



Visual Basic 6.0 アプリケーションの ビジネス プランニング ガイド

ホワイト ペーパー

2008 年 4 月

最新情報については、www.microsoft.com/japan/vstudio を参照してください。



このドキュメントに記載されている情報は、このドキュメントの発行時点におけるマイクロソフトの見解を反映したものです。マイクロソフトは市場の変化に対応する必要があるため、このドキュメントの内容に関する責任をマイクロソフトは問われたいものとします。また、発行日以降に発表される情報の正確性を保証できません。

このホワイトペーパーに記載された内容は情報の提供のみを目的としており、明示、黙示または法律の規定にかかわらず、これらの情報についてマイクロソフトはいかなる責任も負わないものとします。

お客様ご自身の責任において、適用されるすべての著作権関連法規に従ったご使用をお願いします。このドキュメントのいかなる部分も、米国 **Microsoft Corporation** の書面による許諾を受けることなく、その目的を問わず、どのような形態であっても、複製または譲渡、あるいは検索システムに格納または公開することは禁じられています。ここでいう形態とは、複写や記録など、電子的な、または物理的なすべての手段を含みます。

マイクロソフトは、このドキュメントに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

© 2008 Microsoft Corporation. All rights reserved.

Microsoft、IntelliSense、Microsoft Dynamics、SharePoint、Visual Basic、Visual Studio、Windows、Windows Server、および Windows Vista は、マイクロソフトグループの商標です。

その他、記載されている会社名、製品名には、各社の商標のものもあります。

概要	1
背景	2
Visual Basic 6.0 アプリケーションのサポート戦略	8
サポート戦略を決定するための指針	15
プロジェクト チーム向けの リソース	20
Visual Basic の各バージョンのサポートの概要	25
まとめ.....	26

概要

このホワイト ペーパーは、ビジネス意思決定者、最高情報責任者、最高技術責任者、最高経営責任者、およびプロジェクトのスポンサーが、**Microsoft Visual Basic 6.0** プラットフォームを基盤とする特注システムの今後のサポートと方向性を計画する際のガイドとなることを目的としています。

Visual Basic 6.0 プラットフォームは **1998** 年にリリースされて以来、多数のカスタム アプリケーションの基盤として使用されてきました。ただし、**Visual Basic 6.0** の後続バージョンが **4** 回リリースされたことを受けて、ビジネス意思決定者とプロジェクト チームが個々のビジネス要件、機能、およびコスト要因を基にアプリケーションを適宜移行できるようにするための、**Visual Basic 6.0** アプリケーションの一部のサポート、移行、および置換オプションの提供を終了しました。

このホワイト ペーパーは、次の **3** 部構成になっています。

1. **Visual Basic** の背景と歴史。**Visual Basic .NET** をあまりご存じない読者のために、**Visual Basic 6.0** と **Visual Basic .NET** の機能レベルの比較情報を提示します。
2. どのような戦略をどのようなときに選択すべきかについての戦略とガイドライン。**Visual Basic 6.0** アプリケーションの将来のロードマップを計画するのに役立ちます。
3. **Visual Basic 6.0** アプリケーションを使用しているチーム向けのリソース。

背景

1991年の Visual Basic 1.0 のリリース以来、Visual Basic は、世界で最もよく利用されているプログラミング プラットフォーム言語に成長しました。ほぼ 20 年にわたり、機能拡張を続け、10 回もの製品リリースを行いました。全世界で 600 万名以上の開発者が Visual Basic を使用して、世界のビジネスを動かすカスタムアプリケーションを作成しています。

Visual Basic のバージョン 1.0 から 6.0 までは、他の開発プラットフォームよりもすばやく簡単に Microsoft Windows アプリケーションと COM インフラストラクチャを作成できるようにすることに重点を置いていました。2002 年にリリースされた Visual Basic 2002 は .NET プラットフォームを基盤とし、新世代の機能を実現できるようになりました。たとえば、Windows アプリケーションとサービス、モバイルデバイスアプリケーション、スケーラブルなインターネットおよびイントラネット アプリケーション、XML Web サービス対応アプリケーションなどです。また、アクセシビリティ、オブジェクト指向開発、相互運用性、接続性、標準への準拠、ドメイン モデリング、自動テスト機能を提供するなど、.NET プラットフォームを基盤とすることで、それまでの COM ベースの Visual Basic と比較して、エンタープライズソフトウェア開発ライフサイクルの要件により適切に対応できるようになりました。

マイクロソフトでは、これまでに .NET プラットフォームを基盤とする Visual Basic (通称、Visual Basic .NET) の 4 つのバージョンをリリースし、着実に Visual Basic の生産性機能の拡張を続けています。最新バージョンである Visual Basic 2008 は、2007 年 11 月にリリースされました。

すべてのプログラミング言語は、顧客の要件と業界の動きに従って、成長および変化します。Visual Basic 2002 から Visual Basic 2008 に至るまでは、それまでのすべてのバージョンの Visual Basic において行われた機能強化の合計よりも多くの機能強化が行われています。しかし、顧客の要求に応えるために必要だったこの大きな進化によって、Visual Basic 6.0 とのコードの互換性が失われることになりました。Visual Basic 6.0 で作成されたアプリケーションには、Visual Basic 2008 ではそのままの状態ですべて再コンパイルができないというデメリットがありますが、新しい Visual Basic 2008 の機能を使用すると、非常に魅力的な新しいチャンスを創出できるというメリットがあるため、多くの組織が Visual Basic 6.0 アプリケーションの廃止を進めており、単に次期バージョンのプラットフォームに移行する以上の計画を立てています。

アプリケーション、運用、プロジェクト チームを Visual Basic 6.0 からできるだけ柔軟に移行できるように、マイクロソフトは、Visual Basic 2002 のリリースから Visual Basic 2008 のリリースまでの間に、ソフトウェア ツール、トレーニング サービス、および技術サポートを順次提供してきました。Visual Basic コミュニティが Visual Basic 6.0 から移行することを支援するこの取り組みは、最大規模のサポートの提供につながり、迅速なアプリケーション開発の最高のツールとして Visual Basic の今後の成長に尽力するマイクロソフトの姿勢を浮き彫りにしています。

マイクロソフトでは Visual Basic 2008 のリリースを機に、このような移行オプションを利用して、Visual Basic 6.0 アプリケーションの廃止を計画されることをお勧めしています。Visual Basic 2008 が開発者コミュニティに提供する新しい機能とチャンスをぜひご活用ください。

非互換について

Visual Basic 2002 から Visual Basic 2008 までのバージョンでは、Visual Basic 6.0 との互換性が失われました。具体的な影響としては、Visual Basic の場合、次の 2 点です。

1. 非常に単純なプロジェクトを除いて、Visual Basic 6.0 で開発されたアプリケーションは、何かしらの手直しをしない限り、それ以降のバージョンの Visual Basic から再コンパイルできません。
2. .NET プラットフォームの新機能を使用すると、Visual Basic アプリケーションでは、従来にも増してビジネス要件に的確に対応できます。多くの企業では、より多くのビジネス価値を創出できるようアプリケーションの見直しと拡張を行う足掛かりとして、Visual Basic 6.0 から Visual Basic 2008 にアプリケーションを移行する機会を利用しています。

