

FAQ／よくある質問

こちらの記事には未翻訳の箇所があります。

Migration from VB6 to .NET／VB6 から.NET への移行

- [My VB6 applications work well. Why should I want to port it to .NET?](#)／弊社の VB6 アプリケーションは問題なく動作しています。それを.NET へ移植したがるはずがありませんよ。
- [I read that .NET applications are slower than VB6 applications, because they are interpreted. Is that true?](#)／.NET アプリケーションは解釈されるため VB6 アプリケーションに比べると遅いと聞いています。それは本当ですか？
- [Can VB Migration Partner convert to C#?](#)／VB Migration Partner は C#に変換することはできますか？
- [I also read that Visual C# is better than VB.NET. Shouldn't I migrate to C#, then?](#)／Visual C#が VB.NET よりも良いとも聞いています。むしろ C#に移行するべきじゃないですか？
- [What are the pros and cons of an automatic translation tool vs. manually rewriting the application from scratch?](#)／一からアプリケーションを手作業で書きなおすことに対する自動変換ツールの利点と欠点は何ですか？

VB Migration Partner／製品について

- [What is VB Migration Partner?](#)／VB Migration Partner とは何ですか？
- [Who wrote it?](#)／誰が作ったのですか？
- [Visual Studio comes with a free VB6 conversion tool. Why should I pay for a similar product?](#)／Visual Studio は無償の VB6 変換ツールを搭載しています。類似品にお金を払うと思いますか？
- [How can VB Migration Partner reach such a high success ratio?](#)／VB Migration Partner はどうやって高い変換率を達成しているのですか？
- [What are migration pragmas?](#)／変換プラグマとは何ですか？
- [How does it relate to other migration tools, such as Code Advisor and Upgrade Assessment Tools?](#)／Code Advisor や Upgrade Assessment Tool のような他の移行ツールとの関係はどうなっていますか？
- [How does VB Migration Partner compare to other commercial VB6 converters?](#)／VB Migration Partner は市販されている他の移行ツールに比べてどうですか？
- [Do I need to be a .NET expert to leverage VB Migration Partner's full power?](#)／VB Migration Partner の性能を完全に引き出すために.NET のエキスパートになる必要がありますか？
- [Can VB Migration Partner manage large VB6 business applications?](#)／VB Migration Partner は巨大な VB6 ビジネスアプリケーションを扱うことができますか？
- [I need to extend and improve the VB6 application while the migration process proceeds. Does VB Migration Partner handle such a requirement?](#)／移行作業を進めながら VB6 アプリケーションを拡張したり改修する必要があります。VB Migration Partner はそのような要求に応じられますか？
- [What is the convert-test-fix cycle?](#)／convert-test-fix サイクルとは何ですか？
- [Can the convert-test-fix cycle really handle all migration issues?](#)／convert-test-fix サイクルは本当にすべての移行上の問題を扱うことができますか？
- [What is the Trace-Match methodology?](#)／Trace-Match メソッドロジーとは何ですか？
- [What are extenders?](#)／エクステンダーとは何ですか？

- Our VB6 application doesn't clearly separate the user interface, the business logic layer, and the data layer. Can your tool help in this respect?／私たちの VB6 アプリケーションはユーザーインターフェースとビジネスロジック層とデータ層が明確に分かれていません。この点について御社のツール役立ちますか？
- Where I can learn more about VB Migration Partner?／VB Migration Partner についてどこで学べますか？
- What if I need more assistance during the migration process?／移行作業中により多くの支援が必要が必要になったらどうすればいいですか？
- Why does VB Migration Partner emit more warnings than Microsoft Upgrade Wizard?／なぜ VB Migration Partner は Microsoft Upgrade Wizard に比べて多くの警告を出すのですか？

Technical／技術的な質問

- Which Visual Basic versions does VB Migration Partner convert from?／VB Migration Partner はどのバージョンの Visual Basic から変換できますか？
- Which Visual Basic .NET versions does it convert to?／どのバージョンの Visual Basic .NET に変換できますか？
- Do I need to run VB Migration Partner on a computer where VB6 is installed?／VB6 がインストールされた PC で VB Migration Partner を実行する必要がありますか？
- Do I need to run VB Migration Partner on a computer where Visual Studio .NET is installed?／Visual Studio .NET がインストールされた PC で VB Migration Partner を実行する必要がありますか？
- Can I test VB Migration Partner even if I haven't purchased Microsoft Visual Studio yet?／Microsoft Visual Studio をまだ購入していなかったとしても VB Migration Partner をテストすることはできますか？
- Which Windows versions does the tool support?／このツールはどのバージョンの Windows をサポートしていますか？
- Which VB6 controls does VB Migration Partner support?／VB Migration Partner はどの VB6 コントロールをサポートしていますか？
- Which VB6 features aren't supported?／サポートされない VB6 の機能はどれですか？
- Why these features aren't supported?／これらの機能はなぜサポートされないのですか？
- Do converted .NET applications depend on COM and ActiveX components?／変換された .NET アプリケーションは COM コンポーネントと ActiveX コンポーネントに依存しますか？
- Are converted applications "true" .NET Framework executables?／変換されたアプリケーションは「本当に」.NET Framework 上で実行可能ですか？
- Do converted .NET applications require a support library?／変換された .NET アプリケーションはサポートライブラリを必要としますか？
- Do converted .NET applications depend on Visual Basic 6 runtime?／変換された .NET アプリケーションは Visual Basic 6 のランタイムに依存しますか？
- Can VB Migration Partner convert 3rd-party ActiveX controls?／VB Migration Partner はサードパーティ製の ActiveX コントロールを変換できますか？
- What about ActiveX controls authored in VB6?／VB6 で署名された ActiveX コントロールについてはどうですか？
- What is the best migration strategy if the VB6 applications uses *a lot* of ActiveX controls?／VB6 アプリケーションが多数の ActiveX コントロールを使用している場合、最も良い移行方法は何ですか？
- Do you plan to provide support for additional ActiveX controls?／ActiveX コントロールに対するサポートを提供する予定はありますか？
- Can VB Migration Partner convert MTS/COM+ applications?／VB Migration Partner は MTS/COM+ アプリケーションを変換することができますか？

- Can VB Migration Partner convert NT service applications?／VB Migration Partner は NT サービスアプリケーションを変換することができますか？
- Do converted .NET applications run under Microsoft Vista and Windows 7?／変換された.NET アプリケーションは Microsoft Vista や Windows 7 で動作しますか？
- Do I need administrative privileges to run VB Migration Partner?／VB Migration Partner を実行するのに管理者権限は必要ですか？
- Can VB Migration Partner generate ASP.NET applications?／VB Migration Partner は ASP.NET アプリケーションを生成することができますか？

