
Windows 10 互換性ガイドブック:

2016 年 8 月 20 日

概要

このガイドブックは、Windows 10 オペレーティング システム (OS) に対するプラットフォームの変更に関する情報を開発者に提供するものです。また、既存のアプリや構想中のアプリと最新バージョンの OS の互換性を確認するためのガイドラインも提供します。このガイドブックは、以前のバージョンの Windows を十分に理解していることを前提としています。

この内容は、Windows 10 に適用されます。

このガイドブックは、Microsoft Windows 環境で使用するアプリやデバイスの設計に携わるサード パーティ開発者に向けて記述されています。

免責事項: 本書は現状有姿で提供されます。本書に記載されている URL やその他のインターネット Web サイトの参考文献を含む情報および見解は、予告なく変更されることがあります。お客様ご自身の責任でご使用ください。本書は Microsoft 製品の知的財産に関するいかなる権利をもお客様に許諾するものではありません。お客様は本書を私的な参照目的として複製および使用することができます。お客様は本書を内部の参照目的として変更することができます。

© 2016 Microsoft. All rights reserved.

Microsoft

目次

概要	3
サービスとしての Windows (WaaS) とは？	4
Windows 10 のリリースの種類と更新間隔	5
サービスとしての Windows におけるアプリのサポート	7
互換性の確保に関連する Windows 7 以降の主な変更点	9
Windows 10 を強化するためにデータを使用	11
複合的なエコシステムにおける互換性の確保方法	12
Windows のバージョン チェック	13
文書化されていない API	17
UWP アプリと Centennial アプリの開発	18
ウィンドウ モードで実行しない場合、モダン アプリの機能に対する影響	20
Windows Update での適切なグラフィックス ドライバーの入手	22
Windows 10 へのグラフィックス ドライバーの移行	24
Bluetooth ドライバー	28
SAM RPC プロトコルを使用するリモート呼び出しは、ローカルの Administrators グループに限定される	29
未署名のドライバに対するカーネル チェック	30
Credential Guard による サード パーティ製セキュリティのサポート	31
最適化されたテスト戦略とフライティング	32
手順 1. Windows Insider になってフライティングに参加する	33
手順 2. シナリオをテストする	35
手順 3. フィードバックを提供する	37
手順 4. Windows 10 に登録する	38
まとめ	39

概要

Windows 10 には、世界中のアプリ/ドライバー開発者や企業が使用できる最新の OS テクノロジとソフトウェア開発プラットフォームが導入されています。Windows のセキュリティ、信頼性、パフォーマンス、ユーザー エクスペリエンスをさらに向上するために、Microsoft はさまざまな新機能の追加、既存の機能の強化、その他の機能の廃止を行っています。

Windows 10 は、過去にリリースされた Windows OS バージョン向けに作成されたアプリとの互換性を保つことを目標としています。しかし、革新やセキュリティの強化、信頼性の向上などから生じる、ある程度の互換性の喪失は避けて通れません。全体としては、最新バージョンの Windows OS と既存のアプリ/デバイスとの互換性は、高く保たれています。

このドキュメントでは、アプリやデバイスに最新バージョンの OS との互換性があることを確認する方法を説明し、アプリの互換性に関する既知の問題の概要を紹介します。

アプリを最適化して最新バージョンの Windows の機能を活用する方法については、以下のトピックをご覧ください。

「[Windows 互換性ガイドブック \(英語\)](#)」は、Windows 8 と Windows Server 2012 を解説しています。また、同じように Windows 8.1 と Windows Server 2012 R2 を解説したガイドブックも提供していますので、参考資料として参照してください。

サービスとしての Windows (WaaS) とは？

現在の環境では、ユーザーはデバイス中心のエクスペリエンスに期待を抱くことが多いため、完全な製品サイクルを年ごとではなく月ごとで測定する必要があります。また、新しいリリースは頻繁に入手可能にする必要があり、ユーザーへの影響を最小限に抑えて展開できる必要があります。Microsoft は革新的な機能を開発および配信するための新しいアプローチを実装することで、これらの要件を満たすように Windows 10 を設計しました。このアプローチは、[サービスとしての Windows \(WaaS\)](#) と呼ばれます。高い品質レベルを維持しながら、非常に短い製品サイクルを実現するためにキーとなるのは、Microsoft が Windows 10 向けに実装したテストに対するコミュニティ中心の革新的なアプローチです。Windows Insider として知られるこのコミュニティは、世界中の何百万ものユーザーで構成されています。コミュニティが Windows Insider にオプトインすると、製品サイクルの経過と共に数多くのビルドをテストし、フライティングという反復的な方法で Microsoft にフィードバックを提供します。

フライトとして配布されるビルドは、実際の使用中にビルドが適切に実行されるかどうかに関する重要なデータを Windows エンジニアリング チームに提供します。また、Windows Insider によるフライティングを利用することで、Microsoft は以前よりはるかに多様なハードウェア、アプリケーション、およびネットワーク環境でビルドをテストし、より早く問題を特定することができます。その結果、Microsoft はコミュニティに重点を置いたフライティングによって、これまでより速いペースで革新的機能を配信でき、より高品質な一般リリースを公開できると考えています。

Windows 10 のリリースの種類と更新間隔

Microsoft は Windows Insider にフライト ビルドをリリースしますが、それに加えて継続的に次の 2 種類の Windows 10 リリースを広く一般に公開します。