Visual Basic 2002 から Visual Basic 2008 までのバージョンで Visual Basic 6.0 との互換性が失われた理由

違和感を感じるかもしれませんが、互換性が失われた理由は、Visual Basic 6.0 ではどうしても提供できない機能に対するユーザーからの要望に対応したことにあります。ユーザーから寄せられたいくつかの重要な要件を叶えるには、互換性を維持しないことが、Visual Basic とそのコミュニティにとって最適な選択でした。

インターネットへの対応 - Visual Basic 6.0 は、物理的に同じ 1 つの場所で少数の同時ユーザーによって使用されるデスクトップ Windows アプリケーション用の製品です。世界的に企業のインターネット対応が急速に進み、数万人の同時ユーザーがアクセスできる、コスト効率がよくスケーラブルで信頼性の高い Web ベース アプリケーションがすぐに必要になったために、Visual Basic 6.0 のステータフルな COM ベースのアーキテクチャから、よりスケーラブルで、セキュリティが強化された、障害に強いプラットフォームへの移行が必要でした。

相互運用性に対する要望 - Visual Basic 6.0 では、他の言語との相互運用が考慮されていなかったため、インターネットを介したより大規模な企業間の接続を実現する業界標準に準拠するには、Visual Basic 6.0 専用のデータ形式を作り変える必要がありました。また、パートナーやベンダとのシステムの相互運用性が、組織の成長に非常に重要なものとして見なされるようになり、標準ベースの "疎結合" を実現するように、アプリケーションを設計する必要がありました。相互運用性の規約に基づき、オペレーティング システムやプログラミング言語の種類、場所に関係なく、他のシステムと相互運用できる機能が必要になりました。

アクセシビリティとセキュリティの必要性 - ビジネス ソフトウェア アプリケーションの戦略的な重要性が高まるにつれて、ソフトウェア業界も成熟し、アクセシビリティの規定、ローカライズ要件、信頼できるコンピューティングの達成に重点を置くようになりました。このような機能を実現するには、Visual Basic 6.0 に実装できないプラットフォーム アーキテクチャへの変更が必要でした。

機能的な要件 - 継承とオブジェクト指向プログラミングのサポート、構造化例外処理、高度なグラフィックのサポート、バージョン管理機能の強化、堅牢な配置メカニズム、DLL Hell の終焉など、多数の長年懸案事項となっていたユーザーからの要望に応えるには、Visual Basic 開発システムのアーキテクチャに抜本的な変更が必要でした。

.NET プラットフォームに合わせたアーキテクチャの変更により、Visual Basic は将来のビジネス アプリケーション開発に対応できる機能を備えています。Visual Basic のコードの読みやすさはそのままに、.NET プラットフォームの威力と機能を簡単に活用できます。言語が進化を遂げても、Visual Basic の将来のバージョンが、ビジネスの生産性と迅速なアプリケーション開発を優先するという Visual Basic の当初の目標を忠実に守っていきます。

Visual Basic 2002 以降の各バージョンで新機能が追加されていますが、Visual Basic の以前のバージョンを使用した開発で習得した知識やスキルを、再び適用し、将来のアプリケーション開発の基礎として使用できるようにしています。

マイクロソフトは、今後も引き続き Visual Basic の拡張と改良に取り組み、Visual Basic を使用して重要なアプリケーションを構築およびサポートする企業や開発者を支え、新しいチャンスを創出していきます。Visual Basic 2002 において Visual Basic 6.0 との互換性が失われたときにお知らせしたように、マイクロソフトは、将来の Visual Basic に向けてこれまで以上の取り組みを進めていきます。

Visual Basic 6.0 と Visual Basic 2002 から Visual Basic 2008 までのバージョンの Visual Basic の相違点

Visual Basic 6.0 からその後のバージョンでは何が変わったかを考えるには、Visual Basic 6.0 と Visual Basic 2008 (現行バージョン) との違いを理解することが重要です。

Visual Basic 6.0 と Visual Basic 2008 の違いは、アプリケーション開発の新機能と、非互換性をもたらした変更の 2 種類に分けられます。

新機能

webclass と呼ばれる Visual Basic 6.0 の限られた Web プログラミング ソリューションは、スケーラビリティ、セキュリティ、デザイナのサポートの面で制限がありました。機能性、パフォーマンス、スケーラビリティの高いアプリケーションの迅速なアプリケーション開発においては、Visual Basic 2008 と ASP.NET の方が Visual Basic 6.0 よりも格段に優れています。

Visual Basic 2008 では、Windows アプリケーション開発だけでなく、Windows サービス、モバイル デバイス、コンソール アプリケーション、32 ビットおよび 64 ビット アプリケーションの開発にも対応しています。

Visual Basic 2008 は、Visual Basic 6.0 の専用の API と比べて、C# や Visual Basic 6.0 などのマイクロソフト テクノロジーを使用して作成されたものだけでなく、Linux 上で Java を使用して作成されたものなどマイクロソフト以外のプラットフォーム ベースのアプリケーションやサービスとも相互運用できる機能を備えており、相互運用の面では格段に柔軟性とスケーラビリティが向上しています。Visual Basic 6.0 では Microsoft Office クライアント製品と緊密に統合されていましたが、Visual Basic 2008 では、この機能がさらに拡張され、Microsoft Office クライアント製品だけでなく、Microsoft Office サーバー製品、Microsoft SQL Server、Microsoft Office SharePoint 製品ファミリ、Microsoft CRM など、マイクロソフトが開発したほぼすべてのエンタープライズ製品との統合をサポートしています。

Visual Basic 6.0 のセキュリティ機能は、アプリケーション開発者自身が構築したものにしか適用できませんでしたが、Visual Basic 2008 では、信頼レベル、ロール ベースのセキュリティ、暗号化、複数の認証メカニズムをサポートする、すぐに使用できる堅牢なセキュリティ機能が導入されています。

Visual Basic 6.0 でのオブジェクト指向開発のサポートは限られており、アプリケーション開発を支援する、再利用可能なライブラリ コンポーネントは基本的なものしか提供されていませんでした。Visual Basic 2008 では、包括的なオブジェクト指向機能、標準のサポート、完全な .NET Framework ライブラリ コンポーネントへのアクセスが実装されていて、エンタープライズ レベルの最新のアプリケーションをすばやく、安全に、より堅牢に構築できます。

非互換性をもたらした変更

Visual Basic 6.0 では、豊富な機能を備えた Windows フォーム開発環境を提供していました。Visual Basic 2008 でも、Visual Basic 6.0 の流れを引き継いで豊富な機能を備えた Windows フォーム開発環境を提供していますが、完全な下位互換性は確保されていません。特にデータ バインド モデルが改良され、サーバー上およびネットワークで接続されていない遠隔地にあるデータへのアクセスが向上した、ADO.NET ベースの柔軟で堅牢なバインド モデルが作成されています。Visual Basic 6.0 の RDO と DAO のデータ バインドは、1 人のユーザーが 1 台のコンピュータ上にあるデータベースにアクセスすることを想定して最適化されていましたが、ADODB は、ローカル エリア ネットワーク内で密に結合されているデータベース サーバーにアクセスするためのものとして設計されていました。Visual Basic 2008 と ADO.NET は、インターネットまたはワイドエリア ネットワーク上での複数ユーザーによるクライアント/サーバー アクセス用に最適化されています。

Visual Basic 6.0 では、専用の変数と型のモデルが使用されていましたが、Visual Basic 2008 では、型と変数の変換がより厳格になり、実行時の問題を防ぎ、他の言語との相互運用性が向上しています。