Purchasing VB Migration Partner／購入に関する質問

- How much does VB Migration Partner cost?／VB Migration Partner はいくらですか？
- What are the terms of end user license agreement (EULA)?／エンドユーザー使用許諾契約条項はどうなっていますか？
- Is the support library's source code available for purchase?／サポートライブラリのソースコードを購入することはできますか？
- Is it possible to get a Trial?／トライアル版を入手することはできますか？
- What do I have to do to get your free advice for a successful migration?／うまく移行するためにどのような無償アドバイスを受けるべきでしょうか？
- Is technical support included in the license price?／技術サポートはライセンス価格に含まれていますか？
- May I extend the subscription period?／ライセンスの有効期間を延長することはできますか？
- We provide migration services to 3rd party companies. Can we later transfer the license to our customers?／弊社ではサードパーティベンダーに対する移行サービスを提供していますが、将来、弊社の顧客にライセンスを譲渡することはできますか？
- What kind of guarantee do you offer that VB Migration Partner will correctly convert my VB6 application to .NET?／VB Migration Partner が VB6 アプリケーションを.NET へ正常に移行することについて御社ではどのように保証していますか？
- How many years will VB Migration Partner library be supported?／VB Migration Partner のライブラリは何年間サポートしてもらえますか？
- There will never be a charge for the support library?／サポートライブラリに対する課金はまったく発生しないのですか？

Migration from VB6 to .NET／移行に関する質問

- My VB6 applications work well. Why should I want to port it to .NET?／弊社の VB6 アプリケーションは問題なく動作しています。それを.NET へ移植したがるはずがありません。
 - VB.NET is modern programming language that offers powerful object-oriented features that allow you to write code that is more efficient and that can be maintained more easily. More importantly, VB.NET gives you full access to the virtually unlimited power of the .NET Framework and allows you to easily integrate your application with many great features that you can use to extend your application, such as Web Services.

VB.NET はより効率的でメンテナンスし易いコードを書くことができる強力なオブジェクト指向的機能を備えた最新のプログラミング言語です。さらに重要なことは、VB.NET が .NET Framework の事実上無限のパワーを

完全に利用できるばかりか、ウェブサービスのようなアプリケーションを拡張するために使用することができる多くの強力な機能でユーザーのアプリケーションを簡単に統合することができるということです。

Even if you don't need to extend your VB6 application, however, you should seriously weigh the necessity of porting it to the .NET Framework for at least two reasons. First and foremost, *your VB6 application might not work correctly under Microsoft Vista or Windows 7*. Microsoft has stated that they will continue to support the Visual Basic runtime under future versions of the operating system (but not the Visual Basic IDE), but the key point is that many manufacturer of ActiveX controls won't offer this degree of support. How many of them have already gone out of business or have discontinued support for their older products?

たとえ VB6 アプリケーションを拡張する必要がないとしても、少なくとも二つの理由から .NET Framework に移行する必要性について真剣に検討するべきです。まず最初に、*VB6 アプリケーションは Microsoft Vista や Windows 7 上では正常に動作しない可能性があるということです！* マイクロソフト社は将来の登場する OS についても Visual Basic のランタイムをサポートし続ける(しかし Visual Basic IDE はそうではありません)と公式に発表していますが、問題は ActiveX コントロールの製造元の多くは現状のサポートを継続しないであろうということです。そのような会社のうち、どれくらいがすでに、過去の製品を取り扱うビジネスから撤退、またはサポートの継続を取りやめたでしょうか？

The second point to consider is that Microsoft [ended support for Microsoft Visual Basic](#) in March 2008. Microsoft clearly considers “classic” Visual Basic as a dead language. It is the right time to move to the more modern .NET Framework architecture.

二つ目には考慮すべきことは、マイクロソフト社は 2008 年 3 月に [Microsoft Visual Basic のサポートを終了した](#)ことです。マイクロソフト社は完全に「古い」Visual Basic を死語だと見なしています。最新の .NET Framework アーキテクチャに移行するにはいい時期なのです。

A third important consideration is that VB6 applications are inherently 32-bit and can't leverage the power of newer 64-bit CPUs and operating systems. Conversely, .NET programs run well in both 32-bit and 64-bit modes, because the .NET Framework Just-in-Time (JIT) compiler generates the right CPU opcodes and the most efficient code in all cases. The performance boost can be quite significant for some kinds of code. Also, your users will perceive the “64-bit” tag as a clear indication that your app uses state-of-the-art techniques, which in turn will help sales.

三つ目は VB6 アプリケーションが本質的に 32 ビットであり、最新の 64 ビットの CPU と OS の性能を利用できないことです。理由は .NET Framework の JIT コンパイラがすべての条件において適切な CPU の命令コードと最も効率的なコードを生成するためです。特定のコードに対する性能向上はかなり顕著です。また、エンドユーザーはアプリケーションが最先端技術を使用しているという分かりやすい「64 ビット」のシールに気付くでしょうし、それは返ってセールスの役に立つでしょう。

- I read that .NET applications are slower than VB6 applications, because they are interpreted. Is that true?/.NET アプリケーションは解釈されるため VB6 アプリケーションに比べると遅いと聞いています。それは本当ですか？
 - It's a false myth. All .NET compilers produce the so-called IL (Intermediate Language) code, which is just-in-time-compiled (or JITted) one instant before it is executed and then runs at the highest speed. You can also use the NGen tool provided with the .NET Framework to compile the entire application at deployment time, so that you avoid the JIT-compilation.

それは間違った神話です。すべての.NET コンパイラは実行される直前に JIT コンパイルされて、最高速度で実行される、いわゆる IL(中間言語)コードを作成します。また、配布する際にアプリケーション全体をコンパイルするために.NET Framework と一緒に提供されている NGen ツールを使用することもできますので、JIT コンパイルを避けることもできます。

In most cases, migrated .NET applications run as fast (or as slow) as the original VB6 program. In half of the remaining cases .NET executables are faster than VB6 apps (for example, in array and object creations), in the other cases VB6 is faster. For 99% of the tasks that a business application performs – such as building the UI, interacting with the database, and so forth – speed differences are quite negligible.

ほとんどの場合、変換された.NET アプリケーションは元の VB6 プログラムと同じ速度で実行されますが、VB6 より高速に実行される場合もあります(例えば、配列やオブジェクトを作成する場合)し、VB6 の方が速い場合もあります。ビジネスアプリケーションが行うタスクの 99%—ユーザーインターフェースの構築やデータベースとの連携など—にとって、性能差はまったく取るに足りないものです。

There are two important exceptions to the abovementioned general rule. First, when you access COM objects from .NET you pay a small speed penalty, because the call goes through COM Interop, a layer of the .NET Framework that handles data marshaling between the managed (.NET) and unmanaged (VB6) world. This happens, for example, when you access the fields of an ADO Recordset. However, in most cases the COM Interop overhead is negligible; if it not, you should move to ADO.NET technology as soon as possible after you have completed the migration to .NET.