Feature updates (機能更新プログラム) は、既に Windows 10 を実行しているデバイスに、最新の機能とエクスペリエンスをインストールします。機能更新プログラムには Windows の全体のコピーが含まれているため、Windows 7 または Windows 8.1 を実行している既存のデバイス、およびオペレーティング システムがインストールされていない新しいデバイスに Windows 10 をインストールするためにも使用できます。平均して 1 年間に 1 ～ 2 個の新しい機能更新プログラムを公開することを想定しています。

Quality updates (品質更新プログラム) は、セキュリティ問題の解決策や、その他重要なバグ修正を提供します。品質更新プログラムは、現在サポート中の各機能を強化するため、1 か月に 1 回～複数回のペースで提供することを想定しています。Microsoft は、引き続き Update Tuesday (Patch Tuesday と呼ばれます) に品質更新プログラムを公開していきます。また、Microsoft はユーザーのニーズへの対応が必要なとき、Update Tuesday のプロセス以外で Windows 10 向けの追加の品質更新プログラムを公開することがあります。

Windows 10 の開発中、ユーザーが求める機能やエクスペリエンスを Microsoft がこれまでよりすばやく提供できるように、Windows 製品のエンジニアリングおよびリリース サイクルを合理化しました。また、機能更新プログラムと品質更新プログラムを配信およびインストールするための新しい手段を開発しました。これにより、展開および継続的な管理が簡略化され、最新の Windows の機能とエクスペリエンスによって最新の環境に保たれる従業員の PC 基盤が広がり、総保有コストが削減されます。さらに、Current Branch (CB)、Current Branch for Business (CBB)、および Long-Term

Servicing Branch (LTSB) と呼ばれる新しいサービス オプションを実装しました。これらは、エンタープライズ環境で以前より多くのデバイスをより新しい状態に保つための実地的なソリューションを提供します。

以下の表に、さまざまなサービス ブランチと、その主な属性を示します。

サービス オプション	インストールにおける 新しい機能アップ グレードの利用可能性	主な利点	サポートされる エディション
Current Branch (CB)	Microsoft が最初に発行した直後	できるだけ早く新機能をユーザーに利用可能にする	Home, Pro, Education, Enterprise, Mobile, IoT Core, Windows 10 IoT Core Pro (IoT Core Pro)
Current Branch for Business (CBB)	Microsoft が最初に発行してから約 4 か月後	展開の前に新しい機能アップグレードをテストするための追加時間を提供する	Pro, Education, Enterprise, Mobile Enterprise, IoT Core Pro
Long-Term Servicing Branch (LTSB)	Microsoft が発行した直後	変更の少ない構成で選択された Windows 10 リリースの長期的な展開を可能にする	Enterprise LTSB

詳細については、「[更新プログラムおよびアップグレードに関する Windows 10 のサービス オプション](#)」(英語) をご覧ください。

サービスとしての Windows におけるアプリのサポート

従来、アプリのサポートについては、Windows がリリースされるタイミングに合わせて新しいバージョンのアプリをリリースするという手法が取られていました。これは、基盤となる OS に重大な変更があり、アプリに不具合が生じることを見越したものです。このモデルでは、専用の開発および検証サイクルを確保しなければならないので、ISV パートナーが Windows のリリースのペースに合わせる必要がありました。

WaaS モデルでは、基盤となる OS の互換性を維持します。つまり Microsoft は、アプリのエコシステムに悪影響を及ぼす重大な変更がないように努めます。このシナリオでは、新しい Windows ビルドがリリースされたとき、ほとんどのアプリ (カーネルの依存関係がないもの) が機能し続けます。

この変更を踏まえ、Microsoft は、アプリのリリースとサポートを特定の Windows ビルドから切り離すことを ISV パートナーに推奨しています。アプリ ライフサイクル アプローチを取れば、新しいバージョンのアプリがリリースされたら Windows ビルドのリリースに関係なく一定期間サポートされることになるため、Microsoft と ISV の共通顧客にさらに良いサービスを提供することが可能になります。ISV は、そのバージョンのアプリがライフサイクルでサポートされる限り、サポートを提供し続けます。Microsoft も、同様のライフサイクル アプローチを Windows に取っています。詳細については[こちら](#)からご覧ください。

このアプローチは、アプリのスケジュールを Windows のリリースに合わせて続けることの負荷を軽減します。ISV パートナーは、自社のペースで機能や更新プログラムをリリースできるので、Windows のリリースを考慮することなく、最新のアプリの更新プログラムで顧客ベースを常に最新の状態に保つことが可能になります。さらにユーザーも、Windows ビルドがリリースされたときに明示的なサポート情報を探す必要がなくなります。以下に、アプリ

が複数のバージョンの OS でどのようにサポートされるかを説明したサポート情報の例を示します。

アプリケーション ライフサイクル

ソフトウェア開発企業の *Contoso* は、エンタープライズ分野で大きなシェアを持つ人気アプリ “*Mojave*” を所有しています。*Contoso* は、次期メジャーリリースとなる *Mojave 14.0* をリリースし、リリース日から 3 年間にわたりメインストリーム サポートを提供します。メインストリーム サポート期間中は、すべての更新プログラムとサポートを、ライセンス済みの製品に無償提供します。また、サポートは 2 年間延長し、その猶予期間内にユーザーは更新プログラムとサポートを購入することができます。延長サポートの終了日を過ぎると、この製品バージョンはサポートされなくなります。メインストリーム サポート期間中、*Contoso* はリリース済みの *Windows* ビルドすべてで *Mojave 14.0* をサポートします。また、*Windows* の製品リリースとは関係なく、必要に応じて *Mojave* の更新プログラムをリリースします。