Visual Basic 2008 アプリケーションは、他の .NET ベースのアプリケーションと同様に、マネージ アプリケーションです。つまり、.NET 共通言語ランタイムによって監視され、ガベージコレクション、バージョン管理、セキュリティ インフラストラクチャ、およびアンマネージの Visual Basic 6.0 Windows アプリケーションと比べて優れたエラー防止機能を備えているアプリケーションです。Visual Basic 2008 アプリケーションと Visual Basic 6.0 の動作にはわずかに違いがあり、パフォーマンス プロファイル、実行時のイベント処理モデル、およびオブジェクトが作成および破棄されるライフサイクルが異なります。

変更されていない点

どのバージョンの Visual Basic も、生産性の向上を目標としています。ほとんどの Visual Basic 6.0 開発者は、Visual Basic 2008 にすぐに適応し、より大きな成果を上げています。Visual Basic 2008 には、エディット コンティニュー、IntelliSense、大文字小文字の区別がされないことなど、Visual Basic 6.0 と同じ迅速なアプリケーション開発をサポートする機能の多くを備えています。さらに、Visual Basic 2008 では、入力時に行われるバックグラウンドでのコンパイルやエラー報告、コード テンプレート、コンピュータ リソースへのアクセスを容易にする My* クラスなど、新しい生産性機能が導入されています。

プログラミングとビジネス ロジックは、非常に似ています。Visual Basic 6.0 のビジネス ロジックのほとんどは、わずかな変更を加えるだけで、Visual Basic 2008 でも使用できます。Visual Basic 2008 には、コードの移行を支援するアップグレード ツールや、.NET プラットフォームのライブラリ クラスに対して、使い慣れた VB6 インターフェイスを使用できるようにする互換性クラスが用意されています。

Visual Basic 2008 は将来の開発に適しているか

マイクロソフトが .NET プラットフォームをリリースする前は、アプリケーションの種類ごとに異なるプログラミング パラダイムが使用されていました。たとえば、Web アプリケーションには VBScript、Windows アプリケーションには Visual J++、Visual C++、および Visual Basic、SQL データベース プログラミングには Transact SQL、Microsoft Office のカスタマイズには VBA が使用されてきました。また、製品ごとに、専用のカスタマイズ ツールキットが同梱されていました。各ツールキットには、専用のライブラリ コンポーネント、変数型、および開発環境がありました。プロジェクト チームは専門知識を蓄積し、別の種類のソリューションでは容易に再利用できない、特定のソリューションにのみ取り組んでいました。.NET プラットフォームと Visual Studio .NET 開発環境が導入され、急速に普及したことで、マイクロソフトでは、.NET プラットフォームを基盤に構築を計画される予定のすべての製品のカスタマイズ モデルを標準化しました。その結果、プロジェクト チームは、別の種類のソリューションを作成する場合に、よりスムーズにスキルを応用でき、一貫性を確保できるようになりました。

2 つの非常に優れている .NET 言語は Visual Basic と Visual C# で、どちらも Microsoft SQL Server、Windows SharePoint Server、Microsoft Office Server System、Microsoft Office クライアント、Microsoft CRM などのスタンドアロン アプリケーションやソリューションの作成に使用できます。.NET プラットフォームは、アプリケーション開発とエンタープライズ規模でのアプリケーション統合の標準として認められています。

Visual C# と Visual Basic は、どちらもエンタープライズソリューションを作成する言語としては自然な選択です。どちらの言語も .NET プラットフォームを基盤としていて、同じ変数と型をサポートし、.NET Framework のリッチなプログラミング コンポーネントを使用し、同様のパフォーマンス、スケーラビリティ、相互運用機能を備えています。Visual C# と Visual Basic のどちらを選ぶかは、好みの問題です。Visual Basic または VBA での開発経験がある場合は Visual Basic を、C++ または Java での開発経験がある場合は Visual C# を使用すると、より生産的に作業できるでしょう。Visual Basic 2008 には迅速なアプリケーション開発をサポートする機能が、Visual C# 2008 にはコードの詳細な調整をサポートする機能がより多く実装されています。どちらの言語も、Microsoft Team System のテスト、ソース管理、ビジュアル デザイン ツールを完全にサポートしています。また、マイクロソフトは、Visual Studio と .NET プラットフォームの将来のバージョンにおいて、この 2 つの言語を強化していく予定です。

ほとんどの Visual Basic 開発者にとって、今後の開発に Visual Basic 2008 を選択することは自然な流れです。Visual Basic 2008 でも、使い慣れた言語、開発環境機能、ビジネス アプリケーション作成言語として Visual Basic の以前のバージョンが驚くほどの人気を博す要因となった、学習しやすい生産性機能をサポートしています。

Visual Basic 1.0 ~ 6.0 は、簡単に習得できて、取り掛かりやすいツールであると見なされてきましたが、高度な機能が提供されていません。これは、"VB のガラスの天井" と呼ばれることがありました。Visual Basic 2008 にはそのような制限はなく、Monsanto やワシントン州の運転免許交付局など、大規模な組織で、Visual Basic と .NET プラットフォームを使用したカスタム ビジネス アプリケーションの構築が行われています。Visual Basic 2008 は、.NET プラットフォームを基盤とするエンタープライズ レベルのアーキテクチャによって、この Visual

Basic 6.0 のガラスの天井を打破しています。このアーキテクチャは、世界の最大規模のビジネスの要件にも対応できるように、ゼロから設計されています。

サードパーティによる .NET プラットフォームも非常に充実しています。アプリケーション開発支援専用のコンポーネントを販売しているコンポーネント ベンダの巨大なコミュニティがあります。このようなコンポーネントの大半では、Visual Basic と Visual C# の両方がサポートされています。開発者やプロジェクトチームによる .NET プラットフォームの採用も進んでいて、他のどのプログラミング プラットフォームよりも、生産性が高く強力であることが認められています。新しい Visual Basic 2008 ソフトウェアの作成や、既存の Visual Basic 6.0 アプリケーションの Visual Basic 2008 への移行を専門とするベンダもいます。

これまで Visual Basic 6.0 アプリケーションに投資してきた組織にとっては、Visual Basic 2008 はさらに魅力ある選択肢です。マイクロソフトでは、Visual Basic 2008 への移行、および Visual Basic 6.0 と Visual Basic 2008 の相互運用を支援する、トレーニング、ツール、相互運用サポートを提供しています。

Visual Basic 2008 が将来を見据えた開発に適していることは、間違いありません。Visual Basic 2008 は、Visual Basic プログラマとエンタープライズクラス開発の支援を目的としています。

今後リリースされる Visual Basic では、引き続き生産性の向上と迅速なアプリケーション開発を支援する機能の追加を行い、Visual Basic が .NET プラットフォームを完全にサポートし、その威力を活用できるようにする予定です。Visual Basic 6.0 との互換性を維持しなかったことが、将来の開発に対応できる Visual Basic の誕生に貢献しています。.NET プラットフォームのアーキテクチャにより、再び互換性維持の問題に直面することなく、64 ビットプロセッサのサポートなどの機能を Visual Basic 2008 に容易に追加できます。マイクロソフトは、今後のバージョンでも Visual Basic を強化し、可能な限り少ない時間で学習でき、価値を実現できる、強力なプログラミング環境をお届けできるよう取り組んでいきます。

