>,<specular_b>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。 (1以上の整数)×1 <mat_color_data>を参照します。
data_head	データの先頭位置。 (0以上の整数)×1 <mat_color_data>を参照します。

内容 なし

5.4.3.8 <emission_r>,<emission_g>,<emission_b>要素

説明 エミッション Red,Green,Blue 成分の参照情報が格納されます。

出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。

出力例

```
<mat_color_anm index="0" material_name="lambert2">
  <diffuse_r frame_step="4" data_size="28" data_head="0"/>
  <diffuse_g frame_step="4" data_size="28" data_head="28"/>
  <diffuse_b frame_step="4" data_size="28" data_head="56"/>
  <ambient_r frame_step="1" data_size="1" data_head="13"/>
  <ambient_g frame_step="1" data_size="1" data_head="13"/>
  <ambient_b frame_step="1" data_size="1" data_head="13"/>
  <specular_r frame_step="1" data_size="1" data_head="13"/>
  <specular_g frame_step="1" data_size="1" data_head="13"/>
  <specular_b frame_step="1" data_size="1" data_head="13"/>
  <emission_r frame_step="1" data_size="1" data_head="13"/>
  <emission_g frame_step="1" data_size="1" data_head="13"/>
  <emission_b frame_step="1" data_size="1" data_head="13"/>
  <polygon_alpha frame_step="1" data_size="1" data_head="84"/>
</mat_color_anm>
```

属性

表 5-25 <emission_r>,<emission_g>,<emission_b>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。 (1以上の整数)×1 <mat_color_data>を参照します。
data_head	データの先頭位置。 (0以上の整数)×1 <mat_color_data>を参照します。

内容 なし

5.4.3.9 <polygon_alpha>要素

説明 ポリゴンアルファの参照情報が格納されます。

出力条件 マテリアルカラーアニメーションが設定されている時だけ出力されます。

出力例

```
<mat_color_anm index="0" material_name="lambert2">
  <diffuse_r frame_step="4" data_size="28" data_head="0"/>
  <diffuse_g frame_step="4" data_size="28" data_head="28"/>
  <diffuse_b frame_step="4" data_size="28" data_head="56"/>
  <ambient_r frame_step="1" data_size="1" data_head="13"/>
  <ambient_g frame_step="1" data_size="1" data_head="13"/>
  <ambient_b frame_step="1" data_size="1" data_head="13"/>
  <specular_r frame_step="1" data_size="1" data_head="13"/>
  <specular_g frame_step="1" data_size="1" data_head="13"/>
  <specular_b frame_step="1" data_size="1" data_head="13"/>
  <emission_r frame_step="1" data_size="1" data_head="13"/>
  <emission_g frame_step="1" data_size="1" data_head="13"/>
  <emission_b frame_step="1" data_size="1" data_head="13"/>
  <polygon_alpha frame_step="1" data_size="1" data_head="84"/>
</mat_color_anm>
```

属性

表 5-26 <polygon_alpha>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。 (1以上の整数)×1 <mat_color_data>を参照します。
data_head	データの先頭位置。 (0以上の整数)×1 <mat_color_data>を参照します。

内容 なし

5.4.4 imaの補足

- imd との関連付けについて

ima を再生するには、imd ファイルが必要です。

imd と ima は、「imd の<material>要素の属性 name」と「ima の<mat_color_anm>要素の属性 material_name」で対応付けます。imd 内に該当するマテリアル名が存在しない場合は反映されません。

- G3Dライブラリで再生する際の注意

現在 TWL-System からリリースされている G3D ライブラリでは、ima の小数フレームに対する線形補間 (interpolation = linear の状態) には対応していません。interpolation を linear に設定しても、frame と同じ再生になります。

将来的に G3D ライブラリが小数フレームに対する線形補間に対応すれば、interpolation = linear にする事で小数フレームに対する再生も可能になりますが、この場合は CPU での計算処理が増加します。

interpolation を linear に設定する必要がないようなアニメーションデータについては、上記の CPU の処理速度の観点から、interpolation = frame にして出力するように注意してください。

5.5 テクスチャパターンアニメーションファイル (itp)

5.5.1 itpの概要

テクスチャパターンアニメーションファイル(itp: Intermediate Texture Pattern Animation file)には、テクスチャのイメージ及びパレットを差し替えるアニメーションのためのデータが出力されます。テクスチャパターンアニメーションデータは、「フレーム番号」と「イメージ番号」と「パレット番号」の3つを1組とした**FVキー形式**で出力されます。

5.5.2 itpの基本的な要素構成

<itp>	…ルート
<head>	…ドキュメント全体に関する情報
… (省略) …	
</head>	
<body>	…データ本体
<tex_pattern_info />	…アニメーションの設定情報
<tex_pattern_list_data>	…イメージ及びパレットリスト情報
<image_name />	…イメージリスト情報
<palette_name />	…パレットリスト情報
</tex_pattern_list_data>	
<tex_pattern_data>	…アニメーションデータ
<frame_idx />	…フレームデータ
<image_idx />	…イメージデータ
<palette_idx />	…パレットデータ
</tex_pattern_data>	
<tex_pattern_anm_array>	…アニメーション配列
<tex_pattern_anm />	…アニメーションの参照情報
</tex_pattern_anm_array>	
</body>	
</itp>	

5.5.3 itpの要素及び属性説明

5.5.3.1 <tex_pattern_info>要素

説明 テクスチャパターンアニメーション全体に関する情報が出力されます。

出力条件 必ず出力されます。

出力例

```
<tex_pattern_info
  frame_size="60"
  tool_start_frame="1"
  tool_end_frame="60"
  compress_material="on" material_size="1 1"
/>
```