上述の一般的なルールに対して二つ例外があります。一つは、.NET から COM オブジェクトにアクセスするときに若干スピードが落ちます。理由は、マネージド(.NET)とアンマネージド(VB6)の間でデータのマーシャリングを扱う.NET Framework 層である COM 相互運用を通して呼び出すためです。例えば、ADO のレコードセットのフィールドにアクセスするときに起こります。しかし、COM 相互運用のオーバーヘッドは取るに足りないものであることがほとんどです。もしそうでないなら、.NET へ移行した後、できるだけ早く ADO.NET へ移行すべきです。

Second, string concatenations may be much slower or much faster under .NET, depending on which .NET string you use. The .NET Framework comes with two different string flavors: the standard String object and the StringBuilder object. If you perform hundreds or thousands concatenations on the same string, the standard String object can be many times slower than the VB6 string, and this fact has gained .NET with the reputation of being slower. On the other hand, if you use the StringBuilder object your .NET concatenations can be much faster than in VB6.

ふたつめは、使用する.NETの文字によって、文字連結が.NETでは非常に遅くなったり、速くなったりする可能性があるということです。.NET Frameworkは異なる二つの文字特性を持っています。標準文字オブジェクトとStringBuilderオブジェクトです。同じ文字の連結を何千回も行う場合、標準文字オブジェクトはVB6の文字より何倍も遅くなる可能性があります。またこの事実は.NETでは反復処理が遅くなっているということを示しています。一方、StringBuilderオブジェクトを使う場合、.NETの連結処理はVB6よりもはるかに高速となる可能性があります。

(Incidentally, please notice that VB Migration Partner automatically spots the code portions that would greatly benefit from using a StringBuilder. You can replace a String variable with a StringBuilder variable by just inserting one pragma.)

(ちなみに、注目すべきことにVB Migration PartnerはStringBuilderを使用するコードを自動的に生成します。ユーザーはプラグマを挿入するだけで、String変数をStringBuilder変数に置き換えることができます。)

- Can VB Migration Partner convert to C#? / VB Migration PartnerはC#に変換することはできますか？
 - VB Migration Partner can currently convert only to VB.NET, and supports VB2005, VB2008 and VB2010. We plan to have beta C# support in Q1 2012.

現行のVB Migration PartnerはVB.NET(VB2005、VB2008、VB2010をサポートしています)にのみ変換することができます。弊社では2012年の第1四半期にC#版のベータバージョンを準備する計画です。

- I also read that Visual C# is better than VB.NET. Shouldn't I migrate to C#, then? / Visual C#がVB.NETよりも良いとも聞いています。むしろC#に移行すべきじゃないですか？
 - The supposed supremacy of C# over VB is a false myth, fed mostly by C++ zealots who moved to C# and love putting curly braces everywhere. It is true that the first edition of C# had a few keywords that VB 2002 didn't have, but the gap has been filled in VB2008. As of this writing, virtually all major C# features are supported by VB.NET as well.

C#がVBより勝っているという—C#に移った、どこにでも波括弧をつけるのが好きなC++信者によって拡張された—噂は間違っています。初期のC#がVB2002にないキーワードを少し持っていたのは本当ですが、VB2008ではその差は埋まりました。これを書いている時点では、事実上、C#の主要な機能はすべてVB.NETでもサポートされています。

What many detractors of VB seem to (deliberately) ignore is that all .NET Framework languages are compiled to basically the same IL opcode sequence and that the JIT-compiler compiles this sequence to virtually the same sequence of CPU instructions. The bottom line: VB.NET is a first-class .NET citizen that is as fast as C# or any other .NET Framework language. You can read more about VB health [here](#) and [here](#).

VB を中傷する人々の多くが意図的に無視しているのは、.NET Framework 言語がすべて基本的に同じ中間言語の命令コードシーケンスにコンパイルされ、このシーケンスを JIT コンパイラが実際には同じ CPU の命令シーケンスにコンパイルするという事です。結論は、VB.NET は C# やその他 .NET Framework 言語と同じ速さの 1 等 .NET 言語だということです。VB の健全性についての詳細は [こちら](#) と [こちら](#) で読むことができます。

Going a bit more deeper: The only thing that C# has and that VB.NET hasn't is support for unsafe code, that is the ability to generate code that can't be verified by the .NET Framework's CLR. Unsafe code can speed up execution in some cases, but it deprives your application of some important benefits. As a matter of fact, very few C# developers use unsafe code at all. You can read more about the differences in syntax between the two languages [here](#) and [here](#). Even when developers find out the IL that the two languages generate [isn't exactly the same](#), they assert that the differences don't justify choosing one language over the other.

もう少し深く話せば、C# が持っていて、VB.NET が持っていない唯一の機能はアンセーフコードのためにサポートされており、.NET Framework によって検証できないコードを生成する機能です。アンセーフコードはある条件下で実行速度を上げることができますが、アプリケーションのいくつかの重要な利点を損ないます。実際のところ、アンセーフコードを使う C# 技術者はほとんどいません。二つの言語の構文の相違については [こちら](#) と [こちら](#) で詳細を読むことができます。二つの言語が生成する中間言語が [正確に同じものではない](#) ことを技術者たちが知ったときでさえ、結果は言語の選択の正当性を証明していないと主張しました。

The main problem of migrating from VB6 to C# is that C# lacks a few essential VB6 features (that are included in VB.NET). More specifically, all C# versions up to 2008 didn't support [late-binding and optional parameters](#), which made it very difficult to generate reasonably concise and readable C# code during the migration and still preserve functional equivalence with the original VB6 code.

VB6 から C# へ移行する際の主要な問題は C# が (VB.NET には含まれている) VB6 の基本的な機能のいくつかを持っていないことです。さらに厳密に言えば、バージョン 2008 までの C# のすべてが [遅延バインディングとオプションパラメータ](#) をサポートしていませんでした。そのことが変換中に合理的で簡潔で読みやすい C# のコードを生成することと、元の VB6 との機能的等価を維持することを非常に困難なものにしていました。

C# 2010 partially solves the problem of late-binding and optional parameters, but still has a *big* disadvantage over VB.NET: the lack of the On Error statement. If your original VB6 code frequently uses On Error, then the migration to C# can be a nightmare and it's going to last much longer than you might have anticipated.

C#2010 は部分的に遅延バインディングとオプションパラメータの問題を解決しましたが、VB.NET に対しては以前として *非常に* 不利な状況にあります。つまり、On Error ステートメントの不足です。元の VB6 コードが On Error を頻繁に使用している場合、C# への移行は悪夢のような作業となるばかりか、予想以上に時間がかかるでしょう。

- What are the pros and cons of an automatic translation tool vs. manually rewriting the application from scratch? / ーからアプリケーションを手作業で書きなおすことに対する自動変換ツールの利点と欠点は何ですか？

- The decision of using a code translator vs. rewriting the application from scratch should be based mainly (if not exclusively) on the balance of benefits and costs. If you have unlimited time, skill, and money, the re-writing approach is undeniably the way to go. In most real-world scenarios, however, you need to go to the market as soon as possible, have money constraints, don't have a team of experienced .NET developers, and can't afford to scrutinize each and every line of code and/or write thousand of unit tests to ensure that the .NET application behaves like the original one.