VISUAL BASIC 6.0 アプリケーションの サポート戦略

このセクションでは、Visual Basic 6.0 アプリケーションへの既存の投資に対して利用できる戦略について説明します。また、採用すべき戦略を決定するための指針も提供します。

評価

Visual Basic 6.0 アプリケーションをサポートする戦略を検討する前に、まず既存のアプリケーションの評価を行う必要があります。

- 組織内には、いくつの Visual Basic 6.0 アプリケーションが配置されていますか。
- 各アプリケーションは、どのような状態でしょうか。運用されていますか、運用停止されていますか。
- 各アプリケーションは、どの程度重要 (ミッションクリティカル) ですか。
- 各アプリケーションは、現在もビジネス要件を満たしていますか。
- 各アプリケーションをサポート、変更、および機能強化できる人員は配置されていますか。各アプリケーションの技術ドキュメントは、最新の状態ですか。

評価が済んだら、各アプリケーションを検討する順序を決定し、以下のガイドに従って、アプリケーションごとに戦略を策定します。

戦略

Visual Basic 6.0 アプリケーションに対する戦略には、以下の 4 種類があります。

- 維持
- 相互運用
- 移行
- 置換

維持

意味

アプリケーションを Visual Basic 6.0 ベースのままとし、サポートを継続します。

この戦略が適している状況

アプリケーションが成熟していて、頻繁に更新する必要がない場合。アプリケーションを更新するということは、別の戦略がより適している可能性があるときに、適切なリソースを探し、アプリケーションへの投資を続行することになります。

これはコスト効率の高い戦略ですが、アプリケーションを Visual Basic 6.0 ベースのまま維持することは、長期的な実用性がない方法であると見なす必要があります。ビジネスが変化し、既存のアプリケーションに対して新しい要件が発生した場合に、アクセシビリティや信用できるセキュリティなど、.NET プラットフォームでは標準で対応できる新しいビジネス要件の多くに、Visual Basic 6.0 では対応できない可能性があります。

アプリケーションを Visual Basic 6.0 ベースで維持する場合の対応として重要なことは、アプリケーションが運用できる状態を維持することです。これには、以下の 3 つの重要な要素があります。

開発の専門家 - アプリケーションをサポートし、重要な修正プログラムを適用し、アプリケーションを継続して変更できるプロジェクトチームを確保する必要があります。

ランタイム環境 - Windows Vista や今後リリースされるオペレーティングシステムをデスクトップに展開する際に、Visual Basic 6.0 アプリケーションが新しいオペレーティングシステムでも機能することを確認する必要があります。マイクロソフトは、2017 年まで Visual Basic 6.0 アプリケーションのランタイムサポートを提供することを発表しています。つまり、Windows Vista と Windows 2008 も含め、オペレーティングシステムでは Visual Basic 6.0 アプリケーションの実行が引き続きサポートされます。これは、Visual Basic ランタイムとコンポーネントにも適用されます。ただし、サードパーティ製のコンポーネントを使用しているアプリケーションについては、製造元に当該コンポーネントが引き続きサポートされることを確認する必要があります。

開発環境 - Visual Basic 6.0 アプリケーションに機能強化を施す場合、Visual Basic 6.0 統合開発環境 (IDE) を使用してアプリケーションを変更し、再コンパイルする必要があります。ランタイム環境と同様に、Visual Basic 6.0 IDE をサポートするワークステーションを引き続き提供する必要があります。マイクロソフトは、Windows Vista での Visual Basic 6.0 IDE のサポートを発表しており、2008 年 4 月までサポートを延長しています。つまり、開発チームは、引き続き

Windows Vista または Windows XP を使用して、Visual Basic 6.0 アプリケーションを変更できるということです。ランタイム サポートと同様に、サードパーティ製のコンポーネントもサポートされることを、当該コンポーネントの製造元に確認してください。

Visual Basic 6.0 ベースでアプリケーションを維持することは、多くの場合、暫定的な措置と考えられ、予想される残りの運用期間が限られている、成熟したアプリケーションに適しています。最終的には、Visual Basic 6.0 アプリケーションの運用を停止して最新バージョンの Visual Basic に移行するか、仮想化テクノロジーを使用して、アプリケーションが依存しているオペレーティングシステムや他のソフトウェアのサポート ライフサイクルの終了後も、アプリケーションを実行できるようにする必要があります。短期間であれば Visual Basic 6.0 ベースでアプリケーションを維持することで、組織は、アプリケーションの移行に必要な投資と、利用できる資金やリソースのバランスを取ることができます。

相互運用

意味 相互運用では、Visual Basic 2008 を使用して Visual Basic 6.0 アプリケーションを強化するか、徐々に Visual Basic 6.0 コンポーネントを Visual Basic 2008 コンポーネントに置き換えます。この戦略では、長い時間をかけて徐々に書き直しを行うことになり、その間 Visual Basic 6.0 と Visual Basic 2008 のハイブリッド アプリケーションをサポートすることになります。

この戦略が適している状況 定期的に変更される Visual Basic 6.0 アプリケーションがある (つまり、完全なアップグレードや書き直しに必要な時間、アプリケーションを停止することができない場合)。

Visual Basic 2008 では、相互運用を非常に魅力的な選択肢とするさまざまな機能を提供しています。たとえば、Visual Basic 6.0 ビジネス オブジェクトを Visual Basic 2008 から参照する機能、Visual Basic 2008 から Visual Basic 6.0 のデバッグを実行する機能、Visual Basic 6.0 アプリケーションから Visual Basic 2008 のフォームとコントロールを使用できるようにする Interop Forms Toolkitなどを提供しています。

この戦略では、Visual Basic 2008 機能を使用して、既存のアプリケーションに価値をすぐに追加できるため、コスト効率の良い戦略になり得ます。この戦略は、既存の Visual Basic 6.0 アプリケーションが、コンポーネント化されたアーキテクチャを持ち、適切に構築および記述されている場合に最適です。このような場合には、新機能の追加と古いコンポーネントの置換を簡単に行えます。

Visual Basic 6.0 ベースでアプリケーションを維持する戦略と同様に、相互運用も暫定的な措置と見なす必要があります。アプリケーションのコンポーネントであっても、Visual Basic 6.0 ベースで維持した場合、Visual Basic 6.0 ベースでアプリケーション全体を維持したときと同じサポートの問題が発生します。相互運用は、既存のアプリケーションの運用期間を一時的に延長するか、アプリケーションのコンポーネントを徐々に Visual Basic 2008 に移行する場合に、最も多く使用される戦略です。

移行

意味

Visual Studio 2008 の自動アップグレード ツールを使用して、Visual Basic 6.0 アプリケーションを Visual Basic 2008 に移行します。

この戦略が適している状況

アプリケーションでコンポーネント化されたアーキテクチャが採用されていて、徐々に移行できる場合や、開発チームがアプリケーションを頻繁に変更する必要がないほど、アプリケーションやビジネスが安定している場合。

マイクロソフトでは、Visual Basic 6.0 アプリケーションの Visual Basic 2008 への自動的な移行を支援する多数のユーティリティをリリースしています。通常、アップグレードは、次の 3 段階のプロセスで行われます。