属性

表 5-27 <tex_pattern_info>要素の属性

属性名	内容
frame_size	フレーム数。 (1以上の整数)×1
tool_start_frame	3DCGツールからファイル出力する際に指定した StartFrame。 整数×1 (itp 出力状態確認用)
tool_end_frame	3DCGツールからファイル出力する際に指定した EndFrame。 整数×1 (itp 出力状態確認用)
compress_material	マテリアルの圧縮状態。 文字列[off / on]のいずれか off : ツール上と同じマテリアル数を出力。 on : 設定内容が全て一致するマテリアルがあれば名前の若い方のみを出力。 (itp 出力状態確認用)
material_size	未圧縮時と圧縮時のマテリアル数。 (0以上の整数)×2 compress_material=off の時は、未圧縮時の数が2つ並びます。 (itp 出力状態確認用)

内容 なし

5.5.3.2 <tex_pattern_list_data>要素

説明 テクスチャパターンアニメーションで使うイメージ名とパレット名のリストが格納されます。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_pattern_list_data image_size="3" palette_size="1">
  <image_name index="0" name="p1" />
  <image_name index="1" name="p2" />
  <image_name index="2" name="p3" />
  <palette_name index="0" name="p1_index" />
</tex_pattern_list_data>
```

属性

表 5-28 <tex_pattern_list_data>要素の属性

属性名	内容
image_size	このファイル内で使うイメージの数。 (1以上の整数)×1
palette_size	このファイル内で使うパレットの数。 (0以上の整数)×1

内容 <image_name>要素を image_size 個、<palette_name>要素を palette_size 個格納します。
イメージ名・パレット名はそれぞれアルファベット順(a→z,0→9)に並びます。

5.5.3.3 <image_name>要素

説明 イメージ名が格納されます。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_pattern_list_data image_size="3" palette_size="1">
  <image_name index="0" name="p1"/>
  <image_name index="1" name="p2"/>
  <image_name index="2" name="p3"/>
  <palette_name index="0" name="p1_index"/>
</tex_pattern_list_data>
```

属性

表 5-29 <image_name>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)×1
name	イメージ名。 文字列×1

内容 なし

5.5.3.4 <palette_name>要素

説明 パレット名が格納されます。

出力条件 ダイレクトカラーテクスチャ以外のテクスチャを使ってテクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_pattern_list_data image_size="3" palette_size="1">
  <image_name index="0" name="p1"/>
  <image_name index="1" name="p2"/>
  <image_name index="2" name="p3"/>
  <palette_name index="0" name="p1_index"/>
</tex_pattern_list_data>
```

属性

表 5-30 <palette_name>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)×1
name	イメージ名。 文字列×1

内容 なし

5.5.3.5 <tex_pattern_data>要素

説明 アニメーションを再生するための実データが格納されます。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```

<tex_pattern_data>
  <frame_idx size="3">
    0 20 40
  </frame_idx>
  <image_idx size="3">
    0 1 2
  </image_idx>
  <palette_idx size="3">
    0 0 0
  </palette_idx>
</tex_pattern_data>

```

属性 なし

内容 <frame_idx> , <img_idx> , <plt_idx> 要素をこの順序で1つずつ格納します。

5.5.3.6 <frame_idx>要素

説明 テクスチャパターンアニメーションのフレーム番号が格納されます。
格納するデータの数と並びは、<image_idx>,<palette_idx>と対になります。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```

<tex_pattern_data>
  <frame_idx size="3">
    0 20 40
  </frame_idx>
  <image_idx size="3">
    0 1 2
  </image_idx>
  <palette_idx size="3">
    0 0 0
  </palette_idx>
</tex_pattern_data>

```

属性

表 5-31 <frame_idx>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 0以上の整数(=フレーム番号)を size 個格納します。

5.5.3.7 <image_idx>要素

説明 テクスチャパターンアニメーションのイメージ番号が格納されます。
格納するデータの数と並びは、<frame_idx>,<palette_idx>と対になります。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```

<tex_pattern_data>
  <frame_idx size="3">
    0 20 40
  </frame_idx>
  <image_idx size="3">
    0 1 2
  </image_idx>
  <palette_idx size="3">
    0 0 0
  </palette_idx>
</tex_pattern_data>

```

属性

表 5-32 <image_idx>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 0以上の整数(=イメージ番号)を size 個格納します。

5.5.3.8 <palette_idx>要素

説明 テクスチャパターンアニメーションのパレット番号が格納されます。
格納するデータの数と並びは、<frame_idx>,<image_idx>と対になります。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```

<tex_pattern_data>
  <frame_idx size="3">
    0 20 40
  </frame_idx>
  <image_idx size="3">
    0 1 2
  </image_idx>
  <palette_idx size="3">
    0 0 0
  </palette_idx>
</tex_pattern_data>

```

属性

表 5-33 <palette_idx>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 -1 以上の整数(=パレット番号)を size 個格納します。
(パレットを使わないテクスチャフォーマットの場合は-1 が入ります)

5.5.3.9 <tex_pattern_anm_array>要素

説明 テクスチャパターンアニメーションの参照データを格納します。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_pattern_anm_array size="1">
  <tex_pattern_anm index="0"
    material_name="lambert2" data_size="3" data_head="0"/>
</tex_pattern_anm_array>
```