以下のセクションでは、基盤となる OS の互換性を維持するために Microsoft が採用している手段について、さらに詳しく説明します。また、統合された OS とアプリ エコシステムの互換性を維持するために開発者の皆さんができることについても扱います。さらに、*Windows* フライト ビルドを活用して *Windows* ビルドがリリースされる前にアプリの不具合を検出する方法についてのセクションもあります。最後に、Microsoft がインストールメントとテレメトリを利用して *Windows* ビルドの質を高めている方法を説明します。Microsoft は ISV に、これと類似した方法をアプリ ポートフォリオに採用することを推奨しています。

互換性の確保に関連する Windows 7 以降の主な変更点

互換性は開発者にとって重要な問題です。ISV や開発者は、サポートされている Windows OS のバージョンすべてで、アプリが期待どおり動作することを保証する必要があります。ユーザーや企業は、互換性に大きな投資をしています。その理由は、購入したアプリが確実に機能し続けることを望んでいるからです。互換性は、購入を決定する重要な指標です。ベスト プラクティスに基づいて適切に作成されるアプリは、新しい Windows バージョンがリリースされたときにコードへの影響がかなり少なく抑えられ、断片化が少なくなります。このようなアプリは、保守のためのエンジニアリング投資が少なく、短期間で市場投入できます。

Windows 7 の時代、互換性には事後的に対応していました。Windows 8 では別のアプローチを取るようになりました。事後対応するのではなく、Windows 内の設計によって互換性を実現するように変わったのです。Windows 10 は、これまでの OS の中で設計による互換性が最も実現されている OS です。互換性実現に利用した主だった方法のいくつかを以下に示します。

- アプリのテレメトリ: Windows エコシステムにおけるアプリの人気度を把握し、互換性テストへの情報提供に役立てています。
- ISV とのパートナーシップ: 外部パートナーと直接連携して、データの提供や、ユーザーに発生した問題の解決に協力しています。
- 設計レビュー、早期検出: 機能チームと連携して、互換性を損なう Windows の変更の数を減らしています。互換性レビューは機能チームが必ず通過する関門です。
- API の変更の厳密な制御とコミュニケーションの強化
- フライティングとフィードバックの繰り返し: Windows Insider 参加者にフライト ビルドを提供し、互換性の問題の発見にご協力ください

た後、ユーザー向けの最終ビルドをリリースしています。このフィードバック プロセスによって、バグが明らかになるだけでなく、ユーザーの求める機能が搭載されていることが保証されます。

Windows 10 を強化するためにデータを使用

Microsoft は、問題の特定とトラブルシューティング、製品とサービスの向上、個々のユーザーに合ったエクスペリエンスの提供を実現するために、診断データと使用状況データを活用しています。使用状況データの収集範囲は、Windows エコシステムのコンピューターが実行しているアプリにまで及びます。ユーザーが使用しているものに基づいて、新しいバージョンの Windows OS に対してテストするアプリ、デバイス、およびドライバーのリストを作成します。Windows 10 は、これまでの OS の中で設計による互換性が最も実現されている OS で、数千個の人気アプリの 90% 以上と互換性があります。Windows 互換性チームは一般に、問題を発見したら ISV パートナーにフィードバックを提供して、解決策のために協力できるようにしています。共通する顧客が、生産性向上や娯楽を目的とした OS の機能や、アプリの機能を失わずに、Windows をシームレスにアップデートできるのが理想です。

複合的なエコシステムにおける互換性の確保方法

以下に示す Microsoft お勧めのベスト プラクティスを利用して、アプリと Windows 10 の互換性を確保できます。

Windows のバージョン チェック

OS バージョンは、Windows 10 OS のリリースと共にインクリメントされました。つまり、Windows 10 の内部バージョン番号も 10.0 に変更されたということです。Microsoft では、これまでと同様、OS バージョンの変更後も、アプリとデバイスの互換性が維持されるよう、さまざまな対策を尽くしています。(カーネルの依存関係がない) ほとんどのアプリのカテゴリでは、この変更によってアプリの機能に悪影響が及ぶことはありません。既存のアプリは Windows 10 で引き続き問題なく機能します。

具体的な影響

この変更による具体的な影響は、アプリ固有です。つまり、OS のバージョンをチェックするアプリは、大きなバージョン番号を取得することになるため、以下に示す状況が 1 つ以上発生する可能性があります。

- アプリ インストーラーでアプリをインストールできず、アプリが起動しない
- アプリの動作が不安定になる、クラッシュする
- エラー メッセージが表示されるが、アプリは適切に機能する

アプリによっては、バージョン チェックを実行して、ユーザーに警告を表示するだけのものもあります。しかし、非常に密接にバージョン チェックに結びついているアプリもあります (ドライバーや、検出回避のためのカーネルモードの使用)。このような場合、不適切なバージョンが検出されるとアプリは動作しません。バージョン チェックではなく、次のいずれかのアプローチを取ることをお勧めします。