コード変換ツールを使うか一からアプリケーションを作り直すかは(排他的でないのなら)主にコストと利益のバランスに基づいて決められるべきです。もし、無制限の時間、技術力、資金があれば、作り直すという選択は否定しえないものです。しかし、実際はほとんどの場合、出来るだけ早く市場に出さなければならなかったり、コストの制約があったり、経験豊かな.NET 技術者がいなかったり、すべてのコードを精査して元のアプリケーションと同じように動作することを保証するための数多くの単体テストシナリオを作る余裕がなかったりします。

In some cases, the time-to-market factor shadows all other considerations. For example, if the VB6 application doesn't run well under Microsoft Vista – because of incompatible 3rd-party ActiveX controls or issues with Vista's security model – you might lose an unbearable number of sales opportunities if you don't migrate to .NET as quickly as possible.

製品化までの時間という要素がすべての問題について考慮されなければならないこともあります。例えば、VB6 アプリケーションは Microsoft Vista ではうまく動作しなかったら—サードパーティの ActiveX コントロールの互換性がないことや Vista のセキュリティポリシーのせいで—できるだけ早く.NET へ移行しなかったら、耐えたい数の販売機会を逸失するかもしれないのです。

Another factor that should be weighed accurately is whether the original VB6 application must continue to evolve during the migration process, as it's often the case with business applications that are continuously update to comply with customers' requests, current laws, etc. In such scenarios, keeping the migration time as short as possible becomes a critical factor.

正確に検討すべきその他の要素は、移行中に VB6 アプリケーションを改修し続けるかどうかということです。顧客の要求や、法改正などに応じるために改修を続けるビジネスアプリケーションは珍しくありません。そのような場合、できるだけ短時間で移行することは決定的に重要な要素となります。

VB Migration Partner／製品に関する質問

- What is VB Migration Partner?／VB Migration Partner とは何ですか？
 - VB Migration Partner is an innovative tool that converts VB6 applications to VB.NET. It has been designed from the ground up using advanced parsing algorithms and dramatically reduces the overall time and cost of the migration process.

VB Migration Partner は VB6 アプリケーションを VB.NET へ変換する革新的なツールです。それは高度な解析アルゴリズムを使って一から作られており、移行プロセス全体の時間とコストを大幅に削減します。

- Who wrote it? / 誰が作ったのですか？

- Code Architects' VB Migration Partner is a five year effort by a team of top-notch Visual Basic and .NET Framework developers led by **Francesco Balena**, renowned speaker and author of seven books for Microsoft Press, including [Programming Microsoft Visual Basic 6](#) and all the books of the Programming Microsoft Basic series down to [Programming Microsoft Visual Basic 2005](#).

コードアーキテツ社の VB Migration Partner は **Francesco Balena** 率いる Visual Basic と .NET Framework の超一流の技術者チームによる 5 年に渡る努力の結晶です。彼は Microsoft Press から出版された 7 つの書籍 ([Programming Microsoft Visual Basic 6](#) と [Programming Microsoft Visual Basic 2005](#) までの Programming Microsoft Basic シリーズすべて) の著者であり著名な講演家でもあります。

The developer in charge for the user interface and the report features is **Marco Bellinaso**, author of the best selling [ASP.NET 2.0 Website Programming: Problem - Design - Solution](#). Marco's TheBeerHouse application has been selected by Microsoft for inclusion in the official list of [ASP.NET Start Kit Web Sites](#).

ユーザーインターフェースとレポート機能を担当した技術者は **Marco Bellinaso** です。彼はベストセラー [ASP.NET 2.0 Website Programming: Problem - Design - Solution](#) の著者です。Marco の TheBeerHouse アプリケーションはマイクロソフト社によって選ばれ [ASP.NET Start Kit Web Sites](#) の公式リストに含まれています。

Giuseppe Dimauro, MSDN Regional Director for Italy and co-author of [Practical Guidelines and Best Practices for Microsoft Visual Basic and Visual C# Developers](#) by Microsoft Press, took care of fine-tuning, testing and hardcore debugging together with a great team of senior and junior developers.

MSDN のイタリア担当ディレクターだった **Giuseppe Dimauro** は Microsoft Press の [Practical Guidelines and Best Practices for Microsoft Visual Basic and Visual C# Developers](#) の共著者です。彼には微調整、テスト、徹底的なデバッグを技術者チームと共に行って頂きました。

Read more about Code Architects and the VB Migration Partner Team [here](#).

コードアーキテツ社と VB Migration Partner Team については[こちら](#)でお読み頂けます。

- Visual Studio comes with a free VB6 conversion tool. Why should I pay for a similar product? / Visual Studio は無償の VB6 変換ツールを搭載しています。類似品にお金を払うと思いますか？

- When measuring the cost-effectiveness of the automatic migration approach, all depends on the specific translator. If the translator doesn't have a high success rate, many developers may (correctly) estimate that cost/time savings don't compare favorably with the opportunity of re-designing the application from the ground up.

自動変換ツールを使用する移行の費用対効果を測る場合、すべて個々の変換ツールに依存します。変換ツールの変換率が高くない場合、ほとんどの技術者は削減できる費用や時間は一からアプリケーションを作り直す場合と比較しても大きくないことは正確に推測できるかもしれません。

The Upgrade Wizard that comes with Visual Studio does a barely decent job in translating VB6 code to VB.NET, but it fails to correctly convert several VB6 language features that VB Migration Partner fully supports. Our tests on a [random assortment of VB6 applications](#) downloaded from the Internet show that VB Migration Partner generates up to **10-15x fewer compilation errors** than the Upgrade Wizard. VB Migration Partner can convert most of the language features that have been considered as “not automatically convertible”, such as Gosubs, arrays with nonzero lower index, As Any and callback parameters in Declares, auto-instancing variables, deterministic finalization, and more. VB Migration Partner makes the difference when converting large business applications.