属性

表 5-34 <tex_pattern_anm_array>要素の属性

属性名	内容
size	要素内容に格納される<tex_pattern_anm>の数。 (1以上の整数)×1

内容 <tex_pattern_anm>要素を size 個格納します。

5.5.3.10 <tex_pattern_anm>要素

説明 テクスチャパターンアニメーションの参照情報が格納されます。

出力条件 テクスチャパターンアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_pattern_anm_array size="1">
  <tex_pattern_anm index="0"
    material_name="lambert2" data_size="3" data_head="0"/>
</tex_pattern_anm_array>
```

属性

表 5-35 <tex_pattern_anm>要素の属性

属性名	内容
index	通し番号。(0以上の整数)×1
material_name	反映する imd 内のマテリアル名。 文字列×1
data_size	アニメーションのデータサイズ。(1以上の整数)×1 <frame_idx>,<image_idx>,<palette_idx>要素を参照します。
data_head	データの先頭位置。(0以上の整数)×1 <frame_idx>,<image_idx>,<palette_idx>要素を参照します。

内容 なし

5.5.4 itpの補足

- imd との関連付けについて
itp を再生するには、imd ファイルが必要です。
imd と itp は、それぞれ以下のとおりに対応付けします。
マテリアル: 「imd の<material>要素の属性 name」と「itp の<tex_pattern_anm>要素の属性 material_name」で対応付けます。
イメージ: 「imd の<tex_image>要素の属性 name」と「itp の<image_name>要素の属性 name」で対応付けます。
パレット: 「imd の<tex_palette>要素の属性 name」と「itp の<palette_name>要素の属性 name」で対応付けます。
imd 内に該当するマテリアル・イメージ・パレットが存在しない場合は反映されません。
- イメージサイズとパレットの色数について
テクスチャパターンアニメーションでイメージもしくはパレットを差し替える場合、イメージの縦横サイズ、パレットのフォーマット及び色数を揃えておく必要があります。イメージサイズやパレットの色数が異なるテクスチャが混在すると正常に表示できない場合があります。

5.6 テクスチャSRTアニメーションファイル (ita)

5.6.1 itaの概要

テクスチャSRTアニメーションファイル(ita: Intermediate Texture SRT Animation file)には、テクスチャ行列に Scale、Rotate、Translate 行列を乗算して、テクスチャを拡大縮小したり、回転したり、スクロールさせるアニメーションのためのデータが出力されます。テクスチャSRTアニメーションデータは、**コマ形式**で出力されます。

一般的な3DCGツールで使われているエルミート補間によるキー形式出力への対応は未定です。

5.6.2 itaの基本的な要素構成

<code><ita></code>	…ルート
<code><head></code>	…ドキュメント全体に関する情報
… (省略) …	
<code></head></code>	
<code><body></code>	…データ本体
<code><tex_srt_info /></code>	…アニメーションの設定情報
<code><tex_scale_data /></code>	…テクスチャScaleデータ
<code><tex_rotate_data /></code>	…テクスチャRotateデータ
<code><tex_translate_data /></code>	…テクスチャTranslateデータ
<code><tex_srt_anm_array></code>	…アニメーション配列
<code><tex_srt_anm /></code>	…アニメーションの参照情報
<code><tex_scale_s /></code>	…テクスチャScale S方向の参照情報
<code><tex_scale_t /></code>	…テクスチャScale T方向の参照情報
<code><tex_rotate /></code>	…テクスチャRotate の参照情報
<code><tex_translate_s /></code>	…テクスチャTranslate S方向の参照情報
<code><tex_translate_t /></code>	…テクスチャTranslate T方向の参照情報
<code></tex_srt_anm /></code>	
<code></tex_srt_anm_array></code>	
<code></body></code>	
<code></ita></code>	

5.6.3 itaの要素及び属性説明

5.6.3.1 `<tex_srt_info>`要素

説明	テクスチャSRTアニメーション全体に関する情報が出力されます。
出力条件	必ず出力されます。
出力例	


```

<tex_srt_info
    frame_size="150"
    tool_start_frame="1"
    tool_end_frame="150"
    interpolation="linear"
    interp_end_to_start="off"
    tex_matrix_mode="maya"
    compress_material="on" material_size="1 1"
    frame_step_mode="auto"
    tolerance_tex_scale="0.01"
    tolerance_tex_rotate="0.5"
    tolerance_tex_translate="0.5"
/>

```

属性

表 5-36 <tex_srt_info>要素の属性

属性名	内容
frame_size	フレーム数。 (1以上の整数)x1
tool_start_frame	3DCGツールからファイル出力する際に指定した StartFrame。 整数x1 (ita 出力状態確認用)
tool_end_frame	3DCGツールからファイル出力する際に指定した EndFrame。 整数x1 (ita 出力状態確認用)
interpolation	アニメーションの再生で、小数フレームでの値を求めて再生するときの方法を指定します。 文字列 [frame / linear]のいずれか frame : 小数フレームを切り捨てて整数フレームとして扱って再生します。 linear : 小数フレームを切り捨てずにそのまま扱い、小数フレームでの値を線形補間を用いて求めて再生します。 「G3Dライブラリで再生する際の注意」 も合わせてご覧ください。
interp_end_to_start ※interpolation が linear の時のみ出力	最後のフレームから最初のフレームの間に線形補間を行うかどうかのフラグ。 文字列 [off / on]のいずれか off : 最後のフレーム以降は変化しない。 on : 線形補間を行う。
tex_matrix_mode	テクスチャ行列の計算方法。 文字列[maya / si3d / xsi / 3dsmax]のいずれか maya : Maya の計算方法。 si3d : SOFTIMAGE 3D の計算方法。 xsi : SOFTIMAGE XSI の計算方法。 3dsmax : 3ds max の計算方法。 各計算方法については テクスチャ行列の計算方法 をご覧ください。
compress_material	マテリアルの圧縮状態。 文字列[off / on]のいずれか off : ツール上と同じマテリアル数を出力。 on : 設定内容が全て一致するマテリアルがあれば名前の若い方のみを出力。 (ita 出力状態確認用)
material_size	未圧縮時と圧縮後のマテリアル数。 (0以上の整数)x2 compress_material=off の時は、未圧縮時の数が2つ並びます。 (ita 出力状態確認用)
frame_step_mode	データの間引き指定。 文字列[1 / 2 / 4 / auto]のいずれか 1 : 3DCGツール上の全フレームを出力。 2 : 2フレーム毎に出力(=データサイズ約 1/2)。 4 : 4フレーム毎に出力(=データサイズ約 1/4)。 auto : アニメーションカーブごとに最適な間引きを自動で計算します。
tolerance_tex_scale	テクスチャ Scale データを間引く際の許容誤差。 (0.0 以上の実数)x1
tolerance_tex_rotate	テクスチャ Rotate データを間引く際の許容誤差。 (0.0 以上の実数)x1
tolerance_tex_translate	テクスチャ Translate データを間引く際の許容誤差。 (0.0 以上の実数)x1

内容 なし

5.6.3.2 <tex_scale_data>要素

説明 テクスチャ Scale アニメーションの実データが格納されます。

出力条件 テクスチャSRTアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_scale_data size="78">
  1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
  1.000000 1.040039 1.199951 1.360107 1.520020 1.679932 1.840088 2.000000
  ... (省略) ...
  2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 1.000000
</tex_scale_data>
```