1. Visual Basic 6.0 ベースのアプリケーションの準備をします。
2. 自動ツールを使用してアップグレードします。
3. 多少手直しをして、アプリケーションの移行を完了します。

移行後のアプリケーションの品質は、移行前の Visual Basic 6.0 アプリケーションの品質と構造に左右されます。通常、ビジネス ロジックのある中間層コンポーネントの方が、ユーザー インターフェイスの調整や COM の特性に依存するフォームベースのアプリケーションよりも、簡単に移行できます。RDO および DAO のデータ バインドやゼロベースではない配列ロジックなど、Visual Basic 6.0 の一部の要素は、自動ツールを使用して移行できません。Visual Basic 2008 アプリケーションでは、Visual Basic 6.0 とは異なる種類のデータ バインドと変数型の標準を使用しているため、通常、このようなコードは手直しが必要です。

徐々にアプリケーションを移行する (ビジネス ロジックから移行に着手し、最後にユーザー インターフェイス層を移行する) 場合も、自動アップグレード ツールを使用できます。コンポーネント化されたアーキテクチャが採用されているアプリケーションでは、この方法が便利なが多くあります。

適切にメンテナンスされていないアプリケーションや、アーキテクチャが古いアプリケーションでは、移行後のアプリケーションでも同じ制限が継承されることに注意してください。

また、移行後のアプリケーションが、COM コンポーネントへの依存性を継承している場合があり、この依存性は、アプリケーションのメンテナンス時に、どこかの時点で置き換える必要があります。

移行前に、アプリケーションの何パーセントが自動的にアップグレードでき、何パーセントに手直しが必要になるか

を予測することは困難です。多くの組織では、より正確な予測を立てられるように試験的な移行を行う際にマイクロソフトが提供している自動評価ツールを使用して、適合性調査を実施しています。また、アプリケーションの一部のみを移行して、徐々にアプリケーションを移行できるようにし、**Visual Basic 6.0** コンポーネントと移行後の **Visual Basic 2008** コンポーネント間の相互運用性を利用してハイブリッド アプリケーションをユーザーに提供する場合があります。

マイクロソフトが提供している移行ツールだけでなく、外部のコンサルタントに依頼して、移行を実施する場合があります。**Visual Basic 6.0** アプリケーションの移行経験があるコンサルタントであれば、移行プロセスに精通しているため、多くの場合、はるかに短時間で移行を完了できます。

また、移行を自動化する製品を販売している小規模な独立系ソフトウェア ベンダも多数あります。通常、このようなベンダは、**Visual Basic 6.0** 構文のみを移行し、相互運用ライブラリを利用して **.NET Framework** を古い **Visual Basic 6.0** インターフェイスの構造に適応させています。これは、非常に魅力的なオプションであることもありますが、このような相互運用ライブラリと将来のバージョンの **.NET Framework** との互換性が確保されない可能性も考慮する必要があります。

置換

意味

Visual Basic 6.0 アプリケーションを、市販のソリューションと置き換えるか、Visual Basic 2008 で作成し直して、置き換えます。

この戦略が適している状況

Visual Basic 6.0 アプリケーションが、ビジネス要件に合わなくなった場合。

ビジネス要件を満たせなくなったり、かつては特殊だった必要な機能が、現在は独立系アプリケーション ベンダによって提供されているなど、Visual Basic 6.0 アプリケーションのライフ サイクルが終わりに近付いている場合は、アプリケーションの再構築や移行を行うのではなく、運用を停止して置き換えることができます。

置換という戦略を採用する場合は、代わりとなる市販の構築済みアプリケーションを購入するか、新しいカスタムアプリケーションを開発します。

通常、適切な市販の製品がある場合は、市販の製品に置き換える方法が非常にコスト効率が高くなります。一方、新しいアプリケーションを開発する方法は、最もコストがかかりますが、ビジネスの要件を再定義でき、最適な移行を実現できるアプリケーションを配置できます。

構築するか購入するかの決定は、常に複雑です。ただし、多くの Visual Basic 6.0 アプリケーションは、財務、会計、人事などの事務管理のニーズを満たすために構築されたものです。これは、1990 年から 2000 年代初めにかけてのことで、当時は、非常に大規模な企業を対象としたものを除き、ソフトウェア業界によってこのようなニーズに適切に対応できるテクノロジーが提供されておらず、Visual Basic 1.0 ~ 6.0 が広く使用されていました。現在、多くのエンタープライズアプリケーション ベンダが、カスタム アプリケーションの機能を置き換える以上の機能性を備えた、柔軟で、カスタマイズ可能で、強力なアプリケーションを提供しています。Microsoft Dynamics 製品ラインを含め、このようなベンダの多くは、顧客に代わりアプリケーションをホストして管理する、サブスクリプションによるオンデマンドのサービスも提供しています。

サポート戦略を決定するための指針

Visual Basic 6.0 アプリケーションのサポート戦略を計画する場合は、一般的な 6 つの項目について検討する必要があります。Visual Basic 6.0 アプリケーションを各項目に照らし合わせて確認することで、適切な戦略を決定するための一助としたり、プロジェクト チーム内の理解を深める話し合いの材料とすることができます。

ユーザー要件	
<p>既存のソリューションは、ユーザー要件を現在も満たしていますか。</p> <ul style="list-style-type: none"> 明示されている機能要件 明示されていない機能要件 (セキュリティ、相互運用性、展開、アクセシビリティ、使用可能性、信頼性、パフォーマンス) 既存のソリューションは、ビジネスの成長要件を満たすことができますか。 スケーラビリティ 地理的な場所 <p>現在も、大半の要件を満たしているソリューションは、Visual Basic 6.0 ベースで維持、移行、または相互運用による運用延長が適切な戦略の候補として考えられます。要件を満たしていないソリューションには、置換の戦略が適しています。</p>	
維持	現在もすべてのユーザー要件を満たしている
相互運用	ほぼすべてのユーザー要件を満たしている
移行	一部のユーザー要件を満たしている
置換	ユーザー要件を満たしていない

サポート可能性	
<p>既存のソリューションは容易にサポートできますか。</p> <ul style="list-style-type: none"> ソリューション実装の質 アプリケーションをサポートするのに必要な社内およびベンダによる運用チームのスキル サードパーティ製のコントロールとコンポーネントの製造元によるサポート状況 開発標準への準拠 ソリューションをサポートおよびメンテナンスする運用スタッフの能力 <p>容易にサポートできるソリューションには、Visual Basic 6.0 ベースで維持する戦略が適している可能性があります。サポート上の制約があるソリューションには、置換の戦略が適しています。</p>	
維持	Visual Basic 6.0 の十分なサポートを確保できる
相互運用	Visual Basic 6.0 と新しいバージョンの十分なサポートを確保できる
移行	Visual Basic 6.0 と新しいバージョンの十分なサポートを確保できる
置換	Visual Basic の新しいバージョンの十分なサポートを確保できる

コスト

戦略には、どの程度のコストをかけられますか。

- 運用コストと調達コスト
- 年間のライセンスとサポート コスト

短期的には、Visual Basic 6.0 ベースでの維持または相互運用が、最もコスト効率の良い戦略です。長期的には、移行または置換が、プラットフォームの標準化、サポート コストの削減、ライセンスの統合によるメリットが得られるので、最もコスト効率の良い戦略です。