- アプリが特定の API 機能に依存している場合、適切な API バージョンをターゲットにします。
- NTDDI (NT デバイス ドライバー インターフェイス) バージョン番号は、API のターゲット機能が変わる場合にのみ増加します。APISet や他に機能チームが作成して公開している API を通じて変更を検出する

ようにし、機能や修正プログラムのプロキシとしてバージョンを使用しないようにします。重大な変更があり、適切なチェックが公開されていない場合、それはバグになります。

- アプリのバージョン チェック方法が不適切でないことを確認します。不適切とは、レジストリ、ファイル バージョン、オフセット、カーネル モード、ドライバなどを利用する方法を指します。アプリでバージョンのチェックが必ず必要な場合は、GetVersion API を使用します。この API は、メジャー バージョン番号、マイナー バージョン番号、およびビルド番号を返します。
- GetVersion API を使用する場合、この API の動作は Windows 8.1 以降に変更されたことに注意してください。詳細については、[API のドキュメント \(英語\)](#) を参照してください。
- マルウェア対策アプリやファイアウォールがある場合は、通常のフィードバック チャンネルと Windows Insider Program を通じて対処します。

アプリ マニフェスト

アプリ マニフェストの例を次に示します。

XML

```
<exe>.manifest
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1" xmlns:asmv3="urn:schemas-microsoft-com:asm.v3">
  <assemblyIdentity
    type="win32"
    name=SXS_ASSEMBLY_NAME
    version=SXS_ASSEMBLY_VERSION
    processorArchitecture=SXS_PROCESSOR_ARCHITECTURE
  />
  <description> my app exe </description>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel
          level="asInvoker"
          uiAccess="false"
        />
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

```
        </security>
    </trustInfo>
    <compatibility xmlns="urn:schemas-microsoft-
com:compatibility.v1">
        <application>