属性

表 5-37 <tex_scale_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 実数(=Scale 値)を size 個格納します。

5.6.3.3 <tex_rotate_data>要素

説明 テクスチャ Rotate アニメーションの実データが格納されます。

出力条件 テクスチャ SRT アニメーションが設定されている時だけ出力されます。

出力例

```
<tex_rotate_data size="150">
  0.000000 7.346924 14.693848 22.040771 29.387695 36.734619 44.081543
  51.428467 58.775635 66.122559 73.469482 80.816406 88.163330 95.510254
  ... (省略) ...
  -36.734619 -29.387695 -22.040771 -14.693848 -7.346924 0.000000 0.000000
  0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
</tex_rotate_data>
```

属性

表 5-38 <tex_rotate_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 -180.0 以上 180.0 未満の実数(=Rotate 値:degree)を size 個格納します。

5.6.3.4 <tex_translate_data>要素

説明 テクスチャの Translate アニメーションの実データが格納されます。

出力条件 テクスチャ SRT アニメーションが設定されている時だけ出力されます。

出力例

```
<tex_translate_data size="300">
  0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
  20.479980 23.040039 25.600098 28.159912 30.719971 33.280029 35.840088
  ... (省略) ...
  64.000000 64.000000 64.000000 64.000000 64.000000 64.000000 64.000000
</tex_translate_data>
```

属性

表 5-39 <tex_translate_data>要素の属性

属性名	内容
size	要素内容に格納されるデータの数。 (1以上の整数)×1

内容 実数(=Translate 値)を size 個格納します。

5.6.3.5 <tex_srt_anm_array>要素

説明 テクスチャSRTアニメーションの参照データを格納します。

出力条件 テクスチャSRTアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_srt_anm_array size="1">
  <tex_srt_anm index="0" material_name="lambert2">
    <tex_scale_s frame_step="4" data_size="15" data_head="0"/>
    <tex_scale_t frame_step="4" data_size="15" data_head="15"/>
    <tex_rotate frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_s frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_t frame_step="1" data_size="60" data_head="60"/>
  </tex_srt_anm>
</tex_srt_anm_array>
```

属性

表 5-40 <tex_srt_anm_array>要素の属性

属性名	内容
size	要素内容に格納される<tex_srt_anm>の数。 (1以上の整数)×1

内容 <tex_srt_anm>要素を size 個格納します。

5.6.3.6 <tex_srt_anm>要素

説明 テクスチャSRTアニメーションの参照情報が格納されます。

出力条件 テクスチャSRTアニメーションが設定されている時だけ出力されます。

出力例

```

<tex_srt_anm_array size="1">
  <tex_srt_anm index="0" material_name="lambert2">
    <tex_scale_s frame_step="4" data_size="15" data_head="0"/>
    <tex_scale_t frame_step="4" data_size="15" data_head="15"/>
    <tex_rotate frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_s frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_t frame_step="1" data_size="60" data_head="60"/>
  </tex_srt_anm>
</tex_srt_anm_array>

```

属性

表 5-41 <tex_srt_anm>要素の属性

属性名	内容
index	通し番号。 (0以上の整数)x1
material_name	反映するマテリアル名。 文字列x1

内容 <tex_scale_s>,<tex_scale_t>,<tex_rotate>,<tex_translate_s>,<tex_translate_t>要素を1つずつこの順番に格納します。

5.6.3.7 <tex_scale_s>,<tex_scale_t>要素

説明 テクスチャ Scale アニメーションの参照情報が格納されます。

出力条件 テクスチャSRTアニメーションが設定されている時だけ出力されます。

出力例

```

<tex_srt_anm_array size="1">
  <tex_srt_anm index="0" material_name="lambert2">
    <tex_scale_s frame_step="4" data_size="15" data_head="0"/>
    <tex_scale_t frame_step="4" data_size="15" data_head="15"/>
    <tex_rotate frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_s frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_t frame_step="1" data_size="60" data_head="60"/>
  </tex_srt_anm>
</tex_srt_anm_array>

```

属性

表 5-42 <tex_scale_s>,<tex_scale_t>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。 (1以上の整数)x1 <tex_scale_data>を参照します。
data_head	データの先頭位置。 (0以上の整数)x1 <tex_scale_data>を参照します。

内容 なし

5.6.3.8 <tex_rotate>要素

説明 テクスチャ Rotate アニメーションの参照情報が格納されます。

出力条件 テクスチャSRTアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_srt_anm_array size="1">
  <tex_srt_anm index="0" material_name="lambert2">
    <tex_scale_s frame_step="4" data_size="15" data_head="0"/>
    <tex_scale_t frame_step="4" data_size="15" data_head="15"/>
    <tex_rotate frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_s frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_t frame_step="1" data_size="60" data_head="60"/>
  </tex_srt_anm>
</tex_srt_anm_array>
```

属性

表 5-43 <tex_rotate>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。 4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。 (1以上の整数)x1 <tex_rotate_data>を参照します。
data_head	データの先頭位置。 (0以上の整数)x1 <tex_rotate_data>を参照します。

内容 なし

5.6.3.9 <tex_translate_s>, <tex_translate_t>要素

説明 テクスチャ Translate アニメーションの参照情報が格納されます。

出力条件 テクスチャSRTアニメーションが設定されている時だけ出力されます。

出力例

```
<tex_srt_anm_array size="1">
  <tex_srt_anm index="0" material_name="lambert2">
    <tex_scale_s frame_step="4" data_size="15" data_head="0"/>
    <tex_scale_t frame_step="4" data_size="15" data_head="15"/>
    <tex_rotate frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_s frame_step="1" data_size="60" data_head="0"/>
    <tex_translate_t frame_step="1" data_size="60" data_head="60"/>
  </tex_srt_anm>
</tex_srt_anm_array>
```