維持	短期的にはコスト効率が高い
相互運用	短期的にはコスト効率が高い
移行	長期的にはコスト効率が高い
置換	長期的にはコスト効率が高い

リリースまでの時間

アプリケーションは、いつまでに必要ですか。

- 開発と運用までのスピード
- 今後の変更を行うまでのスピード
- プラットフォームの変更に対応できるアジリティ (敏捷性)

リリースまでにかかる時間は、プロジェクトが完了し、サポート戦略を展開するまでに必要な時間として考えます。この判断の基準は、コスト、競合プロジェクト、アプリケーションの重要性、現在の Visual Basic 6.0 アプリケーションに対するサポートが得られるかどうか、およびサポートの質によって影響を受ける可能性があります。

維持	リリースまでの時間が短い
相互運用	リリースまでの時間が短い
移行	リリースまでの時間が長い
置換	リリースまでの時間が長い

ビジネスの成長

Visual Basic 6.0 アプリケーションは、ビジネスの今後の成長をどの程度サポートできますか。

- 予想される運用期間
- 業者、パートナー、顧客との相互運用性と接続性
- 他のビジネス ソフトウェアとの統合機能 (Office SharePoint、Office、マイクロソフト以外のアプリケーション、パートナーおよびベンダのシステム)
- スケーラビリティとパフォーマンス要件

このビジネスの成長という項目は、最も重要な検討事項の 1 つです。アプリケーションを .NET プラットフォームに移行することで、ビジネスの今後の成長に対応できる可能性が高くなるのは明らかですが、同時に、すべてのアプリケーションが永遠に拡張されるわけではなく、予想される運用期間が限られていて、拡張をサポートする必要がほとんどない、短期的なアプリケーションも多数あります。

維持	ビジネスの成長に対応する必要性が低い
相互運用	ビジネスの成長に対応する必要性は中程度
移行	ビジネスの成長に対応する必要性は中程度
置換	ビジネスの成長に対応する必要性が高い

業界要件

既存のアプリケーションで対応する必要がある外部の業界要件には、どのようなものがありますか。

- 政府の法令遵守規定
- パートナーおよびベンダとのインターフェイス
- 標準のサポート
- 競合他社と同等または独自の価値提供

直近のビジネス要件以外に、アプリケーションでは、どのような法令や規制要件を満たす必要がありますか。アクセシビリティやレポートについて法的要件を設けている政府もあり、このような要件に対応するには、Visual Basic 6.0 よりも Visual Studio 2008 の方が適しています。また、標準のサポートやアプリケーションの相互運用性サポートも、Visual Studio 2008 の方が優れています。

維持	業界要件とは無関係である
相互運用	ある程度業界要件に関係がある
移行	ある程度業界要件に関係がある
置換	業界要件にかなり関係がある

戦略決定の指針を使用したアプリケーションの評価

戦略決定の指針を使用した Visual Basic 6.0 アプリケーションのサポート戦略を評価する方法はいくつかあります。簡単な方法としては、項目ごとに戦略の順位 (1 ~ 4) を付けます。たとえば、以下の表では、"ユーザー要件" 項目の場合、最も好ましい戦略は置換で、相互運用、移行と続き、最後が維持の順となって

います。各戦略の順位を付けたら各列の合計を算出します。6項目を総合した場合に最も合計値の低いものが、最も好ましい戦略であると評価します。

評価例	維持	移行	相互運用	置換
ユーザー要件	4	3	2	1
サポート可能性	1	3	2	4
コスト	4	2	1	3
リリースまでの時間	3	2	1	4
ビジネスの成長	4	3	2	1
業界要件	1	3	2	4
合計	17	16	10	17

上記の例では、相互運用の合計値が最も低く、戦略を決定する指針の6項目を総合して最適な戦略であると評価できます。

コストと投資収益率の評価

各戦略のコストは、アプリケーションの種類、プロジェクトチームの能力、アプリケーションの将来の目標によって異なります。明確なルールはありませんが、各戦略を比較検討する際の一般的なガイドラインをいくつか紹介します。

- Visual Basic 6.0 アプリケーションを変更せず、Visual Basic 6.0 ベースで維持する方法が、最もコストのかからないオプションです。コストは、運用サポートに限定されます。
- アプリケーションを記述し直して置き換える方法が、最もコストのかかるオプションです。この場合のコスト見積もりの精度は、他のソフトウェア開発プロジェクトと同程度になります。
- 移行または相互運用の戦略を採用した場合のコストは、アプリケーション、プロジェクトチーム、およびプロジェクトの目標によって異なります。多くの組織では、最初に、移行や相互運用プロジェクトに伴うコスト、課題、および機会をより正確に把握できるように、任意のアプリケーションコンポーネントを移行するか、相互運用機能を一部のアプリケーション機能に追加して、期限を定めた適合性調査を実施しています。移行と相互運用の戦略では、どちらでも、このような段階的な方法が可能です。

同じく、各戦略の投資収益率 (ROI) を比較検討する際の一般的なガイドラインを紹介します。

- Visual Basic 6.0 ベースでアプリケーションを維持する場合、新機能を追加したり、新しいビジネス要件を満たしたりしなければ、新たな ROI は発生しない可能性があります。Visual Basic 6.0 のサポート期間の制限により、投資によるメリットは短期間しか得られません。
- 相互運用の ROI は、コスト効率よく新機能を追加することで実現されます。Visual Basic 6.0 ベースでアプリケーションを維持する場合と同様に、Visual Basic 6.0 のサポート期間の制限により、投資によるメリットは短期間しか得られません。
- 移行プロジェクトに COM コンポーネントの削除と、.NET プラットフォームの機能を利用できるようにアプリケーションのアーキテクチャを変更する作業が含まれる場合、ROI には Visual Basic 6.0 開発サポートの

廃止、プラットフォームの統合による運用コストの削減、新機能によるメリットも含まれます。

- アプリケーションを置換する場合、**Visual Basic 6.0** アプリケーションのアーキテクチャ上の制約を継承する不安を抱えることなく、ビジネス要件に合わせてゼロからアプリケーションを構築できるため、移行よりも高い投資メリットが得られる可能性があります。通常、この投資メリットは、新機能によるメリットと、**Visual Basic 6.0** 開発と運用サポートの廃止により実現されます。

プロジェクト チーム向けの リソース

このセクションでは、プロジェクト チームの観点から 4 種類の戦略の重要な点、つまり、各戦略を実装するためにチームが実行すべきことについて説明します。また、各戦略に関してマイクロソフトが提供しているリソースの情報も提供します。

Visual Basic 6.0 ベースでの維持

Visual Basic 6.0 ベースでアプリケーションを維持する方法は、プロジェクト チームにとって簡単なオプションです。おそらく、新しい開発手法を習得しなくても、アプリケーションを運用し続けることができます。しかし、この方法を利用する場合も調査は必要です。サードパーティ製のコンポーネントは、サポートされ移行できるか、プロジェクト チームと運用チームは、Visual Basic 6.0 アプリケーションが機能する開発環境とランタイム環境を維持できるかなどを確認する必要があります。

Windows Vista と Windows Server 2008 には、ランタイム サポートが実装されませんが、Windows Vista と Windows Server 2008 以降のオペレーティング システムではランタイム サポートの提供は計画されていません。Windows 64 ビット アプリケーション (Windows Vista および Windows Server 2008) も、ランタイム プラットフォームとしてサポートされます。Visual Basic 6.0 アプリケーションは、WOW 層で実行されることに注意してください。64 ビット コンポーネントと 32 ビット コンポーネントを同じプロセス内に混在させて調整することはできません。

