

インテル® レンダリング・ツールキット・リリースノート

本書は、英文「[Intel® Rendering Toolkit Release Notes](#)」(英語)の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

更新日: 2023 年 11 月 8 日

バージョン: 2024.0

インテル® レンダリング・ツールキットには、レンダリングおよびレイトレーシング・ライブラリーのセットが含まれています。ライブラリーは、詳細または生データをリッチでリアルなビジュアルに変換できるように支援します。ライブラリーは、一連のビジュアライゼーション・ニーズ向けに最適化されています。本リリースノートには、インテル® レンダリング・ツールキットの新機能や既知の問題に関する情報が含まれます。ソフトウェアとハードウェアの要件は、「[インテル® レンダリング・ツールキットの動作環境](#)」を参照してください。

インテル® レンダリング・ツールキットには以下のコンポーネントが含まれています。

- [インテル® Embree](#)
- [インテル® オープン・イメージ・デノイズ](#)
- [インテル® オープン・ボリューム・カーネル・ライブラリー \(インテル® オープン VKL\)](#)
- [インテル® OSPRay](#)
- [インテル® オープン・パス・ガイディング・ライブラリー \(インテル® オープン PGL\)](#)
- [インテル® OSPRay Studio \(英語\)](#)
- [インテル® インプリシット SPMD プログラム・コンパイラー \(インテル® ISPC\)](#)
 - ISPCRT ランタイム
- [インテル® レンダリング・ツールキット・ユーティリティ \(rkUtil\)](#)
 - rkUtil superbuild CMake* スクリプト

自動的にインストールされるコンポーネント:

- [インテル® rkcommon C++ インフラストラクチャー・ライブラリー](#)
- [インテル® oneAPI スレッディング・ビルディング・ブロック \(インテル® oneTBB\)](#)

別途配布されているコンポーネント

- [HD Hydra 向けインテル® OSPRay インタラクティブ・レンダリング・プラグイン \(英語\)](#)
- [インテル® OpenSWR](#)

2024.0 の新機能

- インテル® レンダリング・ツールキットのライブラリー (インテル® Embree、インテル® オープン・イメージ・デノイズ、インテル® オープン・ボリューム・カーネル・ライブラリー、インテル® OSPRay、インテル® OSPRay Studio) は、CPU のサポートに加えて、インテル® Xe アーキテクチャー・ベースの GPU (インテル® Arc™ グラフィックス、インテル® データセンター GPU フレックス・シリーズおよびインテル® データセンター GPU マックス・シリーズ) をサポートします。

- インテル® Embree は、受賞歴のある CPU でのレイトレースのサポートに加えて、インテル® Arc™ グラフィックス、インテル® データセンター GPU フレックス・シリーズおよび新しく追加されたインテル® データセンター GPU マックス・シリーズでプロダクションレベルのレイトレーシング・ハードウェア・アクセラレーションを実現します。
- インテル® オープン・イメージ・デノイズは、CPU でのサポートに加えて、インテル® Arc™ グラフィックス、インテル® データセンター GPU フレックス・シリーズおよびインテル® データセンター GPU マックス・シリーズで最先端のノイズ除去をサポートします。
- ディスクリート GPU をターゲットにするには、インテル® oneAPI DPC++ コンパイラーおよびランタイムを使用します。

リリースの入手方法

パッケージの入手方法については、[インテル® レンダリング・ツールキットの製品ページ](#)を参照してください。

インテル® レンダリング・ツールキット 2024.0

コンポーネント名	バージョン
インテル® Embree	4.3
インテル® OSPRay	3.0-Beta
インテル® オープン・パス・ガイディング・ライブラリー (インテル® オープン PGL)	0.5.0
インテル® オープン・イメージ・デノイズ	2.1
インテル® オープン・ボリューム・カーネル・ライブラリー	2.0.0-Beta
インテル® OSPRay Studio	0.13
インテル® インプリシット SPMD プログラム・コンパイラー (インテル® ISPC)	1.21.0
インテル® レンダリング・ツールキット・ユーティリティ	1.9
インテル® rkcommon C++ インフラストラクチャー・ライブラリー	1.12
インテル® oneTBB	2021.11

インテル® Embree 4.3

GPU

インテル® Embree でディスクリート GPU をターゲットにするには、[インテル® oneAPI ベース・ツールキット](#)に含まれるインテル® oneAPI DPC++ コンパイラーおよびランタイム 2024.0 を使用します。

GPU - Linux*

ディスクリート GPU の OS およびドライバーサポートは頻繁に更新されます。OS サポートの更新は、<https://dgpu-docs.intel.com/> (英語) を参照してください。現在、ディスクリート GPU をターゲットにするには、クライアント・プログラムのビルド時と実行時の両方で、互換性のあるディスクリート GPU デバイスが必要です。

GPU - Windows*

ドライバーをインストールするには、次の手順に従ってください。

- [インテル® Arc™ A シリーズ・グラフィックス \(開発コード名 DG2\)](#)
- インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)、インテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio、略称 PVC)
 - インテル® レジストレーション・センターのアクセス方法は、インテル製品の担当者までお問い合わせください。

インテル® Embree 4.3 の変更点

- 同様のインスタンスを大量に含むシーンでのメモリー要件を軽減するため、インスタンス配列プリミティブを追加しました。
- L0 RTAS 拡張機能をロードできるかどうかドライバーを適切にチェックするようにしました。
- インテル® ISPC の `rtcGetGeometryTransform` の異なるバージョンを追加しました。
- インテル® ISPC の `RTCMemoryMonitorFunction` のシグネチャーを修正しました。
- CMake* で ARM64 Windows* プラットフォームのサポートを追加しました。