属性

表 5-44 <tex_translate_s>,<tex_translate_t>要素の属性

属性名	内容
frame_step	データの間引き具合。 整数[1 / 2 / 4]のいずれか 1: 全フレーム出力。 2: 2フレーム毎に出力(=データサイズ約 1/2)。

	4: 4フレーム毎に出力(=データサイズ約 1/4)。
data_size	アニメーションのデータサイズ。 (1以上の整数)×1 <tex_translate_data>を参照します。
data_head	データの先頭位置。 (0以上の整数)×1 <tex_translate_data>を参照します。

内容 なし

5.6.4 itaの補足

- imd との関連付けについて
ita を再生するには、imd ファイルが必要です。
imd と ita は、「imd の<material>要素の属性 name」と「ita の<tex_srt_anm>要素の属性 material_name」で対応付けます。imd 内に該当するマテリアル名が存在しない場合は反映されません。
- テクスチャ行列の計算方法について
テクスチャ行列の計算方法は各3DCGツールごとに異なります。
<tex_srt_info>の tex_matrix_mode で指定されるツールの計算方法を用いてください。

- G3Dライブラリで再生する際の注意

現在 TWL-System からリリースされている G3D ライブラリでは、ita の小数フレームに対する線形補間 (interpolation = linear の状態)には対応していません。interpolation を linear に設定しても、frame と同じ再生になります。

将来的に G3D ライブラリが小数フレームに対する線形補間に対応すれば、interpolation = linear にする事で小数フレームに対する再生も可能になりますが、この場合は CPU での計算処理が増加します。

interpolation を linear に設定する必要がないようなアニメーションデータについては、上記の CPU の処理速度の観点から、interpolation = frame にして出力するように注意してください。

6 グラフィックスシーン用中間ファイル

6.1 グラフィックスシーンデータファイル (isd)

6.1.1 isdの概要

グラフィックスシーンデータファイル (isd: Intermediate graphics Scene Data file) には、カメラやライト・トゥーンテーブル・フォグなど、主に TWL およびニンテンドーDS 上で3Dモデルを表示する際にフレーム単位で設定する情報が格納されます。

6.1.2 isdの基本的な要素構成

<code><isd></code>	…ルート
<code><head></code>	…ドキュメント全体に関する情報
… (省略) …	
<code></head></code>	
<code><body></code>	…データ本体
<code><camera /></code>	…カメラ情報
<code><light></code>	…ライト情報
<code><light0 /></code>	…ライト0情報
<code><light1 /></code>	…ライト1情報
<code><light2 /></code>	…ライト2情報
<code><light3 /></code>	…ライト3情報
<code></light></code>	
<code><shininess_table /></code>	…鏡面反射輝度テーブルデータ
<code><toon_highlight></code>	…トゥーンハイライト情報
<code><toon_table /></code>	…トゥーンテーブルデータ
<code></toon_highlight></code>	
<code><fog></code>	…フォグ情報
<code><fog_table /></code>	…フォグテーブルデータ
<code></fog></code>	
<code><alpha_test /></code>	…アルファテスト情報
<code><alpha_blending /></code>	…アルファブレンディング情報
<code><y_sorting /></code>	…半透明ポリゴンのYソーティング情報
<code><edge_marking></code>	…エッジマーキング情報
<code><edge_color /></code>	…エッジカラーデータ
<code></edge_marking></code>	
<code><antialias /></code>	…アンチエイリアス情報
<code><render_1_pixel_depth /></code>	…1ピクセルポリゴンの表示境界デプス情報
<code><clear_color /></code>	…クリアカラー情報
<code></body></code>	
<code></isd></code>	