サードパーティ製のコントロールとコンポーネントのサポート状況は、製造元ごとに異なります。一般に、Windows Vista よりも Windows XP の方が、多くのサードパーティのコントロールがサポートされています。ただし、Windows Vista は Windows XP よりもかなり多くのセキュリティ機能を備えていて、Windows Vista オペレーティング システムでは、システム リソースへのアクセスや変更に関するセキュリティが強化されているため、一部の古い COM コンポーネントは、適切に動作しない可能性があります。WebBrowser コントロール (TriEdit.ocx) は、Windows Vista と Windows 2008 のどちらでもサポートされていません。

登録を必要としない COM

Visual Basic 6.0 アプリケーションの技術的な問題の主な原因の 1 つは、配置時の DLL の登録です。配置を簡略化するため、Registration-Free COM (登録を必要としない COM) という新機能を Windows 2003、Windows XP、および Windows Vista に組み込んでいます。名前が示すように、登録を必要としない COM は、レジストリ エントリを作成せずに、コンポーネントをインストールして使用する手法です。レジストリ エントリは、アプリケーションと共に配布されるマニフェスト ファイル内に含まれています。DLL アドインのテクノロジーに関していくつ制限がありますが、多くの Visual Basic 6.0 実行可能ファイルでは、これにより配置プロセスが大幅に簡略化されます。これは、Visual Basic 6.0 アプリケーションに施すことができる簡単な機能強化です。

仮想化とターミナル サービス

Windows Server 2008 では、既存の優れたターミナル サービス機能を基に、新しいターミナル サービス機能が取り入れられています。この機能により、Visual Basic 6.0 アプリケーションでは、ターミナル サービス RemoteApp を使用して Windows Server 上でアプリケーションをホストして、クライアント コンピュー

タからそのアプリケーションにアクセスできます。アプリケーションはサーバー上で実行されますが、クライアント上で実行されているかのように動作し、ローカル コンピュータと通信します。また、複数のクライアントが、サーバー上の同じアプリケーションにアクセスできます。この手法は、配置の問題を最小限に抑え、Visual Basic 6.0 クライアント アプリケーションをすばやく配置するうえで有効です。

関連リソース

一般的なリソース	
Windows Vista と Windows Server 2008 における Visual Basic 6.0 のサポートについて	http://www.microsoft.com/japan/msdn/vbasic/support/vistasupport.aspx
Visual Basic 6.0 リソース センター (英語)	http://msdn2.microsoft.com/en-us/vbrun/default.aspx
Visual Basic 6.0 関連のダウンロード	http://www.microsoft.com/japan/msdn/vstudio/downloads/
Visual Basic 6.0 開発者向けの基本的なトレーニング (英語)	http://msdn2.microsoft.com/en-us/vbrun/cc297223.aspx

登録を必要としない COM による Visual Basic 6.0 アプリケーションの配置の簡略化	
ClickOnce と 登録を必要としない COM によるアプリケーション配置の簡略化 (英語)	http://msdn.microsoft.com/en-us/magazine/cc188708.aspx
登録を必要としない COM コンポーネントのアクティベーション: ウォークスルー (英語)	http://msdn2.microsoft.com/en-us/library/ms973913.aspx

仮想化とターミナルサービス	
仮想化を使用した Visual Basic 6.0 クライアント/サーバー アプリケーションの維持 (英語)	http://msdn2.microsoft.com/en-us/library/cc522935.aspx
マイクロソフトの仮想化による Visual Basic 6.0 アプリケーションの可用性 (英語)	http://msdn2.microsoft.com/en-us/library/cc526339.aspx
Windows Server 2008 のターミナルサービス	http://www.microsoft.com/japan/windowsserver2008/terminal-services/default.mspx
ターミナル サービス RemoteApp (英語)	http://technet2.microsoft.com/windowsserver2008/en/library/57995ee7-e204-45a4-bcee-5d1f4a51a09f1033.mspx?mfr=true

仮想化とターミナル サービス

Windows Server 2008 RC0 ターミナル サービス RemoteApp ステップ バイ ステップ ガイド	http://download.microsoft.com/download/0/C/1/0C1EBDD3-C4DA-48C5-ABFB-0940DB1B2109/Windows_Server_2008_Terminal_Services_RemoteApp_Step-by-Step_Guide.doc
---	---

相互運用

相互運用は、Visual Basic 6.0 アプリケーションの運用期間を延長する方法としては、非常にコスト効率の良い方法です。相互運用に関しては、Visual Basic 2002 の初期リリースから Visual Basic 2008 のリリースまでの間に、大幅な機能強化が行われました。現在では、Interop Forms Toolkit を使用して、Visual Basic 2008 の Windows フォームとコントロールを Visual Basic 6.0 アプリケーションに組み込んで、ハイブリッドアプリケーションを作成できます。

また、Visual Basic 2008 DLL のメソッドの Visual Basic 6.0 アプリケーションからの呼び出しや、その逆の呼び出しも容易に行うことができます。Visual Basic 2002 以降の Visual Basic のバージョンごとに、着実に相互運用機能が強化されて堅牢性が向上しているほか、例外をトラップし診断する機能も強化されたことで、開発時間の短縮を実現し、より優れた相互運用アプリケーションを作成できるようになっています。

Visual Basic 2008 では、簡単に DLL が "COM 参照可能" であることを指定できます。つまり、Visual Basic 6.0 からアクセスすることを明示して、簡単に DLL を作成できます。アプリケーションの Visual Basic 6.0 の部分は、Visual Basic 6.0 アプリケーションのライフタイムの問題を継承しますが、このような Visual Basic 6.0 と Visual Basic 2008 のハイブリッドのアプリケーションにより、アプリケーションを Visual Basic 2008 に徐々に移行するプロジェクト チームは、より多くの選択肢を得ることができます。

関連リソース

相互運用	
VB Fusion: Visual Basic 6.0 アプリケーションの拡張 (英語)	http://msdn2.microsoft.com/en-us/vbrun/ms788241.aspx
Visual Basic Fusion: Visual Basic 6.0 と Visual Basic .NET を共有するための ベスト プラクティス (英語)	http://msdn2.microsoft.com/en-us/library/ms364069.aspx
販売サポート : Visual Basic 6.0 と Visual Basic 2005 のハイブリッド アプリケーション (英語)	http://msdn2.microsoft.com/en-us/library/cc300139(VS.80).aspx
Interop Forms Toolkit 2.0 (英語)	http://msdn2.microsoft.com/en-us/vbasic/bb419144.aspx
5000 個のクラスに興味はありますか? (Visual Basic 6 から .NET Framework の全)	http://www.microsoft.com/japan/msdn/vs/vb/VB5000Cl.aspx

相互運用

機能を使用する)	
Visual Basic 6.0 開発者向けの基本的なトレーニング : Visual Basic 相互運用機能の紹介 (英語)	http://msdn2.microsoft.com/en-us/vbrun/cc297223.aspx
"ハウツー" Visual Basic 相互運用機能についてのビデオシリーズ (英語)	http://msdn2.microsoft.com/en-us/vbasic/bb466226.aspx?wt.slv=topsectionsee#interop