            * <!-- Windows 10 -->
            * <supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-
48fd50a15a9a}"/>
                <!-- Windows 8.1 -->
                <supportedOS Id="{1f676c76-80e1-4239-95bb-
83d0f6d0da78}"/>
                <!-- Windows Vista -->
                <supportedOS Id="{e2011457-1546-43c5-a5fe-
008deee3d3f0}"/>
                <!-- Windows 7 -->
                <supportedOS Id="{35138b9a-5d96-4fbd-8e2d-
a2440225f93a}"/>
                <!-- Windows 8 -->
                <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-
d69d4a4a6e38}"/>
            </application>
        </compatibility>
    </assembly>
```

以下の変数をソースに追加します。

SXS_MANIFEST_RESOURCE_ID=1

SXS_MANIFEST=foo.manifest

SXS_ASSEMBLY_NAME=Microsoft.Windows.Foo

SXS_ASSEMBLY_VERSION=1.0

SXS_ASSEMBLY_LANGUAGE_INDEPENDENT=1

SXS_MANIFEST_IN_RESOURCES=1

Windows 10 では、上記でアスタリスク (*) の付いている 2 行がアプリケーションで Windows 10 バージョンの OS を正確にターゲットにする方法を示します。Windows 10 用の .exe の記述は、以前のバージョンの Windows OS を実行している場合にはまったく影響しません。 .rc ファイルを既に定義している場合には、これを .rc ファイルにも追加できます。

trustInfo の追加は必須ではありませんが、追加することを強くお勧めします。追加すると、OS が Windows 10 と Windows 8.1 のどちらの場合でも、.exe が常に適切なバージョンを取得できるようになります。

リソース

- [Application Compatibility Toolkit のダウンロード: Windows ADK for Windows 10 のダウンロード](#)
- [既知の互換性修正プログラム、互換モード、および AppHelp メッセージ](#)
- [Version Helper API \(英語\)](#)

文書化されていない API

アプリでは、文書化されていない Windows API を呼び出すことや、特定の Windows ファイルのエクスポートやレジストリ キーと依存関係を持つことがないようにします。これに従わない場合、機能が動作しない、データが喪失する、潜在的なセキュリティ問題が発生するなどのおそれがあります。アプリに必要な機能で利用できないものがある場合は、通常のフィードバックチャネルや Windows Insider Program を通じてフィードバックをお寄せください。

UWP アプリと Centennial アプリの開発

Microsoft では、すべての Win32 アプリの ISV に[ユニバーサル Windows プラットフォーム \(UWP\)](#) アプリ、(特に) [Centennial アプリ](#)の開発を進めるよう推奨しています。このようなアプリ パッケージの開発には、従来の Win32 インストーラーを使用する以上に大きなメリットがあります。UWP アプリは [Windows ストア](#)でもサポートされているため、簡単にユーザーのアプリを一貫性のあるバージョンに更新して、サポート コストを抑えることができます。

Win32 アプリの種類が Centennial モデルで使用できない場合は、適切なインストーラーを使用していること、アプリが十分にテストされていることを確認することを強くお勧めします。アプリを利用するとき、ユーザーや顧客が最初に使用するのがインストーラーであるため、インストーラーが適切に機能することを確認してください。残念なことに、アプリが機能しないことや、すべてのシナリオに対して十分にテストされていないことが少なくありません。[Windows アプリ認定キット](#)を使用すると、Win32 のインストールとアンインストールのテストが容易になり、ユーザーに提供する前に文書化されていない API の使用の特定や、その他の基本的なパフォーマンス関連のベスト プラクティスの問題の特定が簡単になります。

ベスト プラクティス

- 32 ビット バージョンと 64 ビット バージョンの両方の Windows で機能するインストーラーを使用する。
- 複数のシナリオ (ユーザーや PC のレベル) で実行できるようにインストーラーを設計する。
- すべての Windows 再配布可能パッケージを元のパッケージに保つ。そうしないで再パッケージ化した場合、インストーラーが機能しなくなる可能性があります。

-
- インストーラーの開発時間をスケジュールに組み込む。なぜなら、ソフトウェア開発ライフサイクル中、インストーラーは成果物として見落とされがちだからです。

ウィンドウ モードで実行しない場合、モダン アプリの機能に対する影響

Windows 10 では、モダン アプリは既定で全画面表示されなくなりました。その代わりに、ウィンドウ表示となり、最小化、元のサイズに戻す、最大化、サイズ変更などの操作 (およびすべてのクラシック アプリのウィンドウに対して常に実行できた他の操作) を行えるようになっています。

具体的な影響

最も顕著な変更は、サイズを画面のサイズや高さに合わせるのではなく、任意のサイズに変更できるようになったことです。アプリ ウィンドウのサイズは、(アプリの最小ウィンドウ サイズまでの) 好きなサイズに連続的に変更できます。これは、Windows 8 や Windows 8.1 とは異なります。これらの OS では、ウィンドウのサイズ変更操作は個別のアクションでした (ウィンドウのサムネイルのサイズを変更し、そのアクションが完了すると、ウィンドウのサイズが変更される)。その代わりに、現在は、ウィンドウの下隅 (または他の境界線の位置) をドラッグすると、ウィンドウのサイズが連続的に変化します。アプリではサイズを変更するのではなく、連続でサイズ変更メッセージを受け取ります。

対策

Windows 8.0 および Windows 8.1 アプリでこれに対処するには、以下を行います。

- アプリの機能として期待される機能が動作しない場合、その解決策は、アプリを全画面表示モードにする (タイトル バーの右上隅にある [最大化] ボタンを使用する) ことです。
- アプリの起動に影響があり、期待どおりにアプリが開かない場合、タブレット モードに切り替えると煩わしい操作なしでアプリを強制的に全画面表示モードで起動することができます。