Visual Studio に付属している Upgrade Wizard は VB6 のコードから VB.NET への変換においてとても満足できるものとは言えないばかりか、VB Migration Partner が完全にサポートする VB6 のいくつかの言語的特徴を正しく変換することもできません。インターネットから[ランダムにダウンロードした複数の VB6 アプリケーション](#)に対する弊社のテストの結果、VB Migration Partner は Upgrade Wizard に比べて**コンパイルエラーが 1/10 から 1/15**となりました。VB Migration Partner は、Gosubs や下限が 0 ではない配列や As Any や Declare のコールバックパラメータや自動インスタンス化変数や決定性ファイナライゼーションなどの「自動変換できない」と考えられていた言語的特徴のほとんどを変換することができます。VB Migration Partner は巨大なビジネスアプリケーションを変換する場合にも効果があります。

- How can VB Migration Partner reach such a high success ratio? / VB Migration Partner はどうやって高い変換率を達成しているのですか？
 - Three main factors contribute to VB Migration Partner's superior performance. First, unlike some tools that are based on generic conversion engines that have been tailored to handle VB syntax, VB Migration Partner's **parser engine** has been designed from scratch with the only purpose to understand VB6 syntax. Second, VB Migration Partner comes with a comprehensive **support library** that virtually eliminates the differences between the two languages. For example, VB Migration Partner can convert VB6 applications that use graphic statements, custom coordinate systems, drag-and-drop, Windows subclassing, DAO, RDO, and ADO data binding (including variations such as ADO data source and data consumer classes), DDE (Dynamic Data Exchange) protocol, and a lot more. Third, VB Migration Partner recognizes the so-called **migration pragmas**.

VB Migration Partner の優れた性能の主な要因は 3 つあります。第 1 に、VB の構文を扱うために作られた市販の変換エンジンを核とするツールと異なり、VB Migration Partner の**解析エンジン**は VB6 の構文を解析する

目的のためだけに一から作られています。第2に、VB Migration Partner は二つの言語の違いを事実上無効にする包括的な**サポートライブラリ**を持っています。例えば、VB Migration Partner は画像やドラッグ & ドロップやサブクラスウインドウや DAO や RDO や ADO データバインディング (ADO データソースやデータコンシューマクラスを含む) や DDE (動的データ交換) プロトコルなど (他多数) を取り扱う VB6 アプリケーションを変換することができます。第3に、VB Migration Partner はいわゆる**変換プラグマ**を使用します。

- What are migration pragmas? / 変換プラグマとは何ですか？

- Migration pragmas are special remarks that developer can insert in the original VB6 code to drive the migration process and to help VB Migration Partner to generate better .NET code. For example, pragmas can dictate how auto-instantiating variables must be translated, whether an object requires deterministic finalization, or if support for default properties in late-bound mode is necessary. Pragmas can be used also to optimize the resulting application and decide what kinds of warnings VB Migration Partner should emit.

変換プラグマは変換プロセスを操作し、VB Migration Partner のコード生成を支援するために技術者が元の VB6 コードに挿入することができる特別なコメントです。例えば、プラグマは自動インスタンス化変数をどのように変換するかを決めたり、オブジェクトが決定性ファイナライズ処理を必要とするかどうかを決めたり、遅延バインディングのデフォルトプロパティの対応が必要かどうかを決めたりすることができます。また、プラグマは変換後のアプリケーションを最適化するためや VB Migration Partner がどのような警告を出すべきかを決めるために使うこともできます。

Pragmas can be defined at the solution, project, file, method, and individual variable lever, a granularity that offers huge advantages over conversion tools from other vendors, which only offer project-level settings. For example, you can have a group of project-level pragmas that establish the general conversion rules, plus a few other pragmas that handle exceptions to these rules inside specific files or methods.

プラグマはソリューション、プロジェクト、ファイル、メソッド、そして個々の変数のレベルで定義することができます。また、プロジェクトレベルでの設定は他社の変換ツールに対して非常に大きなアドバンテージとなる粒度です。例えば、ユーザーはよく使う変換ルールを定めたプロジェクトレベルプラグマのグループを作ることができますし、これらのルールがエラーとなってしまう特定のファイルやメソッドに対応するための他のプラグマを追加することもできます。

- How does it relate to other migration tools, such as Code Advisor and Upgrade Assessment Tools? / Code Advisor や Upgrade Assessment Tool のような他の移行ツールとの関係はどうなっていますか？

- In addition to the Upgrade Wizard included in Visual Studio, Microsoft offers two tools that can be useful during the migration process. [Code Advisor](#) quickly scans a VB6 application and highlights code constructs that can't be converted to VB.NET; the [Upgrade Assessment Tool](#) provides a cost estimation of the migration process.

Visual Studio の Upgrade Wizard の他に、マイクロソフト社は移行において役立つふたつのツールを提供しています。[Code Advisor](#) は VB6 アプリケーションを高速スキャンして VB.NET に変換できないコードを強調表示します。また、[Upgrade Assessment Tool](#) は移行にかかるコストを見積ってくれます。

VB Migration Partner embeds and augments the functionality of these two utilities. In fact, VB Migration Partner matches the Upgrade Assessment Tool's features, in that it allows you to generate project metrics, export them to Microsoft Excel, evaluate cost and time of the entire process, and customize the report as you see fit. (Notice that this feature requires that Microsoft Excel be installed on the local computer.) As for the Code Advisor tool, VB Migration Partner parser and code generation engine is so fast that it can convert an entire VB6 application, calculate code metrics, and optimize the resulting .NET code in about the same time that Code Advisor takes to create its reports.

VB Migration Partner はこれら二つのツールの機能を組み込み補強しました。実際に、VB Migration Partner は Upgrade Assessment Tool と比べて、プロジェクトメトリクスを生成したり、Excel にエクスポートしたり、工程全体の工数を見積ったり、レポートを好きなようにカスタマイズできるという点では機能的に遜色ありません。(ただし、この機能を使用するには Excel を PC にインストールしておく必要があります。) Code Advisor についてはどうかというと、VB Migration Partner の解析プログラムとコード生成エンジンはとても高速なので、Code Advisor がレポートを作成するのにかかる時間で、VB6 アプリケーション全体を変換し、コードメトリクスを測定し、結果として得られた .NET コードを最適化することができます。

- How does VB Migration Partner compare to other commercial VB6 converters? / VB Migration Partner は市販されている他の移行ツールに比べてどうですか？

- VB Migration Partner was officially launched in May 2008, but we began working on it in January 2006. At that time there was only another VB6 converter on the market, namely Artinsoft VB Upgrade Companion, which had no competitors to be worried about. We were newcomers to this market and of course worked very hard to deliver [a product that had to be significantly better](#) than theirs, otherwise all our work would have been void.

VB Migration Partner は 2008 年 5 月に公式リリースされましたが、私たちは 2006 年 1 月に作業を開始しました。当時は市販の VB6 変換ツールは Artinsoft の Upgrade Companion 一つしか存在せず、脅威となる競合もいませんでした。私たちはこの市場に新規参入し、[はるかに優れた製品](#)を送り出すために非常な努力を払いました。また、そうしなければ私たちの努力は無意味なものになっていたでしょう。

As of this writing, any unbiased test can prove that VB Migration Partner has more features and is more powerful than any other converter on the market. Our competitors are trying to catch up and now release new versions much more frequently than in the past, yet they still have to match the majority of features we introduced in version 1.0, such as support for Gosubs, graphic statements, drag-and-drop, and others.