6.1.3 isdの要素及び属性説明

6.1.3.1 <camera>要素

説明 カメラの情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<camera
  position="0.000000 0.000000 0.000000"
  lookout="0.000000 0.000000 0.000000"
  twist="0.000000"
  projection="perspective"
  depth_buffer="w"
  scale_w="1.000000"
  near="10.000000"
  far="10000.000000"
  perspective_fovy="30.000000"
  perspective_aspect="1.333333"
  ortho_width="256.000000"
  ortho_height="192.000000"
/>
```

属性

表 6-1 <camera>要素の属性

属性名	内容
position	カメラの位置(=視点)座標。 実数×3
lookat	カメラの注視点座標。 実数×3
twist	カメラの視線方向に対する水平方向の傾き角度。 (-180.0 以上 180.0 以下の 実数)×1 視線方向に対して何度回転しているかを示します。 正数が左方向へ、負数が右方向へと回転します。 ※現在、3Dマテリアルエディタから出力した時、この値は常に 0.0 になります。これはカメラ行列に指定するカメラの上方向のベクトルが(0.0, 1.0, 0.0)の状態に相当します。
projection	射影方法。 [perspective / ortho] のいずれか perspective : 透視射影で表示します。 ortho : 正射影で表示します。
depth_buffer	デプスバッファリングの方法を指定します。 [z / w] のいずれか z : Z値によるバッファリングを選択します。 w : W値によるバッファリングを選択します。 NITRO コマンド SwapBuffers のデプスバッファリング選択フラグに相当します。
scale_w	射影行列にかける scaleW 値。 0より大きい実数×1
near	ニアクリップ面までの距離。 (0.0 以上の実数)×1
far	ファークリップ面までの距離。 (0.0 以上の実数)×1
perspective_fovy	透視射影時の縦(y)方向の視界角度(画角)。 (0.0 以上 180.0 以下の実数)×1
perspective_aspect	透視射影時の縦横比。 (0.0 以上の実数)×1
ortho_width	正射影時の射影幅。 (0.0 以上の実数)×1
ortho_height	正射影時の射影高さ。 (0.0 以上の実数)×1

<camera>要素に格納される属性を図で示すと次の図のようになります。

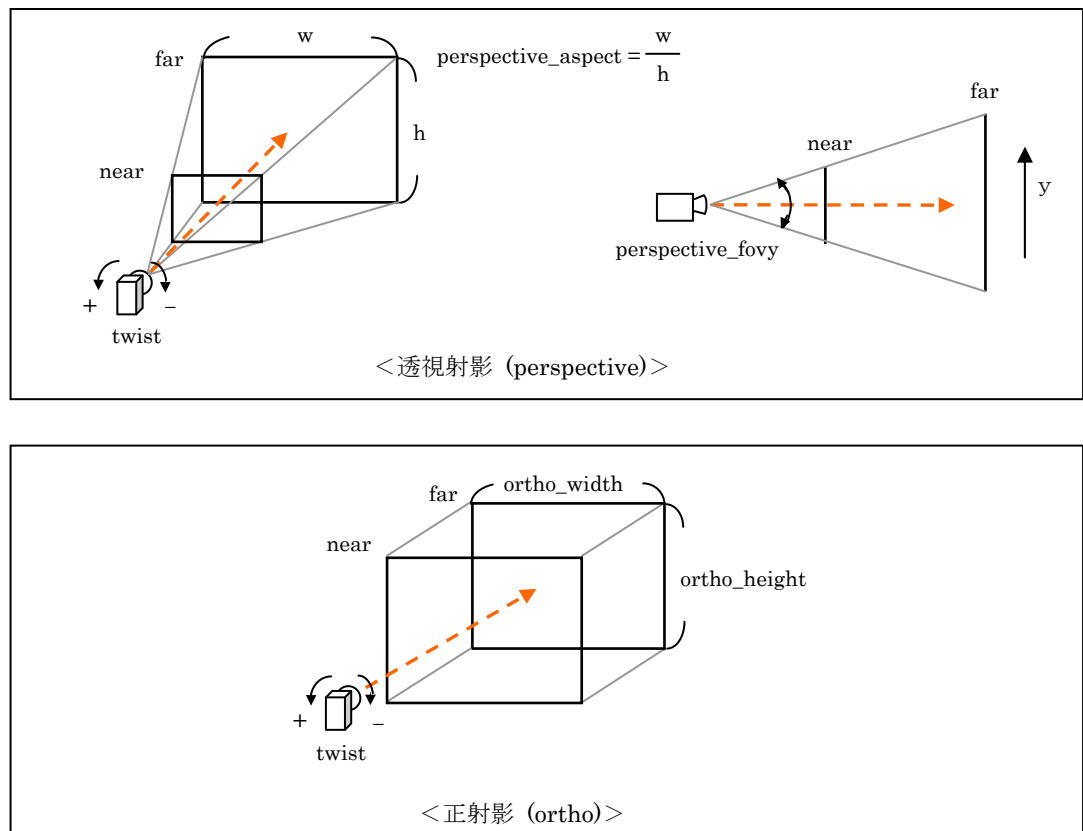


図 6-1 <camera>要素の属性

内容 なし

6.1.3.2 <light>要素

説明 ライトの情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<light>
  <light0
    ... (省略) ...
  />
  <light1
    ... (省略) ...
  />
  <light2
    ... (省略) ...
  />
  <light3
    ... (省略) ...
  />
</light>
```

属性 なし

内容 <light0>, <light1>, <light2>, <light3>要素を一つずつこの順序で格納します。

6.1.3.3 <light0>,<light1>,<light2>,<light3>要素

説明 ライト0、ライト1、ライト2、ライト3の情報が格納されます。

出力条件 必ず1つずつ出力されます。

出力例

```
<light>
  <light0
    direction="-1.000000 0.000000 0.000000"
    color="31 31 31"
  />
  <light1
    direction="-1.000000 0.000000 0.000000"
    color="31 0 0"
  />
  <light2
    direction="0.000000 -1.000000 0.000000"
    color="0 31 0"
  />
  <light3
    direction="0.000000 0.000000 -1.000000"
    color="0 0 31"
  />
</light>
```

属性

表 6-2 <light0>,<light1>,<light2>,<light3>要素の属性

属性名	内容
direction	方向ベクトル。 (-1.0 以上 1.0 未満の実数)×3 NITRO コマンド LightVector の方向ベクトル(X, Y, Z)に相当します。
color	色。 (0以上 31 以下の整数)×3 NITRO コマンド LightColor のライトカラー(R, G, B)に相当します。

内容 なし

6.1.3.4 <shininess_table>要素

説明 鏡面反射輝度テーブルデータが格納されます。

NITRO コマンド Shininess に相当します。

出力条件 必ず1つ出力されます。

出力例

```
<shininess_table>
  0 2 4 6 8 10 12 14
  ... (省略) ...
  241 243 245 247 249 251 253 255
</shininess_table>
```

属性 なし

内容 0 以上 255 以下の整数が128個並びます。

※NITRO コマンド Shininess に設定する符号なし小数部8ビット値を0以上 255 以下の整数と

みなして格納します。

6.1.3.5 <toon_highlight>要素

説明 トゥーン/ハイライトポリゴンのモード情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<toon_highlight mode="toon">
  <toon_table>
    ... (省略) ...
  </toon_table>
</toon_highlight>
```

属性

表 6-3 <toon_highlight>要素の属性

属性名	内容
mode	トゥーン/ハイライトポリゴンのモード。 [toon / highlight] のいずれか toon: トゥーンシェーディングを用います。 highlight: ハイライトシェーディングを用います。 3D 表示コントロールレジスタ (DISP3DCNT) のトゥーン/ハイライトポリゴンモードに相当します。

内容 <toon_table>要素を一つ格納します。

6.1.3.6 <toon_table>要素

説明 トゥーンテーブルデータが格納されます。

トゥーンテーブルレジスタ (ToonTable) に相当します。

出力条件 必ず1つ出力されます。

出力例

```
<toon_highlight>
  <toon_table>
    0 0 0
    ... (省略) ...
    31 31 31
  </toon_table>
</toon_highlight>
```

属性 なし

内容 0以上 31 以下の整数が 96 個 (RGB 要素 x32) 並びます。

6.1.3.7 <fog>要素

説明 フォグ情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```

<fog
  color="31 31 31"
  alpha="0"
  offset="0"
  shift="0"
  mode="color_alpha"
>
  <fog_table>
    ... (省略) ...
  </fog_table>
</fog>