これに対処する最良の方法は、モニターのサイズに一致しない場所や高さにサイズ変更できることを認識するようにアプリを変更することです (ハードコードされた高さ/幅のテーブルを用意せず、それらを想定するだけにするか、ハードコードされた比率を想定するようにします)。アプリ開発者が注意すべきなのは、アプリのサイズが変更されている間、現在のサイズ変更メッセージが送信された直後に別のサイズ変更メッセージが発生する可能性があることです。そのため、サイズ変更中にアプリでアニメーションを開始する場合、アプリはその直後に別のサイズ変更メッセージを受信する可能性があります (そのため、このような状況でアプリがクラッシュしないようにすることが重要です)。

注 モダン アプリとは、Windows 8 / Windows 8.1 で Windows ストア アプリと呼ばれています。一方で、タブレット モードで従来のデスクトップ アプリ (クラシック Windows アプリ)を実行した場合は、**強制的に全画面表示になることにも注意**してください。

Windows Update での適切なグラフィックス ドライバーの入手

Windows Update (WU) でグラフィックス ドライバーを入手できるかどうかによって、Windows 7、Windows 8、または Windows 8.1 から Windows 10 へのアップグレードが自動的に提供されるかどうかが決まります。ハードウェア スキャンによって、PC のグラフィックス アダプターに対応するグラフィックス ドライバーが Windows Update から入手できないことがわかった場合、更新された Windows 10 OS は提供されません。ただし、その場合でもユーザーは他のソースから Windows 10 を入手して、手動でアップグレードを実行することができます。

ハードウェアに利用可能なグラフィックス ドライバーがないとアップグレード アドバイザーに表示されていても、それが他のユーザーにはアップグレードが提供されていて自分には提供されていない理由だと確信が持てない場合があります。

さらに、その状態でアップグレードを強制的に実行すると、ハードウェア ドライバーがないために、グラフィックス機能とパフォーマンスが低下することになります。

対策

この問題への対処として、ユーザーは IHV または OEM の Web サイトにアクセスして、以前 (Windows 7、Windows 8、または Windows 8.1) のドライバを明示的にダウンロードして、手動でインストールできます。これは、アップグレードの後に行う必要があります。また、効果を発揮する保証はありません。適切なグラフィックス ドライバーが WU で提供されていない場合、サポートされている解決策はありません。ユーザーは以下のいずれの対応を取るかを判断する必要があります。

- 以前の OS に保つ。

-
- ソフトウェア ドライバー (Microsoft 基本ディスプレイ アダプター (MSBDA)) の制限を受け入れる。
 - サポートされた Windows 10 ドライバーのある新しいグラフィックス カードを購入する。

解決策

重要なのは、IHV および OEM が、サポート予定のすべてのハードウェアに対する Windows 10 グラフィックス ドライバーを WU にアップロードすることです。

生産が終了しているハードウェアのドライバーは WU にない場合があることに注意してください。この場合の解決策はなく、ユーザーは上記の「対策」の下にあるオプションのいずれかを選択する必要があります。

Windows 10 へのグラフィックス ドライバーの移行

Windows 10 メディアと Windows 10 には、その前身の Windows 8.1 と同様に、Windows メディア キット (「インボックス」) にサード パーティのグラフィックス ドライバーは用意されていません。代わりに、幅広いデバイスに対応したグラフィックス ドライバーが、WU に用意されています。そのため、ハードウェア ベンダーはオペレーティング システム イメージを変更することなくドライバーを更新できます。また、Windows 7、Windows 8、または Windows 8.1 から Windows 10 に OS をアップグレードする場合、既存のドライバーは Windows 10 には移行されません。これは Windows Server 2012 からのアップグレードにも影響します。

アップグレードとインストール

アップグレードおよび新しいインストールを行う場合、グラフィックス ドライバーは Windows Update (WU) または関連するハードウェアの IHV や OEM の Web サイトから取得する必要があります。これにはインターネット接続が必要です。Windows 7、Windows 8.x システムから Windows 10 にアップグレードする場合、動的更新 (DU) によって WU のドライバーが OS セットアップに挿入されます。

注 これは、小売店で購入する市販の PC など、事前に Windows がインストールされている PC には当てはまりません。このような PC には OEM によってグラフィックス ドライバーが既にインストールされています。OEM がドライバーを含む (OS の再インストール用の) DVD を提供する場合もあります。

Windows 10 にアップグレードした後、PC にグラフィックス ドライバーがインストールされていないことが明らかになる場合があります。その原因はいくつかあります。

-
- アップグレードではなく、クリーン インストールを選択した。
 - アップグレード中に更新を確認するオプションをオフにして、動的更新 (DU) を実質的に無効にした。
 - アップグレード中にインターネット接続が機能していなかった。
 - ドライバーのインストールが失敗した。

OS をクリーン インストールした後は、WU クライアントが自動的に起動して適切なドライバーをダウンロードするまで、PC にグラフィックス ドライバーが存在しません。その間、PC は Microsoft 基本ディスプレイ アダプター (MSBDA) を実行します。このアダプターは、複数のモニターをサポートしていないなどの機能制限があります。また、ユーザーは、フレーム レートが低い、ビデオの再生が途切れるなど、ハードウェア ドライバーと比較してパフォーマンスが低いと感じる可能性があります。

具体的な影響

古い PC (通常は Windows 7 より前に作られた PC) の場合、グラフィックス アダプターの生産が終了して、ハードウェア ベンダーのサポートが提供されなくなったため、WU に Windows 10 用のドライバーが存在しない可能性があります。前の OS からアップグレードして現在 Windows 7 または 8.x を実行している PC でも、生産が終了したグラフィックス アダプターが使われている可能性があります。

新しい PC でも、ハードウェアのアップグレードなどでグラフィックス アダプターを古い PC から移し変えたために、利用可能なドライバーが存在しないことがあります。多くの場合、このような状況が発生するのは、新しい PC を購入した際に複数のディスプレイを使用するために古いグラフィックス カードを残して、複数のグラフィックス アダプターを搭載した PC です。

他にも、ごく一部の PC には、Windows Update に「カバレッジ」ドライバーしかない可能性があります。対象範囲 (カバレッジ) ドライバーとは、OEM によるカスタマイズが行われていない汎用的なドライバーを指します。ユー

ザーが Windows 10 へアップグレードした後にそのようなドライバーを入手した場合、画面の明るさ調節用のファンクション キーがまったく機能しないなど、いくつかの機能を利用できないおそれがあります。

対策