インテル® Embree 4.2 の変更点

- GPU をサポートする SYCL* バージョンのインテル® Embree はベータ段階ではなくなりました。
- スレッドをオーバーサブスクライブするアプリケーションのメニーコアマシン上での BVH ビルド・パフォーマンスが向上しました。
- SYCL* カーネル内で使用できる `rtcGetGeometryTransformFromScene` API 関数を追加しました。
- CPU レンダリングのレベルゼロ依存関係を回避するため、SYCL* モードで `ze_loader` にリンクしないようになりました。
- インテル® Embree をテストするテストパッケージをリリースしました。

インテル® OSPRay 3.0

GPU - Linux*

ディスクリート GPU の OS およびドライバーサポートは頻繁に更新されます。OS サポートの更新は、<https://dgpu-docs.intel.com/> (英語) を参照してください。現在、ディスクリート GPU をターゲットにするには、クライアント・プログラムのビルド時と実行時の両方で、互換性のあるディスクリート GPU デバイスが必要です。

GPU - Windows*

ドライバーをインストールするには、次の手順に従ってください。

- [インテル® Arc™ A シリーズ・グラフィックス \(開発コード名 DG2\)](#)
- インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)、インテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio、略称 PVC)
 - インテル® レジストレーション・センターのアクセス方法は、インテル製品の担当者までお問い合わせください。
- [第 12 世代インテル® プロセッサ・グラフィックス、インテル® Iris® X^e MAX \(開発コード名 DG1\) デバイス](#)はインテル® オープン・イメージ・デノイズで機能します。

GPU 使用状況

インテル® OSPRay 3.0 で GPU をターゲットにするには、C++17/C99 システム・コンパイラー・ツールチェーンとインテル® oneAPI DPC++ ランタイムのみ必要です。インテル® oneAPI DPC++ を使用したクライアント・アプリケーションの作成はオプションです。インテル® oneAPI DPC++ を使用する最も簡単な方法は、[インテル® oneAPI ベース・ツールキット](#)からコンパイラーとランタイムをインストールすることです。

GPU 上でのアプリケーションの実行方法は、[ドキュメント](#) (英語) の「Modules and Devices/GPU (モジュールとデバイス/GPU)」セクションを参照してください。インテル® OSPRay 3.0 の GPU のサポートはベータレベルであり、すべての機能がサポートされているわけではないことに注意してください。

サポートされているディスクリート GPU デバイスを検出した場合、インテル® OSPRay は、インテル® オープン・イメージ・デノイズのイメージ操作が有効なときにデバイスを使用します。

インテル® OSPRay 3.0 の変更点

- インテル® Xe アーキテクチャー・ベースの GPU (インテル® Arc™ GPU、インテル® データセンター GPU フレックス・シリーズ、インテル® データセンター GPU マックス・シリーズ) のレイトレーシング・ハードウェアをベータ版としてサポートしました。実装は、[インテル® oneAPI データ並列 C++ \(DPC++\)](#) (英語) で実装された [SYCL*](#) (英語) クロスプラットフォーム・プログラミング言語に基づいています。

次の機能はまだ未実装で、新しい gpu デバイスでは正しく動作しないことに注意してください。

- シーンでの複数ボリューム
- クリッピング
- モーションブラー
- 細分化サーフェス
- `ospGetProgress` による進捗レポートまたは `ospCancel` によるフレームのキャンセル
- `ospPick` によるピッキング
- `OSP_FB_VARIANCE` および `varianceThreshold` によるアダプティブ累積
- フレームバッファ・チャンネル `OSP_FB_ID_*` (id バッファ)
- `structuredRegular` ボリュームでのみ動作する共有デバイスのみデータの試験的サポート
- その他の既知の問題
 - カーネルコードは JIT コンパイルされ特殊化されているため、起動時またはシーンの変更時に多少の遅延が発生します。`SYCL_CACHE_PERSISTENT=1` 環境変数を設定して JIT コンパイルをキャッシュすると、遅延なく連続起動や同一シーン構成のレンダリングを行うことができます。
 - コンパイラー、GPU ドライバー、シーンの一部の組み合わせで、レンダリングされたイメージにアーティファクト (垂直線や小さなブロックなど) が表示されることがあります。
 - マルチデバイスはノイズ除去やトーンマッピングで `OSPImageOperation` をサポートしていません。
- `mesh` ジオメトリの暗黙的なインデックス作成を追加しました。
- 一時バッファの所有権転送をサポートしました。`ospNewSharedData` でオプションの削除子コールバックを指定できます。
- 新しい統合パフォーマンス・トレーシング・フレームワークでサポートされる MPI モジュールを最適化しました。

- `scivis` 勾配シェーディングを最適化しました。
- メインスレッドは非正規化数の処理に `FTZ/DAZ` を設定しないようになりました。
- 自己交差によるレンダリング・アーティファクトを回避するため、メッシュの交差イプシロンを計算するようになりました。
- 特定のパラメーターの組み合わせにおける `Pricipled` マテリアルの省電力を修正しました。
- アルファチャンネルをワイプしないように `denoiser` を修正しました。
- HDRI ライトのクラッシュを修正しました。
- Windows* のデバッグビルドのリンク順を修正しました。
- 依存ファイルの最小バージョンを変更しました。
 - インテル® Embree 4.3.0
 - インテル® オープン VKL 2.0.0
 - インテル® オープン・イメージ・デノイズ 2.1.0
 - インテル® ISPC 1.21.1
 - インテルの `rkCommon` ライブラリー 1.12.0
- API の次の点を変更しました。
 - `OSP_TEXTURE_FILTER_BILINEAR` の名前を `OSP_TEXTURE_FILTER_LINEAR` に、`OSP_VOLUME_FILTER_TRI [LINEAR|CUBIC]` の名前を `OSP_VOLUME_FILTER_ [LINEAR|CUBIC]` に、それぞれ変更しました。
 - 多くの enum でストレージ型 `uint32` を使用するようになりました。
 - 球面の通常のポリュームの `gridSpacing` のデフォルトを完全な球に変更しました。
 - 非推奨のパラメーターと API 呼び出しを削除しました。
 - ユーザーポインターのないエラー・コールバックのシグネチャー
 - `ospNewMaterial` の最初の引数
 - モジュール名 `ispc.cpu` を使用
 - `volume` テクスチャーと `isosurface` ジオメトリー: `volumetricModel`。 `OSPVolume volume` を使用
 - 伝達関数 `vec2f valueRange`、`box1f value` を使用
 - `hdri` および `sun-sky` ライト: `intensityQuantity` `OSP_INTENSITY_QUANTITY_RADIANCE`
 - `intensityDistribution` を含む `spot` ライト: `OSP_INTENSITY_QUANTITY_SCALE` 以外の `intensityQuantity`