これを書いている今、公平なテストによって VB Migration Partner がより多くの機能を持ち、市販されているどの変換ツールよりも強力であることが証明されました。競合他社は今まで以上に頻繁にバージョンアップして

追いつこうとしています、依然として彼らは、Gosubs や画像処理文やドラッグ & ドロップなどのような弊社のバージョン 1.0 における主要な機能を持っていません。

In spite of our technical advantage, we continue to relentlessly improve VB Migration Partner and add great features at each new release, such as the great [Trace-Match methodology](#) that allows you to dramatically reduce the time you spend to establish functional equivalence with the original VB6 application. No other vendor in this market segment is so reactive to customers' needs.

技術的に有利な立場であるにもかかわらず、私たちは熱心に VB Migration Partner を改良し新しいバージョンでは、元の VB6 アプリケーションとの機能的等価を達成するための時間を飛躍的に短縮することができる [Trace-Match メソッドロジー](#) のような優れた機能を追加しています。この市場において弊社はもっとも速く顧客の必要に応えています。

- Do I need to be a .NET expert to leverage VB Migration Partner's full power? / VB Migration Partner の性能を完全に引き出すために .NET のエキスパートになる必要がありますか？

- For sure, a certain degree of familiarity with both the source and the target language helps during the migration process. This is true for all conversion tools. However, VB Migration Partner doesn't strictly require that you are a VB6 or a .NET guru because the support library hides most of the differences between the two languages. For example, VB Migration Partner automatically converts graphic methods even if the .NET way to implement them is completely different from the VB6 way, and doesn't require that you study all the nitty-gritty details of the GDI+ portion of the .NET Framework.

確かに、移行の際、ソースと対象言語に対するある程度の知識は役に立ちます。これはどのような変換ツールに対しても真理です。しかし、VB Migration Partner はユーザーが VB6 や .NET のエキスパートであることをそれほど必要としません。なぜなら、サポートライブラリが二つの言語間の相違を隠してくれるからです。例えば、VB Migration Partner は自動的に画面メソッドを変換します。それがたとえ、.NET のやり方が VB6 のやり方とまったく異なっていたとしてもです。だから、.NET Framework の GDI+ の詳細を全部理解する必要はありません。

To recap, VB Migration Partner allows you to have a running .NET application without having to train your *entire* team before facing the migration effort. This detail alone makes VB Migration Partner a clear winner over the free Upgrade Wizard and other conversion tools.

要点をまとめると、VB Migration Partner は移行作業に入る前にチーム全員がトレーニングすることなく、実行可能な .NET アプリケーションを作ることができます。このことだけでも VB Migration Partner が Upgrade Wizard やその他の変換ツールに勝っていることは明白です。

- Can VB Migration Partner manage large VB6 business applications? / VB Migration Partner は巨大な VB6 ビジネスアプリケーションを扱うことができますか？

- Sure, in many ways. First, unlike the Upgrade Wizard tool, VB Migration Partner supports the migration of VB6 **project groups**, a feature that saves time during the conversion of large projects consisting of two or more projects and assures that cross-project calls are converted correctly. Second, VB Migration Partner includes sophisticated code analysis techniques that let developers mark unused classes and methods, so that they can focus on the code that actually needs to be converted. Third, VB Migration Partner can generate reports that include code metrics and estimation of the time and cost of the migration process. If your app is divided in dozens or hundreds distinct VB6 projects, you can use VB Migration Partner's **command-line version** for batch conversions and generate a single report file that shows the result of each individual project.

もちろん、多くの方法があります。第 1 に、Upgrade Wizard は違って、VB Migration Partner は VB6 **プロジェクトグループ**を変換することができます。この機能は複数のプロジェクトで構成される巨大なプロジェクトを変換する時間を短縮してくれます。また、プロジェクト間の呼び出しも正常に変換されることが保証されています。第 2 に、VB Migration Partner は使用されていないクラスやメソッドを通知する高度な解析機能を備えているので、技術者は実際に変換する必要のあるコードに集中することができます。第 3 に、VB Migration Partner は移行にかかる時間と費用の見積とコードメトリクスを含むレポートを作成することができます。もしアプリケーションが非常に多くの VB6 プロジェクトに分割されていたら、一括変換するために VB Migration Partner の **コマンドライン版**を使用して、個々のプロジェクトの結果を含む一つのレポートを作成することができます。

Last but not the least: VB Migration Partner lets you do your chores in less time simply because it's much faster than other similar tools: it can be up to **8 times faster** than the Upgrade Wizard and other converters that use the same engine as Upgrade Wizard and can convert 1-20000 lines per second on a modern computer. (Notice that this timing includes additional tasks such as advanced code analysis, dead code detection, code refactoring, and creation of code metrics.)

最後にとっても大切なことですが、VB Migration Partner は他の同様なツールに比べてはるかに高速であるため、面倒な作業を短時間で行うことができます。Upgrade Wizard や Upgrade Wizard と同じエンジンを使用する他の変換ツールに比べて最大 **8 倍も速く**、最新の PC なら 2 万行を 1 秒で変換することができます。(注目すべきは、この時間には高度なコード解析やデッドコードの検出やコードリファクタリングやコードメトリクス作成のような付加的作業が含まれていることです。)

Here's another feature that is essential in converting large N-tier applications split in multiple projects: VB Migration Partner can optionally preserve all project GUIDs when it regenerates a VB.NET project. If you have one or more Visual Studio 2005/2008/2010 solutions that include that project, all references are correctly preserved and you don't have to manually rebuild the solution.

次の機能は、複数のプロジェクトに分割された巨大な N 層アプリケーションを変換するには必要不可欠です。VB Migration Partner は VB.NET プロジェクトを再生成するときにすべてのプロジェクトの GUID を失わないようにすることができます。そのようなプロジェクトを含む Visual Studio 2005/2008/2010 のソリューションがある場合、すべての参照はすべて正しく維持され、ソリューションを手作業でリビルドする必要はありません。

Finally, VB Migration Partner even allows you to migrate only a *fraction* of an existing VB6 project, for

example just a few forms or a single module. [This important feature](#) allows you to focus your attention on a form or a class at a time, which means that you don't have to worry on the hundreds compilation errors that you might receive if migrating a huge VB6 project in one shot. Needless to say, no other VB6 converter offers anything similar.