```

属性

表 6-4 <fog>要素の属性

属性名	内容
color	フォグカラーのRGB成分。 (0 以上 31 以下の整数)×3 フォグカラーレジスタ(FogColor)のフォグカラー(R, G, B)に相当します。
alpha	フォグのα成分。 (0 以上 31 以下の整数)×1 フォグカラーレジスタ(FogColor)のフォグα値に相当します。
offset	フォグのオフセット。 (0以上 32767 以下の整数)×1 フォグオフセットレジスタ(FogOffset)に相当します。
shift	フォグシフト値。 (0以上 10 以下の整数)×1 3D 表示コントロールレジスタ(DISPCNT)のフォグシフトに相当します。
mode	フォグモード。 [color_alpha / alpha]のいずれか color_alpha : ピクセルのカラー値とα値にブレンディング alpha : ピクセルのα値のみにブレンディング 3D 表示コントロールレジスタ(DISPCNT)のフォグモードに相当します。

内容 <fog_table>要素を一つ格納します。

6.1.3.8 <fog_table>要素

説明 フォグテーブルデータが格納されます。
 フォグ濃度テーブルレジスタ(FogTable)に相当します。

出力条件 必ず1つ出力されます。

出力例

```

<fog_table>
  0 0 0 0 0 0 0 0
  ... (省略) ...
  0 0 0 0 0 0 0 0
</fog_table>

```

属性 なし

内容 0以上 127 以下の整数が 32 個並びます。

6.1.3.9 <alpha_test>要素

説明 αテスト情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<alpha_test
  enable="off"
  reference="0"
/>
```

属性

表 6-5 <alpha_test>要素の属性

属性名	内容
enable	α テストのイネーブルフラグ。 [off / on] のいずれか 3D 表示コントロールレジスタ (DISP3DCNT) の α テストイネーブルフラグに相当します。
reference	α テストの比較値。 (0以上 31 以下の整数)×1 α テスト比較値レジスタ (AlphaReference) の α テスト比較値に相当します。

内容 なし

6.1.3.10 <alpha_blending>要素

説明 α ブレンディング情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<alpha_blending enable="off"/>
```

属性

表 6-6 <alpha_blending>要素の属性

属性名	内容
enable	α ブレンディングのイネーブルフラグ。 [off / on] のいずれか 3D 表示コントロールレジスタ (DISP3DCNT) の α ブレンディングイネーブルフラグに相当します。

内容 なし

6.1.3.11 <y_sorting>要素

説明 半透明ポリゴンのYソーティング情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<y_sorting mode="auto"/>
```

属性

表 6-7 <y_sorting>要素の属性

属性名	内容
mode	半透明ポリゴンのYソーティング選択フラグ。 [auto / manual] のいずれか auto: オートソートモード manual: マニュアルソートモード NITRO コマンド SwapBuffers の半透明ポリゴンのYソーティング選択フラグに相当します。

内容 なし

6.1.3.12 <edge_marking>要素

説明 エッジマーキング情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<edge_marking enable="off"/>
  <edge_color>
    31 31 31
    ... (省略) ...
    0 0 0
  </edge_color>
</edge_marking>
```

属性

表 6-8 <edge_marking>要素の属性

属性名	内容
enable	エッジマーキングのイネーブルフラグ。 [off / on] のいずれか 3D 表示コントロールレジスタ (DISP3DCNT) のエッジマーキングイネーブルフラグ に 相当します。

内容 <edge_color>要素を一つ格納します。

6.1.3.13 <edge_color>要素

説明 エッジカラーデータが格納されます。

エッジカラーレジスタ (EdgeColor) に相当します。

出力条件 必ず1つ出力されます。

出力例

```
<edge_color>
  31 31 31
  ... (省略) ...
  31 0 0
</edge_color>
```

属性 なし

内容 0以上 31 以下の整数が24個 (RGB要素 x 8)並びます。

6.1.3.14 <antialias>要素

説明 アンチエイリアス情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<antialias enable="off"/>
```

属性

表 6-9 <antialias>要素の属性

属性名	内容
enable	アンチエイリアスのイネーブルフラグ。 [off / on]のいずれか 3D表示コントロールレジスタ(DISP3DCNT)のアンチエイリアシングイネーブルフラグに相当します。

内容 なし

6.1.3.15 <render_1_pixel_depth>要素

説明 1ピクセル(ドット)ポリゴンの表示境界デプス値が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<render_1_pixel_depth w="2048.000000"/>
```

属性

表 6-10 <render_1_pixel_depth>要素の属性

属性名	内容
w	1ピクセル(ドット)ポリゴンの表示境界デプス値。 (0以上の実数) x1 1ドットポリゴン表示境界デプス値レジスタ(Disp1DotDepth)に相当します。

内容 なし

6.1.3.16 <clear_color>要素

説明 クリアカラー情報が格納されます。

出力条件 必ず1つ出力されます。

出力例

```
<clear_color
  color="31 31 31"
  alpha="0"
  polygon_id="0"
  fog_enable="off"
  depth="0"
/>
```

属性

表 6-11 <clear_color>要素の属性

属性名	内容
color	クリアカラー値。 (0以上31以下の整数) x3 クリアカラーアトリビュートレジスタ(ClearColorAttr)のクリアカラー(R, G, B)に相当します。
alpha	α 初期値。 (0以上31以下の整数) x1 クリアカラーアトリビュートレジスタ(ClearColorAttr)のα 初期値に相当します。

polygon_id	ポリゴンID初期値。 (0以上63以下の整数) x1 クリアカラーアトリビュートレジスタ (ClearColorAttr) のポリゴンID初期値に相当します。
fog_enable	フォグイネーブルフラグ。 [off / on] のいずれか クリアカラーアトリビュートレジスタ (ClearColorAttr) のフォグイネーブルフラグに相当します。
depth	クリアデプス値。 (0以上 32767 以下の整数) x1 クリアデプスレジスタ (ClearDepth) に相当します。