インテル® オープン・パス・ガイディング・ライブラリー (インテル® オープン PGL) 0.5.0

インテル® レンダリング・ツールキット 2023.x からの変更点はありません。

インテル® オープン・イメージ・デノイズ 2.1

主要な改訂により API シグネチャーや予想が追加または変更される可能性があるため、新しい API 呼び出しについては [API \(英語\)](#) を参照してください。

GPU サポート

- インテルのデバイス:
 - インテル® Arc™ A シリーズ・グラフィックス (開発コード名 DG2)
 - インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)
 - インテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio、略称 PVC)
 - 第 12 世代インテル® プロセッサー・グラフィックス
 - インテル® Iris® Xe MAX グラフィックス (開発コード名 DG1)
- 他社のデバイス:
 - NVIDIA* GPU:
 - Volta、Turing、Ampere、Ada Lovelace、および Hopper アーキテクチャー
 - AMD* GPU:
 - RDNA* 2 (Navi 21 のみ) および RDNA* 3 (Navi 3x) アーキテクチャー

GPU - Linux*

ディスクリート GPU の OS およびドライバーサポートは頻繁に更新されます。OS サポートの更新は、<https://dgpu-docs.intel.com/> (英語) を参照してください。現在、ディスクリート GPU をターゲットにするには、クライアント・プログラムのビルド時と実行時の両方で、互換性のあるディスクリート GPU デバイスが必要です。

GPU - Windows*

ドライバーをインストールするには、次の手順に従ってください。

- [インテル® Arc™ A シリーズ・グラフィックス \(開発コード名 DG2\)](#)
- インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)、インテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio、略称 PVC)
 - インテル® レジストレーション・センターのアクセス方法は、インテル製品の担当者までお問い合わせください。
- [第 12 世代インテル® プロセッサー・グラフィックス、インテル® Iris® Xe MAX \(開発コード名 DG1\) デバイス](#)はインテル® オープン・イメージ・デノイズで機能します。

GPU 使用状況

インテル® オープン・イメージ・デノイズ 2.1 で GPU をターゲットにするには、C++17/C99 システム・コンパイラー・ツールチェーンとインテル® oneAPI DPC++ ランタイムのみが必要です。インテル® oneAPI DPC++ を使用したクライアント・アプリケーションの作成はオプションです。インテル® oneAPI DPC++ を使用する最も簡単な方法は、[インテル® oneAPI ベース・ツールキット](#)からコンパイラーとランタイムをインストールすることです。

バージョン 2.1.0 での変更点

- 1 チャンネル (アルファなど) および 2 チャンネルのイメージでノイズ除去をサポートしました。
- 入力イメージのデータ型の任意の組み合わせをサポートしました (例: `color` の場合は `OIDN_FORMAT_FLOAT3`、`albedo` の場合は `OIDN_FORMAT_HALF3`)。
- 多くの専用 GPU アーキテクチャーのパフォーマンスが向上しました。

- `OIDN_STATIC_LIB` CMake* オプションが再度追加され、スタティック (CPU サポートのみ) またはハイブリッド・スタティック/共有 (GPU もサポート) ライブラリーとしてビルドできるようになりました。
- C++ API オブジェクト (`DeviceRef`、`BufferRef`、`FilterRef`) に `release()` メソッドを追加しました。
- GPU デバイス、バッファまたはフィルターをリリースするときに発生する可能性があったクラッシュを修正しました。
- SYCL* ランタイムの一部のバージョンでプロセス終了時に発生する可能性があったクラッシュを修正しました。
- インテルの統合 GPU での画質の不整合を修正しました。この修正に伴い、パフォーマンスは若干低下しました。
- インテルの統合 GPU に対する将来の Windows* ドライバーの互換性を修正しました。
- AMD の RDNA* 2 GPU でまれに発生していた出力破壊問題を修正しました。
- ライブラリーのパスに非 ANSI 文字が含まれる場合の Windows* でのデバイス検出を修正しました。
- インテル® oneAPI DPC++/C++ コンパイラー 2024.0 および互換性のあるオープンソースのコンパイラーのバージョンをサポートしました。
- 公式バイナリーでインテル® oneTBB 2021.10.0 にアップグレードしました。
- 古いインテル® oneTBB バージョンの検出を改善しました。

バージョン 2.0.1 での変更点

- 最近の Linux* ドライバーを使用した場合のインテルの統合 GPU のパフォーマンスの問題を修正しました。
- 専用および統合 AMD* GPU の両方を搭載したシステムでのクラッシュを修正しました。
- CUDA* および HIP デバイスでの `D3D12_RESOURCE`、`D3D11_RESOURCE`、`D3D11_RESOURCE_KMT`、`D3D11_TEXTURE` および `D3D11_TEXTURE_KMT` 外部メモリータイプのインポートを修正しました。
- 公式 x86 バイナリーの macOS* デプロイメント・ターゲットを修正しました (11.0 から 10.11 に変更)。
- 詳細出力のマイナーな改善を行いました。