最後に、VB Migration Partner は既存の VB6 プロジェクトの一部を変換することもできます。例えば、少数のフォームやモジュール単体の場合です。[この重要な機能](#)によって、フォームやクラスにだけ集中することができます。これは、巨大な VB6 プロジェクトを変換する場合に発生する大量のコンパイルエラーを気にしなくてもよいことを意味します。言うまでもありませんが、他の VB6 変換ツールには同様の機能は存在しません。

- I need to extend and improve the VB6 application while the migration process proceeds. Does VB Migration Partner handle such a requirement? / 移行作業を進めながら VB6 アプリケーションを拡張したり改修する必要があります。VB Migration Partner はそのような要求に応じられますか？
 - All migration tools from other vendors force you to work on a “snapshot” of the original VB6 application; if you later extend or modify the VB6 code, the converted .NET application won't include all these fixes and enhancements. If the migration process takes weeks or months – quite a reasonable assumption for real world business applications – the neat result is that the .NET application is already outdated by the time it finally compiles and runs correctly. VB Migration Partner copes with this issue by introducing the concept of **convert-test-fix cycle**.

他のベンダーの変換ツールはどれも元の VB6 アプリケーションの「スナップショット」に対して実行しなければなりません。すなわち、後になって VB6 のコードを改変したり修正する場合、変換済みのアプリケーションはこういう改変や修正を含めることができません。もし、移行作業が数週間から数カ月に及ぶ場合—実際のビジネスアプリケーションにとっては現実に起こりえることです—、.NET アプリケーションが最終的にコンパイルされ正常に動作するころにはすでに古くなってしまふことが想定されます。VB Migration Partner はこの問題を **convert-test-fix** サイクル理論によって解決します。

- What is the convert-test-fix cycle? / convert-test-fix サイクルとは何ですか？
 - Typically, migrating a real-world VB6 application is an iterative process that goes through the following three stages. First, you **convert** the original VB6 code and **test** the result; odds are that the converted .NET project contains invalid code that causes compilation errors, thus you must understand how to **fix** the original code to avoid these errors. However, instead of modifying the VB6 application or the converted .NET application, VB Migration Partner allows you to just add one or more pragmas to the original VB6 code: pragmas are just remarks, therefore they can't affect the way the VB6 application works. You can then re-submit the “pragma-enhanced” VB6 project to VB Migration Partner, see the results, and add more pragmas if necessary, and continue this process until no more compilation errors occur.

通常、VB6 アプリケーションの移行は次に述べる 3 つのプロセスを反復します。第 1 に、元の VB6 コードを**変**

換し、結果をテストします。すなわち、変換済み.NET プロジェクトがコンパイルエラーを引き起こす不正なコードを含む可能性があるため、これらのエラーを回避するように元のコードを修正する方法を見つけなければなりません。しかし、VB Migration Partner では、VB6 アプリケーションや変換後の .NET アプリケーションを修正する代わりに、元の VB6 コードにプラグマを挿入するだけなのです。プラグマはただのコメントですので、VB6 アプリケーションの動作には何の影響も与えません。そのため、ユーザーは「プラグマが挿入された」VB6 プロジェクトを VB Migration Partner に投入し、結果を確認し、必要ならばさらにプラグマを追加することができます。この工程をコンパイルエラーがなくなるまで繰り返します。

When the converted .NET project compiles correctly you are now at about the middle of the migration process, because you can now actually run the .NET project and see how it behaves. Again, odds are that you'll experience runtime exceptions or other sorts of malfunctioning. You take note of these problems, understand how to avoid them – our [Knowledge Base](#) is the best source of information for this purpose – and fix the original VB6 code by adding more pragmas. Next, you convert the VB6 code again, test the result, and fix the original code with more pragmas until the code both compiles and executes correctly.

変換された.NET プロジェクトが正常にコンパイルできたら、移行は中間地点です。その理由は、この時点で.NET プロジェクトを実際に行うと、動作を確認することができるからです。繰り返しますが、ランタイムエラーやその他の不具合がでる可能性はあります。ユーザーはこれらの問題に対する回避方法を探し—弊社の[サポート技術情報](#)はこの目的の最高の情報源です—さらにプラグマを追加して元の VB6 コードを修正します。次に、もう一度、VB6 コードを変換し、結果をテストし、コードのコンパイルと実行が正常に行われるまで元のコードにプラグマを追加します。

It is essential to notice that during all the iterations of the convert–test–fix cycle the original VB6 code is untouched (except for the added pragmas, which are just remarks and don't affect how the VB6 application behaves), therefore all features and bug–fixes added to the original code in the time window necessary to complete the migration process are converted to .NET as expected. Only a migration tool that supports pragmas (or equivalent mechanism) can guarantee that the converted application is always in synch with the original VB6 code.

convert–test–fix サイクルを繰り返す間、元の VB6 コードには手が加えられていない(プラグマの追加は除きます。これは単なるコメントで VB6 アプリケーションの動作に影響を与えません)ことは言うまでもありません。そのため、移行期間中の全機能と元のコードに行われたバグ修正は予想通りに変換されます。プラグマ(または同等の機能)をサポートする変換ツールだけが、変換後のアプリケーションと元の VB6 コードが常に同期することを保証できます。

- Can the convert–test–fix cycle really handle all migration issues? / convert–test–fix サイクルは本当にすべての移行上の問題を扱うことができますか？
 - Pragmas are usually enough powerful to convert any VB6 application into a .NET solution that has no compilation errors. Pragmas are also useful to get rid of most runtime errors and misbehaviors, but realistically they can't solve all the problems that might occur after the conversion.

プラグマは VB6 アプリケーションをコンパイルエラーのない .NET ソリューションに変換するには十分な機能を持っていることがほとんどです。プラグマはほとんどのランタイムエラー、誤動作の除去にも役立ちますが、現実的には変換後に発生する問題すべてを解決することはできません。

For example, pragmas can't help if the original application uses the DataReport designer or one or more unrecognized ActiveX controls. In such cases, you must manually fix the converted .NET application. If you postpone such manual fixes at the end of the automatic conversion process you can reduce to the minimum the period during which the original VB6 and the converted .NET applications are out-of-sync.

例えば、プラグマは元のアプリケーションが DataReport デザイナーや未知の ActiveX コントロールを使用している場合は役に立ちません。そのような場合、ユーザーは手作業で .NET アプリケーションを変換しなければなりません。もし、自動変換工程の終わりまで手作業による改修を延期するなら、元の VB6 アプリケーションと変換後の .NET アプリケーションの非同期期間は最小化できるでしょう。

- What is the Trace-Match methodology? / Trace-Match メソッドロジとは何ですか？
 - When migrating from VB6 it is essential to ascertain that the generated code preserves full functional equivalence with the original application. However, “proving” that such equivalence is preserved is easier said than done. In many cases, just comparing the result on the user interface or the data stored in the database isn't enough, because you are seeing the final effects, not all the intermediate steps. It's hard to reach the certainty that the converted code will behave like the original VB6 code in all circumstances.