移行

Visual Basic 6.0 アプリケーションを Visual Basic 2008 に移行 (アップグレード) すると、アプリケーション全体が Visual Basic 2008 ベースになります。通常、移行後のアプリケーションでは、アップグレード ツールで自動的に移行できないコードを変更するため、アプリケーションが移行後も参照している COM オブジェクトを置き換えるために、多少の手直しが必要になります。

移行は、一度に 1 コンポーネントずつ移行するなど、徐々に実行することも、一度にまとめて実行することもできます。元のアプリケーションが持つアーキテクチャ上の制限は、移行後のアプリケーションにも継承されます。アップグレードツールでは、アーキテクチャのリファクタリングは実行されません。

ベスト プラクティスは、まず Visual Basic 6.0 アプリケーションを評価し、Visual Basic 2008 ベースのアプリケーションでの手直しをできるだけ行わずに済むように変更を行います。その後、移行と手直しを実行して、Visual Basic 2008 の新機能によりアプリケーションを機能強化します。

移行と COM オブジェクトの置換が完了したら、アプリケーション全体が Visual Basic 2008 ベースとなり、Visual Basic 6.0 のサポート期間の制限から解放されます。

関連リソース

Visual Basic 6.0 アプリケーションの準備	
Visual Basic 6.0 開発者向けの基本的なトレーニング : Visual Basic アプリケーションのアップグレード (英語)	http://msdn2.microsoft.com/en-us/vbrun/cc297223.aspx
Visual Basic 6.0 to Visual Basic .NET Upgrade Assessment Tool (英語)	http://www.microsoft.com/downloads/details.aspx?FamilyId=10C491A2-FC67-4509-BC10-60C5C039A272&displaylang=en
Visual Basic 6.0 のコード アドバイザ (英語)	http://www.microsoft.com/downloads/details.aspx?FamilyId=A656371A-B5C0-4D40-B015-0CAA02634FAE&displaylang=en
Visual Basic .NET へのアップグレードに向けた Visual Basic 6.0 アプリケーションの準備 (英語)	http://msdn2.microsoft.com/en-us/library/aa260644.aspx

アプリケーションの移行

Microsoft Patterns and Practices ガイド 「Upgrading Visual Basic 6.0 Applications to Visual Basic .NET and Visual Basic 2005」	http://www.microsoft.com/downloads/details.aspx?FamilyID=3D99C39F-4E9A-4F45-A4C5-6C76ADFAB6FC&DisplayLang=ja
無償の書籍『Upgrading Microsoft Visual Basic 6.0 to Microsoft Visual Basic .NET』(英語)	http://msdn2.microsoft.com/en-us/vbrun/ms788236.aspx
Visual Basic 6.0 開発者向けの アップグレードリソース (英語)	http://msdn2.microsoft.com/en-us/vbrun/ms788239.aspx
Visual Basic アップグレード ウィザードの解体 (英語)	http://msdn2.microsoft.com/en-us/library/aa730876(VS.80).aspx
Visual Basic アップグレード ウィザードを使用する方法	http://support.microsoft.com/kb/317885

置換

アプリケーションを置換する場合、新たな市販の製品の購入、新たなアプリケーションの開発、または Visual Basic 6.0 システムの代わりに使用する別のアプリケーションの拡張の 3 通りの方法のうち 1 つを行うことになります。

置換は、アーキテクチャ、開発標準、要件への適合の面で、最も柔軟性の高い戦略です。Visual Basic 2008 は、それ以前のバージョンの Visual Basic よりも強力で高い機能性を備えています。Visual Basic での開発の経験がある場合は、Visual Basic 6.0 アプリケーションを Visual Basic 2008 ベースの新しいシステムに置換する際に応用できる知識とスキルの点で、有利な立場にあります。

関連情報

置換	
Visual Basic 6.0 ユーザー向けの新機能	http://msdn2.microsoft.com/ja-jp/library/ms172617(VS.90).aspx
Visual Basic でのアプリケーションの開発	http://msdn2.microsoft.com/ja-jp/library/k6a6etxs.aspx
Visual Basic の導入事例 (英語)	http://msdn2.microsoft.com/en-us/vbasic/ms789180.aspx
Microsoft Dynamics	http://www.microsoft.com/japan/dynamics/

VISUAL BASIC の 各バージョンのサポートの概要

Visual Basic のマイクロソフトのサポート ポリシーは、サポート ライフ サイクルのガイドライン (<http://support.microsoft.com/?LN=ja-jp&pr=lifecycle>) に準じています。

簡単に言うと、マイクロソフトには、実績のあるサポート モデルがあります。製品のサポートは、メインストリーム サポート、延長サポート、セルフヘルプサポートの 3 段階で提供しています。

"メインストリーム" サポートの期間中は、機能が追加されたり、設計についてのリクエストが受け入れられる可能性があります。また、修正プログラムの配布、無償の電話サポートとオンライン サポートの提供、セキュリティ更新プログラムの作成、有償サポートの提供も行われます。

"延長" サポートの期間中は、セキュリティ更新プログラムが作成され、いくつかの更新プログラムが配布される可能性があります。また、有償の修正プログラムと有償のサポートが提供されます。

"セルフヘルプ" サポートの期間中は、サポート技術情報の記事、オンラインの製品情報、およびオンラインのサポート Web キャストなど、オンライン コンテンツに無償でアクセスできます。

Visual Basic の各バージョンのサポートの概要

これまで Visual Basic では 10 回の製品リリースが行われました。計画時の参考になるよう、以下の表に、各バージョンの現在 (本稿執筆時) のサポート フェーズを示します。

バージョン	サポート ライフサイクルのフェーズ
Visual Basic 1.0 ~ Visual Basic 5.0	セルフヘルプ
Visual Basic 6.0	Visual Basic IDE は 2008 年 4 月まで延長サポート ランタイムは 2017 年まで延長サポート
Visual Basic 2002	2009 年 7 月まで延長サポート
Visual Basic 2003	2008 年 10 月までメインストリーム サポート、 2013 年 8 月まで延長サポート
Visual Basic 2005	2011 年 3 月までメインストリーム サポート、 2016 年 3 月まで延長サポート
Visual Basic 2008	メインストリーム サポート

まとめ

このホワイト ペーパーでは、Visual Basic 6.0 アプリケーションのサポートに利用できる 4 種類の戦略について説明しました。この 4 種類の各戦略を実現するためのテクノロジーが、現在では存在しています。Visual Basic 6.0 のリリースから Visual Basic 2008 のリリースまでの 10 年間に、マイクロソフトは、相互運用性、移行ツール、サポート、およびトレーニング資料を作成、改良、洗練して、Visual Basic 6.0 の COM ベースのインフラストラクチャから、ビジネスを新しい方向に進めるソリューションを提供できる .NET プラットフォームに、アプリケーションを移行するための対策をとれるようにしてきました。

謝辞

このホワイト ペーパーは、1999 ～ 2003 年まで Microsoft Visual Basic のプログラム マネージャを務めていた Ed Robinson 氏によって準備されました。同氏は、多数の Visual Basic のアップグレード ツールの設計者であり、『Upgrading Microsoft Visual Basic 6.0 to Microsoft Visual Basic .NET』の共著者でもあります。現在は、ActionThis, Ltd の CEO を務めています。そのほか、Visual Basic の主任プログラム マネージャであり、多数の Visual Basic の新しい相互運用性ツールの設計者でもある Paul Yuknewicz 氏にもご協力いただきました。