インテル® オープン・ボリューム・カーネル・ライブラリー (インテル® オープン VKL) 2.0.0

GPU

インテル® Embree でディスクリート GPU をターゲットにするには、[インテル® oneAPI ベース・ツールキット](#) に含まれるインテル® oneAPI DPC++ コンパイラーおよびランタイム 2024.0 を使用します。

GPU - Linux*

ディスクリート GPU の OS およびドライバーサポートは頻繁に更新されます。OS サポートの更新は、<https://dgpu-docs.intel.com/> (英語) を参照してください。現在、ディスクリート GPU をターゲットにするには、クライアント・プログラムのビルド時と実行時の両方で、互換性のあるディスクリート GPU デバイスが必要です。

GPU - Windows*

ドライバーをインストールするには、次の手順に従ってください。

- [インテル® Arc™ A シリーズ・グラフィックス \(開発コード名 DG2\)](#)

- インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)、インテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio、略称 PVC)
 - インテル® レジストレーション・センターのアクセス方法は、インテル製品の担当者までお問い合わせください。

バージョン 2.0.0 での変更点

- SYCL* を介したインテル® Arc™ グラフィックス、インテル® データセンター GPU フレックス・シリーズおよびインテル® データセンター GPU マックス・シリーズのサポートを追加しました。
 - このリリースの SYCL* サポートはベータ段階です。現在の機能、品質、GPU パフォーマンスは、最終製品の機能、品質、GPU パフォーマンスを反映したものではありません。
 - このリリースの CPU サポートはゴールドレベルのままで、品質とパフォーマンスは以前のリリースと同じです。
- API の次の点を変更しました。
 - 次の API では、型がポインターで渡されるようになりました。
 - `vk1ComputeSample* ()`
 - `vk1ComputeGradient* ()`
 - `vk1Get*IteratorSize* ()`
 - `vk1Init*Iterator* ()`
 - `vk1Iterate* ()`
 - `vk1LoadModule ()` を削除しました。コンパイル時にインテル® オープン VKL のデバイス実装 (`cpu` または `gpu`) へのリンクが必要になりました。
 - ライブラリーの初期化のために呼び出す必要がある `vk1Init ()` API を追加しました。
 - `VKL_FILTER_[TRILINEAR, TRICUBIC]` の名前を `VKL_FILTER_[LINEAR, CUBIC]` に変更しました。
 - `VKLAMRMethod` enum の型を `uint32_t` に変更しました。
 - `structuredSpherical` ボリューム: `gridSpacing` のデフォルトで、ボリュームが完全な球体を占めるようになりました。
- GPU の使用状況を示す新しい例 (`vk1ExamplesGPU` および `vk1TutorialGPU`) を追加しました。
- `superbuild` の依存関係を最新バージョンに更新しました。

インテル® OSPRay Studio 0.13

- インテル® OSPRay 3.0 と互換性があります。
- 新機能と改善点
 - インテル® OSPRay 3 との互換性:
 - インテル® Xe アーキテクチャー・ベースの GPU をベータ版としてサポート: `--osp:load-modules=gpu --osp:device=gpu` パラメーターを指定して Studio を起動します。
 - インテル® OSPRay 3 の enum との互換性を追加し、すべての enum の使用で型チェックを強化しました。
 - `.sg` ファイルの `json` にインテル® OSPRay 3 の enum 型の内包表記を追加しました。**(注: この変更により既存の `.sg` ファイルは変更されます。最も一般的なパラメーターは「`uchar`」の `subType` を含んでいた「`intensityQuantity`」で、「`OSPIintensityQuantity`」に変更する必要があります。)**

- テクスチャー形式: デフォルトで .exr および .tiff イメージ形式をサポートします。tinyexr および tinydng サードパーティーのヘッダーのみライブラリーを含みます。OpenImageIO はビルドオプションとして引き続き利用できます。
- glTF ライトの強度をルクスからワットに変換しました。glTF は強度をルクスで、インテル® OSPRay をワットで指定します。glTF エクスポーターとの互換性のためにワットに変換しました。
- ほかのマッピングが存在しない場合に使用するデフォルトの HDRI マッピングを追加しました。ライトから HDRI マッピングを削除したとき、またはテクスチャーのロードに失敗したとき、従来はインテル® OSPRay がクラッシュしていましたが、このデフォルトのマッピングに自動的に切り替わるようになりました。
- UI「pick focus distance (焦点距離を選択)」を修正しました。
 - 左 Shift + マウスの左ボタンは、選択ポイントを画面の中心、円弧の中心に設定します。
 - 左 Ctrl + 左 Shift + マウスの左ボタンは、円弧の中心を移動しないで選択ポイントを焦点距離に設定します。
- [すべて検索] アクションと選択したアクションの動作が直感的になりました。
- 変換エディターで、選択したノードの親をトラバースできるようになりました。
- GenerateImGuiWidgets のクリーンアップ/リファクタリングのマイナーな改善を行いました。
- ファイル名ウィジェットの衝突の回避: 複数のファイル名ウィジェットが表示されている場合、それらはすべて fileBrowser により返されるファイルリストを受け取ります。
- Python* バインディングと Python* チュートリアル・スクリプトで OpenImageDenoise を有効にしました。
- インスタンスのみでなくアセットをリロードする機能を追加しました。
- アセットによりロードされたカメラは、置き換えるのではなくシーンカメラに追加されるようになりました。
- selectCamera をプラグインで呼び出して StudioContext で利用できるようにしました。
- mtl ファイル内の一般的な texture.transform の解析を有効にしました。従来は、個別の回転、スケール、および移動に制限されていました。
- クリーンアップおよび問題の修正:
 - 削除されたインポーター・アセットの nullptr 逆参照を防ぐようにしました。
 - 一致する検索語がないときに SearchWidget がクラッシュしていた問題を修正しました。
 - C++17 以前との互換性のためにカスタム削除子を使用するようにしました。
 - Texture2D での削除の誤用を解決しました。
 - 既存のライト・パラメーターを更新するように addLight を修正しました。addLight を既存のライトで呼び出した場合、すぐにリターンしていました。ライト・パラメーターを新しいライト・パラメーターに更新する必要があります。
 - glTF インポーターに一意の名前を追加する場合は注意してください。ノード、ライト、またはカメラの名前が不明な場合、名前に一意の番号が追加されます。インポーターは、この操作を誤って 2 回実行したり、一意の名前に一意の番号を追加していました。