内容 なし

6.1.4 isdの補足

- カメラやライトのアニメーションデータ出力について
3DCG ツール上のカメラやライトなどグラフィックスシーンに含まれる要素の値やそのアニメーションデータの間ファイル出力については、現時点では未定です。

7 データの最適化

ここでは、3DCGツールから中間ファイルを出力する際に行われるデータの最適化について補足説明します。

7.1 ノード削減

3Dモデルを表示する場合、ノード数が多くなるほど行列計算の負荷が大きくなります。NITRO 中間ファイルプラグインを使って3DCGツールから imd ファイルを出力する際、その3Dモデルを描画する上で不要なノードを削減する最適化処理によって、行列計算の負荷を軽減し、またメモリ使用量を抑えることができます。

NITRO 中間ファイルプラグインで指定できるノード削減の種類は以下のとおりです。

詳しい説明は、各3DCGツールの NITRO 中間ファイルプラグインマニュアルをご覧ください。

表 7-1 ノード削減の種類

モード	削減内容
None	ノードを削減しません。 3DCGツール上の階層構造のまま出力します。
Cull Useless Node	モデルの表示に必要なでないノードを削除して出力します。 スキニング(エンベロープ)を使ったキャラクタモデルに効果があります。
Merge Useless Node	Cull Useless Node の処理に加え、描画に不要なノードを他のノードと合成することで Cull Useless Node よりもノード数を減らします。 ただし、"子供を持つノードに非均一な (Scale x、y、zの値が同じでない) Scale が設定されてはいけいない"という制限がつきます。 この制限が守られていない場合、モデルが正しく表示されない場合があります。 条件付きですが、ノード数を最大限に減らしたいキャラクタモデルに有効です。
Unite	ノードを1つにまとめ、すべてのポリゴンをグローバル座標で出力します。 ただし、元々別ノードに属していたポリゴン(imd の<polygon>に相当)はマテリアルが同じでも個別に出力されます。 部分ごとにカリングを行いたい地形データに有効なモードです。
Unite and Combine Polygon	Unite の処理に加え、同じマテリアルで表示するポリゴンを1つにまとめます。ただし、マテリアルが同じでも頂点カラーの有無や描画優先度が異なるポリゴンはまとめられません。 Unite より表示速度の向上が期待できますが、細かいクリッピングはできなくなります。 できる限りデータサイズと処理を軽くしたい地形データに有効です。

7.2 マテリアル圧縮

名前は異なるが設定内容が全て一致するマテリアルを1つにまとめることで、imd のデータサイズを抑え、TWL およびニンテンドーDS 上での処理の効率化をはかります。該当するマテリアルが存在した場合、名前の若い方だけを出力します。ただし、マテリアルカラーアニメーション、テクスチャパターンアニメーション、テクスチャ SRT アニメーションが設定されているマテリアルは、個別にマテリアルが存在する必要があるため圧縮されません。

7.3 ポリゴンの連結化

3DCGツールから中間ファイルを出力する際、出力ダイアログ上で **use_primitive_strip** 機能をONにすると、プラグイン内部で自動的にポリゴンの連結化(連結三角形ポリゴン及び連結四角形ポリゴンを用いる)処理を行い、できるだけ少ない NITRO ジオメトリコマンド数でポリゴンを描画するように最適化します。

ポリゴンを連結化しない場合(=上記機能をOFFにした時)は、全てのポリゴンを三角形ポリゴンもしくは四角形ポリゴンで表示します。

連結化してもしなくてもモデルの表示に変化はありませんが、連結化した方が処理頂点数が少なく、データサイズも小さくなります。

8 参考情報

中間ファイルを使って、独自の表示ルーチンを構築する人向けに参考となる情報を記載します。

8.1 行列計算について

8.1.1 ノードの行列の計算方法

基本的な行列計算方法について

親子構造が a(親) → b(子) → c(孫) のノード c に対する一般的な行列は、左から座標を掛けるとすると $[Sc] * [Rc] * [Tc] * [Sb] * [Rb] * [Tb] * [Sa] * [Ra] * [Ta]$ となります。

これ以外に、各3DCGツールごとに独自の行列計算が用いられる場合があります。

Maya の Segment Scale Compensate について

Maya では、Segment Scale Compensate(スケール補正)というアトリビュートを用いて独自の行列計算を行うこともできます。Segment Scale Compensate を用いた行列計算式については、NITRO 中間ファイルプラグイン for Maya、もしくは Maya のマニュアルを参照ください。

SOFTIMAGE|XSI の Hierarchical Scaling について

SOFTIMAGE|XSI では、Hierarchical Scaling というスイッチがあり、これをオンにすると SOFTIMAGE|3D の Classic Scaling と同様にソフトイメージ独自の行列計算で描画することができます。

ただし、SOFTIMAGE|XSI ではノード毎に Hierarchical Scaling の設定を切り替えることができますが、現時点では SOFTIMAGE|3D の Classic Scaling 同様、モデル全体で設定する項目として扱います。

ソフトイメージ独自の行列計算(Hierarchical Scaling がオン)と同じようにモデルを表示する方法については、NITRO 中間ファイルプラグイン for SOFTIMAGE|XSI のマニュアルを参照ください。

8.1.2 テクスチャ行列の計算方法

テクスチャ行列の計算方法は、3DCGツールごとに異なります。各 3DCG ツールと同じように表示するテクスチャ行列の計算方法については、各 3DCG ツール用の NITRO 中間ファイルプラグインマニュアル、もしくは各3DCGツールに付随するマニュアル等を参照ください。

3ds max、Maya、Softimage、SOFTIMAGE|3D、SOFTIMAGE|XSI は、Autodesk, Inc. の米国およびその他の国における登録商標または商標です。

その他、記載されている会社名、製品名等は、各社の登録商標または商標です。

© 2003–2009 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。