- 適切なグラフィックス ドライバーをアップグレード プロセス中の DU かアップグレード完了直後の WU で提供する必要があります。OEM は、出荷時の Windows 10 のインストールに使用する PC 上のシステム イメージに、適切なグラフィックス ドライバーを含める必要があります。
- ユーザーはアップグレード後、[設定]、[更新とセキュリティ]、[Windows Update] でドライバーを明示的に確認できます (任意)。バックグラウンドでドライバーが自動的にインストールされているときに確認を強制する場合、自動インストールが先に完了すると、ドライバーのインストールの失敗が表示されることがあります。これは無視してもかまいません。
- Windows 10 をクリーン インストールする予定のユーザーは、インストール前に関連するドライバーを取得するか、後で Windows Update を使用してドライバーを取得するようにします。後者の場合は、インターネット接続が機能していることを確認する必要があります。
- 対象範囲 (カバレッジ) ドライバーを受け取るコンピューターに関しては、欠如した機能への対策はありません。ただし、これが起こるのは、ハードウェアのサプライヤーがドライバーを保守していない場合 (数年前に発売された PC の場合) のみです。

注 ノート PC など、ディスプレイが 1 つの PC では MSBDA で十分なため、多くのユーザーはハードウェア ドライバーの欠如に気が付きません。この場合、対策の必要はありません。

解決策

重要なのは、IHV および OEM が、サポート予定のすべてのハードウェアに対する Windows 10 グラフィックス ドライバーを WU にアップロードすることです。

アップグレードの際は、[更新プログラムの確認] を有効 (既定の設定) のままにする必要があります。ネットワーク接続の速度や WU サーバーでの読み込みによっては、OOBE(Out of Box Experience) が完了して初めてログインするまでドライバーがインストールされないことがあります。その間、いくつかの機能が制限されたり、パフォーマンスが低下する可能性があります。

メディア (DVD またはフラッシュ ドライブ) を使用してアップグレードする場合でも、アップグレードの前に、インターネット接続が機能していることを確認する必要があります。

- PC がインターネットに接続されている場合、自動的に適切なドライバーがダウンロードされ、インストールされます。ユーザーは何も操作する必要はありません。
- PC がインターネットに接続されていない場合、ドライバーはインターネットに接続した PC を使用して IHV または OEM の Web サイトからダウンロードし、フラッシュ ドライブや CD/DVD を使用して目的のコンピューターに移し替えた後、手動でインストールする必要があります。

Bluetooth ドライバー

IHV と企業顧客は、Windows 7、Windows 8、Windows 8.1 からのアップグレードの一環として Bluetooth ドライバーとサードパーティ プロファイル パックが Windows 10 に移行されない点に注意する必要があります。

Windows 10 における Bluetooth の変更の詳細については、[こちらの記事](#)を参照してください。

SAM RPC プロトコルを使用するリモート呼び出しは、ローカルの Administrators グループに限定される

この情報は、早期の情報提供を目的にしており、プレ リリースされた製品に基づいており、製品リリース前に変更されることがあることに注意してください。]

SAM RPC(Security Account Manager Remote Procedure Call) プロトコルは、制限されています。これは、ローカルの Administrators グループのメンバーだけが、このプロトコルを使用したリモート呼び出しを行えることを意味します。Active Directory ドメイン コントローラーは、この影響を受けません。

対策

管理者権限を付与するユーザーを、正しく管理してください。アクセス権チェックに使用する ACL は、グループ ポリシーを使うことで構成することができます。

未署名のドライバーに対するカーネル チェック

この情報は、早期の情報提供を目的にしており、プレ リリースされた製品に基づいており、製品リリース前に変更されることがあることに注意してください。]

クリーン インストールされた Windows 10 デバイスは、セキュア ブートが有効になっています (セキュア ブートは、Windows 8 でリリースされた機能であることに注意してください)。新しいすべてのドライバーは、クロス署名用の証明書 (cross-signed certificates) よりも WHQL/Sysdev によって署名されている必要があります。例外としては、このポリシーが発行される前にクロス署名用の証明書で署名されたドライバーがインストールされたデバイスを、アップグレードしたケースです。このケースは、後方互換性の観点から例外としています。この新しいポリシーは、2015 年 4 月に発表されましたが、2016 年 8 月にリリースされた Windows Anniversary Edition で強制されるようになりました。

未署名のドライバーは、証明書の要件を満たさないことから PCA (プログラム互換アシスタント) 通知が表示されます。ユーザー エクスペリエンスの低下を防ぐために、カーネルが透過的にドライバーをブロックします。

詳細については、ドライバーの署名変更アナウンスの[ブログ \(英語\)](#) を参照してください。

注 WHQL は、Windows Hardware Quality Laboratory の略であり、ハードウェアやドライバーの品質を向上させる取り組みになります。Sysdev は、System Developer の略であり、[ハードウェア デベロッパー センター](#)へ登録して、ドライバーを提出して審査に合格することで、WHQL による署名が行われます。

Credential Guard による サード パーティ製セキュリティのサポート

Credential Guard は、サード パーティ製の SSP(Security Support Provider) による LSA (ローカルセキュリティ機関) に対するパスワード ハッシュへのリクエストを許可しません。しかしながら、SSP と AP(Authentication Package) は、パスワード変更やユーザーがログ オンした時の通知を受け取ることができます。カスタムな SSP と AP が、文書化されていない API を使用することをサポートしません。

Credential Guard による保護領域が拡大すると、進化する Windows 10 リリース (WaaS) によって、以前の動作に影響を及ぼすことがあります。たとえば、Credential Guard は、脆弱性を使ったマルウェアによる攻撃からコンポーネントを保護し、クレデンシャルの使用を防止します。それ故に、Credential Guard を実行するデバイスのアップグレードには、十分なテストを行うことを推奨します。

詳細は、推奨する対策を説明する[記事 \(英語\)](#) を参照してください。

最適化されたテスト戦略とフライティング