インテル® OSPRay Studio 0.12.1 での変更点

- インテル® OSPRay 2.12.0 と互換性があります。
- クリーンアップおよび問題の修正:
 - 192 以上の MPI ランクで実行する場合の問題を修正しました。
 - clang コンパイルの警告をクリーンアップしました。
 - カメラの getZoomLevel/setZoomLevel で Python* バインディングを利用できるようにしました。
 - intensity=0 に依存するのではなく、ライトの有効化/無効化プロパティを追加しました。
 - テクスチャー・ロード・キャッシュを最適化しました。
 - バッチ・モード・カメラを修正および改善しました。
 - UI と SearchWidget を改善しました。
 - フレームバッファ・メニューにバッファチャンネルの表示を追加しました (アルベド、標準、深さ、ID バッファ)。
 - 16 進コードの色 (#RGB または s#RGB) の obj/mtl 解析を追加しました。

インテル® インプリシット SPMD プログラム・コンパイラ (インテル® ISPC) 1.21

インテル® Embree アプリケーションで最適化された CPU パフォーマンスを実現するには、インテル® ISPC を推奨します。インテル® ISPC コンパイラを使用して、インテル® 64 プロセッサ (CPU) の SIMD 機能を効率的に使用するクライアント・アプリケーションをコンパイルします。現時点では、インテル® Embree GPU またはインテル® オープン VKL GPU アプリケーション向けのインテル® ISPC のクライアント使用モデルはありません。

バージョン 1.21.0 での変更点

テンプレート関数の特殊化をサポートしました。C/C++ の動作と一致し、積極的な最適化が行われるように符号付き整数オーバーフローのルールを変更しました。インテル® ISPC ランタイムを強化しました。複数の安定性とパフォーマンスの修正が行われました。このリリースは、パッチ適用済みの LLVM 15.0.7 をベースとしています。

言語の変更点

- 明示的なテンプレート引数を使用した関数テンプレートの特殊化をサポートしました。詳細は、ドキュメントの「関数テンプレート」セクションを参照してください。

- 符号付き整数オーバーフローの動作を変更しました。

符号付き整数オーバーフローが発生した場合、インテル® ISPC は C および C++ と同様の未定義の動作を想定します。この変更により、互換性の問題が発生する可能性があります。この動作は、--[no-]wrap-signed-int コンパイラ・オプションを使用して管理できます。デフォルトの動作 (バージョン 1.21.0 以前) を維持するには、--wrap-signed-int を使用します。符号付き整数に対して定義されたラップアラウンド動作は維持されますが、一部のコンパイラの最適化が制限される可能性があります。

新しいハードウェアのサポート

インテル® Core™ Ultra プロセッサ (開発コード名 Meteor Lake) に搭載されている新しいインテル® Xe-LPG アーキテクチャーのグラフィックスをサポートしました。

- 2 つの新しいインテル® ISPC ターゲット (xelpg-x16 および xelpg-x8) を追加しました。

- 2 つの新しいデバイス名 (mtl-m および mtl-p) を追加しました。

インフラストラクチャーの変更点

- インテル® ISPC は LLVM の新しいパス・マネージャーを使用するようになりました。最適化パイプラインは、多くのケースでインテル® ISPC のアンロールされたループと手動でアンロールされたループを一致させる LoopFullUnrollPass を導入することにより変更されました。
- インテル® Xe アーキテクチャーの依存関係 (LLVM、LO、vc-intrinsics、SPIRV-Translator) を使用してインテル® ISPC のビルドを促進するインテル® ISPC スーパービルドを導入しました。依存関係を含むアーカイブを生成するか、事前ビルド済みのアーカイブを使用して、インテル® ISPC のみをビルドできます。LLVM およびインテル® ISPC の LTO または LTO+PGO 対応ビルドを生成することもできます。
- LLVM 16 を使用したインテル® ISPC のビルドをサポートしました。

新しいコンパイラー・オプション

- `--mcmmodel` オプション。small および large 値を指定できます。定義は gcc/clang と同様です。large モデルを使用すると、2GB を超えるプログラムが可能になります。
- `--opt=disable-gathers` および `--opt=disable-scatters` オプション。(パフォーマンスの評価のために) ギャザーおよびスキッター命令をサポートするプラットフォーム上でのギャザーおよびスキッター命令の生成を無効にします。
- `--[no-]wrap-signed-int` オプション。符号付き整数オーバーフロー時のラップアラウンド動作を維持します [維持しません]。

インテル® ISPC ランタイムの改善

- `ispcrtSetTaskingCallbacks` を ISPCRT API に追加しました。ISPCLaunch、ISPCAlloc、および ISPCSync のデフォルト実装をオーバーライドできるようにします。
- ISPCRT が CPU 部分と GPU 部分に分割された後は不要になったため、コンパイル時のレベルゼロの依存関係を ISPCRT から削除しました。

インテル® レンダリング・ツールキット・ユーティリティ (rkUtil) 1.9.0

すべてのコンポーネントを 2023.2 リリースのコンポーネントに更新しました。

インテル® OSPRay rkcommon C++ インフラストラクチャー・ライブラリー 1.12.0

- GPU 対応のデータ構造とインテル® OSPRay、インテル® オープン VKL での使用をサポートしました。

インテル® oneTBB 2021.11

[リリースノート](#) (英語)

以前のリリース

2023.x

- [リリースノート](#) (英語)
- [動作環境](#) (英語)

インテル® レンダリング・ツールキットの動作環境

本書は、英文「[Intel® Rendering Toolkit System Requirements](#)」(英語)の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

バージョン: 最新
2023 年 11 月 8 日

はじめに

インテル® レンダリング・ツールキットには、詳細または生データをリッチでリアルなビジュアルに変換できるように支援するレンダリングおよびレイトレーシング・ライブラリーのセットが含まれています。ライブラリーは、一連のビジュアライゼーション・ニーズ向けに最適化されています。インテル® レンダリング・ツールキットの新機能や既知の問題については、「[リリースノート](#)」を参照してください。

インテル® レンダリング・ツールキットには以下のコンポーネントが含まれています。

- レイトレーシング・ライブラリー
 - [インテル® Embree](#)
 - [インテル® オープン・イメージ・デノイズ](#)
 - [インテル® オープン・ボリューム・カーネル・ライブラリー \(インテル® オープン VKL\)](#)
 - [インテル® OSPRay](#)
 - [インテル® オープン・パス・ガイディング・ライブラリー \(インテル® オープン PGL\) \(英語\)](#)
- 開発ショーケース・アプリケーション
 - [インテル® OSPRay Studio \(英語\)](#)
- カーネル最適化コンパイラー
 - [インテル® インプリシット SPMD プログラム・コンパイラー \(インテル® ISPC\)](#)
 - ISPCRT ランタイム
- レンダリング・ツールキット・ユーティリティ (rkUtil)
 - rkUtil [superbuild \(英語\)](#) CMake* スクリプト
- 内部 3D 演算およびタスク・ライブラリー
 - インテルの rkCommon ライブラリー

別途配布されているコンポーネント

- レンダリング・フレームワーク・プラグイン
 - [HD Hydra 向けインテル® OSPRay インタラクティブ・レンダリング・プラグイン \(英語\)](#)
- OpenGL* 互換のソフトウェア・ラスタライザー (Mesa* Gallium)
 - [インテル® OpenSWR](#)

変更履歴

日付	バージョン	主な変更点
2023 年 11 月	2024.0	インテル® OSPRay、インテル® オープン VKL、インテル® オープン・イメージ・デノイズに導入されたベータ版ディスクリート GPU 機能を修正
2023 年 9 月	2023.2.1	OIDN 2 GPU を修正。インテル® oneAPI DPC++ コンパイラー 2023.2.2 でコンポーネントを再構成。
2023 年 7 月	2023.2.0	OIDN 2 は GPU サポートなし。2023.2.1 を使用してください。
2023 年 5 月	2023.1.1	インテル® Embree の hotfix
2023 年 3 月	2023.1	インテル® Embree にベータ版ディスクリート GPU サポートを追加。 インテル® オープン PGL を追加。
2022 年 12 月	2023.0	2022.3.1 からのコンポーネントのアップデートはありません
2022 年 9 月	2022.3.1	hotfix リリース
2022 年 9 月	2022.3	OIDN を除くすべてのコンポーネントを更新。
2022 年 4 月	2022.2	ISPCRT を追加。すべてのコンポーネントを更新。
2021 年 12 月	2022.1	インテル® ISPC を追加。すべてのライブラリーを更新。

インテル® レンダリング・ツールキット 2024.0

以下のシステム要件の詳細は、サポートされているホストとターゲットシステムのベースラインを示しています。古いオペレーティング・システムやその他のデバイスは、コンポーネント単位で互換性がある可能性があります。2024.0 の時点で、インテル® オープン PGL を除くコンポーネントはすべて GPU をサポートしています。

サポートされているオペレーティング・システム

注: 次のディストリビューションは、ほとんどのコンポーネントでサポートされています。これらの OS ディストリビューションはインテルによってテストされたもの、または動作が確認されているものです。その他のディストリビューションは、動作する場合としない場合があります、推奨されません。質問がある場合は、インテル・コミュニティ・フォーラムでサポートを受けることができます。商用サポートを利用可能な場合は、[サポートチケット](#)を作成してください。

Linux* CPU ホスト

- Ubuntu* 22.04 LTS、20.04
- Rocky Linux* 9
- Red Hat* Enterprise Linux* (RHEL) 8、9
- SUSE* Linux* Enterprise Server (SLES*) 15 SP3、SP4、SP5
- Fedora* 37、38
- Debian* 11
- Amazon* Linux* 2022
- WSL 2

Linux* ディスクリート GPU (インテル® Xe-HPG またはインテル® Xe-HPC アーキテクチャー以降)

- Ubuntu* 22.04 LTS
 - インテル® Arc™ A シリーズ・グラフィックス、インテル® データセンター GPU フレックス・シリーズ
- Red Hat* Enterprise Linux* (RHEL) 8.6
 - インテル® データセンター GPU フレックス・シリーズ
- Red Hat* Enterprise Linux* (RHEL) 9.2
 - インテル® データセンター GPU マックス・シリーズ、インテル® データセンター GPU フレックス・シリーズ
- SUSE* Linux* Enterprise Server (SLES*) 15 SP5
 - インテル® データセンター GPU マックス・シリーズ

ディスクリート GPU の OS およびドライバーサポートは頻繁に更新されます。OS サポートの更新は、<https://dgpu-docs.intel.com/> (英語) を参照してください。現在、ディスクリート GPU をターゲットにするには、クライアント・プログラムのビルド時と実行時の両方で、互換性のあるディスクリート GPU デバイスが必要です。