Windows OS のフライティングとは、最終ビルドを一般リリースする前に Windows Insider に提供する、暫定的なビルドのことです。より多くの Insider にこれらの暫定的なビルドを提供すれば、ビルドの品質や互換性などに関して得られるフィードバックも増えるので、最終ビルドの質向上に役立ちます。このフライティング プログラムに参加すれば、OS の反復的なビルドで自作のアプリが期待どおりに機能することを確認できます。また、これらのフライト ビルドがどのくらい適切に機能しているか、どのような問題に遭遇したかなどについてフィードバックのご提供をお願いしています。アプリを Windows ストアで販売している場合、Windows ストアを通じてアプリをフライティングできるので、Windows Insider にアプリをインストールしてもらうことが可能です。アプリをインストールしたユーザーから、アプリについてリリース前のフィードバックを受け取ることができます。以下のセクションでは、Windows フライト ビルドに対してアプリをテストする手順の概要を示します。

手順 1. Windows Insider になってフライティングに参加する

[Windows Insider](#) になれば、Windows の未来と一緒に作ることができます。Microsoft は、Insider から提供されるフィードバックを、Windows プラットフォームの機能強化に役立てています。Windows Insider は、熱心なユーザーとつながったり、フォーラムに参加したり、アドバイスを交換したり、Insider 限定のイベント情報を得たりすることのできる、活気にあふれたコミュニティです。

Windows 10、Windows 10 Mobile、最新の Windows SDK とエミュレーターのプレビュー ビルドをダウンロードできるので、あらゆるツールを思いのままに使いこなして優れたアプリを開発したり、ユニバーサル Windows プラットフォームや Windows ストアの最新情報を確認したりすることができます。

ハードウェア開発キットのプレビュー ビルドを入手して、Windows 用ユニバーサル ドライバーの構築を開始する絶好の機会でもあります。

また、サポートされている IoT 開発ボードに Windows IoT Core Insider Preview をインストールできるので、ユニバーサル Windows プラットフォームを使用して優れた接続ソリューションを構築することも可能です。

Windows Insider Program に参加する前に、対象ユーザーについて以下で確認してください。

- 開発中のソフトウェアを試したいと考えている。
- ソフトウェアとプラットフォームに関するフィードバックを提供したいと考えている。
- 多くの更新プログラムに対応でき、途中で UI デザインが大幅に変わる可能性があることも承知している。
- PC に関する詳しい知識があり、問題のトラブルシューティング、データのバックアップ、ハード ドライブのフォーマット、オペレーティン

グ システムのクリーン インストールが苦にならず、必要に応じて以前のオペレーティング システムを復元できる。

- ISO ファイルが何であるかとその使い方を理解している。
- 日常的に使うコンピューターやデバイスにはインストールしない。

手順 2. シナリオをテストする

フライト ビルドに更新したら、以下のサンプル テスト ケースを試して、テストとフィードバックの収集を開始します。これらのテストのほとんどで、x86 システムと x64 システムを確実にカバーするようにしてください。

新しい OS ビルドでアプリをテストします。

クリーン インストール テスト: Windows 10 のクリーン インストールでアプリが完全に機能することを確認します。アプリがこのテストとアップグレード テストに失敗する場合、おそらく問題の原因は、基盤となる OS の変更か、アプリのバグです。調査してみて問題の原因が前者だった場合は、Windows Insider Program を使用してフィードバックを提供し、解決策の特定にご協力ください。

アップグレード テスト: アプリが以前のバージョンの Windows (Windows 7 や Windows 8.1) から Windows 10 にアップグレードした後に機能するかどうかを確認します。アプリはアップグレード中にロールバックを引き起こさず、アップグレード後も期待どおりに機能しなければなりません。これは、シームレスなアップグレード エクスペリエンスを実現するために不可欠です。

再インストール テスト: コンピューターを以前のバージョンの OS から Windows 10 にアップグレードした後に、アプリを再インストールすることでアプリの機能を復元できるかどうか確認します。アプリがアップグレード テストに合格せず、その問題の原因を絞り込むことができない場合、再インストールすれば失われた機能を復元できる可能性があります。再インストール テストに合格するということは、アプリの一部が Windows 10 に移行されていない可能性があります。

アプリを OS / デバイス機能に対してテストする

アプリが OS の特定の機能に依存している場合でも期待どおりに機能するかどうかを確認します。一般的には、以下の領域をテストします。多くの場合、カバー率を上げるため、一般的に使用されている PC モデルを対象に実行します。

- オーディオ
- USB デバイスの機能 (キーボード、マウス、メモリー スティック、外付けハード ディスクなど)
- Bluetooth
- グラフィック/ディスプレイ (マルチモニター、プロジェクター、画面の向きなど)
- タッチ スクリーン (方向、スクリーン キーボード、ペン、ジェスチャなど)
- タッチパッド (左/右ボタン、タップ、スクロールなど)
- ペン (シングル/ダブル タップ、押下、長押し、消しゴムなど)
- 印刷/スキャン
- センサー (加速度計、フュージョンなど)
- カメラ

手順 3. フィードバックを提供する

フライト ビルドに対するアプリのパフォーマンスについて、Microsoft にお知らせください。テスト中にアプリの問題を発見したら、パートナー ポータル (アクセスできる場合) でバグを記録するか、Microsoft の担当者にバグを報告してください。ユーザーに上質なエクスペリエンスを構築するため、ぜひこれらの情報を提供し、Microsoft にご協力ください。

手順 4. Windows 10 に登録する

「[Ready for Windows 10](#)」(Windows 10 への準備) Web サイトは、Windows 10 をサポートするソフトウェアのディレクトリです。社内に Windows 10 を展開しようとしている企業の IT 管理者や世界中の組織を対象としています。IT 管理者はこのサイトで、現在社内に展開しているソフトウェアが Windows 10 でサポートされているかどうかを確認することができます。

まとめ

Microsoft は、OS におけるアプリの互換性問題を評価する手段としてテレメトリに組織的に投資しています。主要な信頼性と使われ方のテレメトリを活用して変更の影響を理解することは、Windows 10 のテスト ビルドに対する互換性問題をすみやかに特定し、WaaS サイクルの早期に問題を解決できるようになります。アプリのリリースとサポートを特定の Windows ビルドから切り離し、フライト ビルドでアプリをテストすることを推奨します。アプリをこれらのビルドでテストして、アプリのテレメトリを取得してください。テスト中にアプリの問題を発見したら、Microsoft に問題を報告してください。ユーザーに上質なエクスペリエンスを構築するため、ぜひこれらの情報を提供し、Microsoft にご協力ください。

互換性ガイドブックの形式で、OS のリリースごとに変更点やベスト プラクティスの情報提供を続けていきます。以前の OS のドキュメントについては、MSDN のドキュメント ([Windows 7](#)、[Windows 8](#) と [Windows 8.1](#)) を参照してください。