Windows* CPU

- Windows* 10、11
- Windows Server* 2019、2022
- Windows* 10 と Windows* 11 で Microsoft* Windows* Subsystem for Linux* 2 (WSL2) を使用すると、CPU および GPU ワークフロー向けにインテル® ソフトウェア開発ツールとライブラリーのネイティブ Linux* ディストリビューションを Windows* 上にインストールできます。[詳細](#) (英語)。

Windows* GPU

- Windows* 10、11
 - インテル® Arc™ A シリーズ・グラフィックス
 - インテル® データセンター GPU フレックス・シリーズ
 - 第 12 世代インテル® プロセッサ・グラフィックス、インテル® Iris® Xe MAX (開発コード名 DG1) デバイスはインテル® オープン・イメージ・デノイズで機能します。
- Windows Server* 2019、2022
 - インテル® データセンター GPU フレックス・シリーズのみ
- Windows* 10 と Windows* 11 で Microsoft* Windows* Subsystem for Linux* 2 (WSL2) を使用すると、CPU および GPU ワークフロー向けにインテル® ソフトウェア開発ツールとライブラリーのネイティブ Linux* ディストリビューションを Windows* 上にインストールできます。[詳細](#) (英語)。

Windows* インテル® グラフィックス・ドライバー

ドライバーをインストールするには、次の手順に従ってください。

- [インテル® Arc™ A シリーズ・グラフィックス \(開発コード名 DG2\)](#)
- インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)
 - インテル® レジストレーション・センターのアクセス方法は、インテル製品の担当者までお問い合わせください。

- 第 12 世代 Intel® プロセッサ・グラフィックス、Intel® Iris® X^e MAX (開発コード名 DG1) デバイスは Intel® オープン・イメージ・デノイズで機能します。

macOS* CPU のみ

- Intel® 64 ベースのシステム:
 - macOS* 13.0
- Apple* M1 ベースのシステム:
 - oneAPI ディストリビューションの一部ではありません。Apple* M1 ビルドに関するガイダンスは、GitHub* のリポジトリを使用してください。

サポートされているターゲット・ハードウェア・プラットフォーム

次のプラットフォームは、ほとんどのソフトウェア・コンポーネントでサポートされています。

注: Intel® 64 ハードウェア・プラットフォームは Intel® ストリーミング SIMD 拡張命令 4.2 以上をサポートする必要があります。完全なソフトウェアの最適化は、Intel® アドバンスド・ベクトル・エクステンション 512 向けに実装されています。デバイスの性能は、ark.intel.com で検索できます。

- Intel® 64 アーキテクチャーと互換性があるプロセッサ
 - Intel Atom® プロセッサ
 - Intel® Core™ プロセッサ・ファミリー
 - Intel® Xeon® プロセッサ・ファミリー
 - Intel® Xeon® スケーラブル・プロセッサ・ファミリー
- ARM*
- Apple* M1 (GitHub* のスタンドアロン・リリースを参照)
- Intel® X^e-HPG または Intel® X^e-HPC アーキテクチャー以降 (Intel® Embree、Intel® オープン・イメージ・デノイズ、Intel® OSPRay、Intel® オープン・ボリューム・カーネル・ライブラリー)
 - Intel® Arc™ A シリーズ・グラフィックス
 - Intel® データセンター GPU マックス・シリーズ
 - Intel® データセンター GPU フレックス・シリーズ
- Intel® X^e-LP アーキテクチャーまたは第 12 世代以降の Intel® プロセッサ・グラフィックス (Intel® オープン・イメージ・デノイズのみ)
- Volta、Turing、Ampere、Ada Lovelace、および Hopper アーキテクチャーの NVIDIA* GPU (Intel® オープン・イメージ・デノイズのみ)
- RDNA* 2 (Navi 21 のみ) および RDNA* 3 (Navi 3x) アーキテクチャーの AMD* GPU (Intel® オープン・イメージ・デノイズのみ)
- Apple* M1 ベースのシステム:
 - oneAPI ディストリビューションの一部ではありません。Apple* M1 ビルドに関するガイダンスは、GitHub* のリポジトリを使用してください。

ソフトウェア要件

コンポーネント別の必要条件

インテル® Embree の必要条件

- [インテル® oneTBB](#) 2021.9 以降 (ツールキットにより 2021.11 が自動的にインストールされます)
- ディスクリート GPU のサポート: [インテル® oneAPI ベース・ツールキット](#)に含まれるインテル® oneAPI DPC++/C++ コンパイラーおよびランタイム 2024.0 (ツールキットの構成手順で説明)
- オプションの CPU: [インテル® インプリシット SPMD プログラム・コンパイラー \(インテル® ISPC\)](#) (英語) 1.21 以降 (同梱)

インテル® オープン VKL の必要条件

- オプション: [OpenGL*](#) (英語) - サンプルビューアーの実行に必要
- インテルの rkCommon ライブラリー 1.12.0 以降 (同梱)
- [インテル® oneTBB](#) 2021.9 以降 (ツールキットでは 2021.11 が自動的にインストールされます)
- ディスクリート GPU のサポート: [インテル® oneAPI ベース・ツールキット](#)に含まれるインテル® oneAPI DPC++/C++ コンパイラーおよびランタイム 2024.0 (ツールキットの構成手順で説明)
- インテル® Embree 4.3 以降 (同梱)
- [インテル® ISPC](#) (英語) ランタイム 1.21.0 および依存ファイル (同梱)

インテル® オープン・イメージ・デノイズの必要条件

- [インテル® oneTBB](#) 2021.9 以降 (ツールキットでは 2021.11 が自動的にインストールされます)

インテル® OSPRay の必要条件

- [インテル® oneTBB](#) 2021.9 以降 (ツールキットにより自動的にインストールされます)
- インテル® Embree 4.3 以降 (同梱)
- インテルの rkCommon ライブラリー 1.12.0 以降 (同梱)
- インテル® オープン VKL 2.0.0 (同梱)
- インテル® オープン・イメージ・デノイズ 2.1 (同梱)
- オプション: [OpenGL*](#) (英語) - サンプルビューアーの実行に必要
- Linux* のみ: [インテル® ISPC](#) (英語) ランタイム 1.21.0 および依存ファイル (同梱)

インテル® オープン PGL の必要条件

- [インテル® oneTBB](#) 2021.9 以降 (ツールキットでは 2021.11 が自動的にインストールされます)
- インテル® Embree 4.3 以降 (同梱)

インテル® OSPRay Studio の必要条件

- インテル® OSPRay 3.0 および依存ファイル (同梱)
- [OpenGL*](#) (英語) - ウィンドウモード (非バッチモード) に必要
 - [GLFW](#) (非 Windows* プラットフォーム)
- オプション: [OpenImageIO* 2.x](#), [OpenEXR* \(IlmBase\)](#) - テクスチャーの拡張サポートに必要
- オプション: [Python* 3.9.7](#) - [Python*](#) バインディングのサポートに必要

インテル® レンダリング・ツールキットの rkCommon ライブラリーの必要条件

- [インテル® oneTBB 2021.9](#) 以降 (ツールキットでは 2021.11 が自動的にインストールされます)

インテル® インプリシット SPMD プログラム・コンパイラー (インテル® ISPC) の必要条件

- なし

インテル® レンダリング・ツールキットの rkUtil superbild スクリプトの必要条件

- CMake* 3.25 以降
- インターネット接続 (スクリプトで使用)
 - 必要に応じて、必要な基本コンポーネントを取得
 - 必要に応じて、必要なほとんどのオプション・コンポーネントを取得

インテル® OpenSWR の必要条件

注: インテル® OpenSWR は www.mesa3d.org で公開されている Mesa* OpenGL* オープンソース・コミュニティ・プロジェクトの一部として利用できます。ビルドとデブロイに関する情報は、Mesa のガイダンスを参照してください。

- インテル® OpenSWR は、Mesa* 3D グラフィックス・ライブラリー内に Gallium ドライバーとして組み込まれています。インテル® OpenSWR のビルドの必要条件と手順は、[インテル® OpenSWR のウェブサイト](#) (英語) を参照してください。通常、Meson ベースのビルドパスが推奨されます。

開発環境

サポートされるすべてのプラットフォーム

- CPU
 - C++17 以降のコンパイラーを推奨。
 - クライアント・アプリケーションには C++11 以降のコンパイラー superbild、インテル® OSPRay Studio のビルド、および通常のサードパーティー・ライブラリーの使用には C++14 以降のコンパイラー
 - インテル® oneAPI DPC/C++ コンパイラーとの互換性のため、C++17 以降 (GCC 7 以降) を使用。
 - C99 以降のコンパイラー。
- GPU
 - [インテル® oneAPI ベース・ツールキット](#)に含まれるインテル® oneAPI DPC++/C++ コンパイラーおよびランタイム 2024.0 (ツールキットの構成手順で説明)

Microsoft* Visual Studio*

- Microsoft* Visual Studio* 2019 以降を推奨

コンポーネント別の Windows* のディスク空き容量要件

注: コンポーネントには、[インテル® oneAPI スレッディング・ビルディング・ブロック \(インテル® oneTBB\)](#) など、自動的に含まれる必要条件のコンポーネント用に追加の空き容量が必要です。

コンポーネント	必要なディスク空き容量
インテル® Embree	322MB
インテル® オープン・イメージ・デノイズ	40MB
インテル® OpenSWR	バイナリーは提供されません
インテル® オープン VKL	80MB
インテル® OSPRay	90MB
インテル® OSPRay Studio	8MB
インテルの rkCommon ライブラリー	3MB
インテルの rkUtil superbuild	1MB 未満
インテル® オープン・パス・ガイディング・ライブラリー	2MB
インテル® インプリシット SPMD プログラム・コンパイラー	158MB
自動で含まれるインテル® ベース・ツールキットのコンポーネント	
インテル® oneTBB	470MB
すべてのコンポーネント	1.2GB

コンポーネントの動作環境

インテル® レンダリング・ツールキット・コンポーネントの動作環境 [オープンソース](#)

インテル® Embree (英語)	すべてのホスト (英語)
インテル® オープン・イメージ・デノイズ (英語)	すべてのホスト (英語)
インテル® オープン VKL (英語)	すべてのホスト (英語)
インテル® オープン PGL (英語)	すべてのホスト (英語)
インテル® OSPRay (英語)	すべてのホスト (英語)
インテル® OSPRay Studio (英語)	すべてのホスト (英語)
インテル® インプリシット SPMD プログラム・コンパイラー (英語)	すべてのホスト (英語)

以前のインテル® oneAPI リリース

2023.x

- [リリースノート](#) (英語)
- [動作環境](#) (英語)

関連情報

詳細は、[インテル® レンダリング・ツールキットの製品ページ](#)を参照してください。

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。

© Intel Corporation. Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、Intel Corporation またはその子会社の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

本資料は、(明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず) いかなる知的財産権のライセンスも許諾するものではありません。

本資料で説明されている製品には、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

インテルは、明示されているか否かにかかわらず、いかなる保証もいたしません。ここにいう保証には、商品適格性、特定目的への適合性、および非侵害性の黙示の保証、ならびに履行の過程、取引の過程、または取引での使用から生じるあらゆる保証を含みますが、これらに限定されるわけではありません。

製品および性能に関する情報

¹ 性能は、使用状況、構成、その他の要因によって異なります。詳細については、<http://www.intel.com/PerformanceIndex/> (英語) を参照してください。