VB6 からの移行の際、変換後のコードが元のアプリケーションと同等の機能を維持していることを確認することは不可欠です。しかし、そのような等価を証明することは「言うは行うより易し」です。ほとんどの場合、ユーザーインターフェースやデータベースに格納されたデータの結果を比較するだけでは十分ではありません。なぜなら、ユーザーは最終結果のみを確認しているのであって、中間コード全体を確認しているわけではないからです。変換後のコードがどのような状況においても元の VB6 と同じ動作をすることを確実なものにすることはとても大変なことです。

[Trace-Match methodology](#) is a new VB Migration Partner (added in version 1.32) that can help you in determining that the VB6 and the .NET code work exactly in the same manner, create objects and invoke methods in the same order, use the same values for private variables, and so on.

[Trace-Match メソッドロジ](#)は、オブジェクト作成やメソッドの呼び出しが同じ順番で行われることやプライベート変数に対して同じ値を使うことなど、VB6 と .NET のコードが正確に同じ動作をすることを確認するのに役立つ新しい VB Migration Partner (バージョン 1.32 で追加されました) です。

From a technical perspective, Trace-Match is the combination of a VB6 tracing library, a .NET tracing library, and the ability to automatically insert (and remove) thousands trace statements by means of a simple click. Once all the trace statements are in place, you just need to perform a set of test cases on both the VB6 app and the converted .NET app, and check that the produced trace files are identical (or at least, very similar).

技術的な観点から言えば、Trace-Match は VB6 トレーシングライブラリと .NET トレーシングライブラリの組み合わせであり、クリック一つで大量のトレース文を自動的に挿入（または削除）することができます。すべてのトレース文が配置されれば、後は VB6 アプリケーションと変換後の .NET アプリケーションの両方でテストケースを実行し、生成されたトレースファイルが同一である（少なくとも、非常に似ている）ことを確認するだけです。

Trace Match also comes with basic profiling features, thus it's easier to immediately detect when the original or the converted application spend most of their time. Trace output can go to a file or be displayed by means of a tool such as [DebugView](#).

Trace Match は基本的なプロファイリング機能も備えているため、元のアプリケーションと変換後のアプリケーションのどちらに大半の時間がかかっているを見つけ出すのは容易なことです。トレース結果はファイルに出力することも [DebugView](#) のようなツールを使って表示することも可能です。

- What are extenders? / エクステンダーとは何ですか？
 - VB Migration Partner has an open architecture and supports plug-ins (or *extenders*) that can participate and alter the migration process to add support for special features. An extender is a class that implements a special interface that VB Migration Partner recognizes and uses to communicate with the extender. The extender class must be compiled in a DLL and deployed in VB Migration Partner's main directory.

VB Migration Partner は特別な機能をサポートするために移行プロセスを共有したり変更したりすることができるオープンアーキテクチャを持っており、プラグイン（もしくはエクステンダー）をサポートしています。エクステンダーは VB Migration Partner が認識したり通信したりする特別なインターフェースを実装したクラスです。エクステンダークラスは DLL としてコンパイルされ、VB Migration Partner のメインディレクトリに格納される必要があります。

Right out of the box, we provide an extender that eliminates unused constant and Declare statements and another extender that refines the code produced by the conversion of UserControl classes. You can enable or disable individual extenders from the Extenders tab of the Options dialog box.

細かい設定をする必要はありません。私たちが提供するあるエクステンダーは使用されていない定数や宣言を取り除き、あるエクステンダーはユーザーコントロールクラスの変換によって生成されたコードを最適化します。ユーザーはオプションダイアログボックスのエクステンダータブから個々のエクステンダーを有効にしたり無効にしたりできます。

We use extenders both to add new features to VB Migration Partner and to ease the migration of large and complex business application. For example, the migration of a VB6 application that uses a custom-made library could be improved by writing an extender that recognizes calls to that library and converts them appropriately.

私たちは VB Migration Partner に新しい機能を追加したり、巨大で複雑なビジネスアプリケーションの移行を容易にするためにエクステンダーを使用します。例えば、自作のライブラリを使用している VB6 アプリケーションの移行はそのライブラリの呼び出しを認識し適切に変換するエクステンダーを作成することで改善できる可能性があります。

- Our VB6 application doesn't clearly separate the user interface, the business logic layer, and the data layer. Can your tool help in this respect?／私たちの VB6 アプリケーションはユーザーインターフェースとビジネスロジック層とデータ層が明確に分かれていません。この点について御社のツール役立ちますか？

- An automatic translator can't do much to change the overall architecture. You can solve this issue only by rewriting the application. Strictly speaking, it isn't a "porting" any longer and isn't a "re-writing from scratch" either, because existing users expect that the new application behaves as closely as possible to the original one and you can't completely change the way they interact with the user interface.

自動変換と言ってもアーキテクチャ全体を変えてしまうわけではありません。ユーザーはこの問題をアプリケーションを書き換えることでのみ解決することができます。厳密に言うと、それはもはや移植ではなく、一から書き直すことでもありません。なぜなら、システムの利用者は新しいアプリケーションが元のものと限りなく同じように動作することを期待しているのですから、ユーザーインターフェースを介して相互作用する方法を完全に変更することはできません。

- Where I can learn more about VB Migration Partner?／VB Migration Partner についてどこで学べますか？

- We plan to offer the best technical support through the <http://www.vbmigration.com/> web site. We made all the documentation available on the Internet, including training videos and code samples. A section of the web site is devoted to "generic" migration topics: we share our knowledge with all VB6 developers who wish to migrate their code and their skills to .NET Framework and VB.NET, even if they aren't interested in VB Migration Partner. For example, check out the [list of differences](#) between VB6 and VB.NET, many of which have been never documented anywhere else.

私たちはウェブサイトを使って最良の技術サポートを提供していくつもりです。私たちはすべてのドキュメントをインターネットに公開しました。そこにはトレーニングビデオやサンプルコードも含まれます。あるセクションは「一般的な」移行のトピックスとなっており、私たちは、.NET Framework と VB.NET にソースコードや技術を移行したい VB6 技術者達に情報を提供しています。たとえ、彼らが VB Migration Partner に興味がなかったとしてもです。例えば、VB6 と VB.NET の [相違点の一覧](#) をご覧ください。それらの多くは他の場所では決して文書化されてこなかったものです。

- What if I need more assistance during the migration process?／移行作業中により多くの支援が必要が必要になったらどうすればいいですか？

- Code Architects, the makers of VB Migration Partner, is willing to provide the best custom support for large migration processes, regardless of whether you plan to use our tool. We are based in Southern Italy and we can travel to anywhere in Europe, or even overseas. In most practical cases, however, we can offer a good level of support remotely, via phone, Skype, email, chat, etc. [Send us an email](#) for more information.

御社が弊社のツールを使うつもりかどうかに関わらず、VB Migration Partner の開発元であるコードアーキテクト社は巨大な移行作業に対しては喜んでカスタムサポートを提供いたします。弊社は南イタリアを拠点としており、ヨーロッパのどこへでも出張いたします。それが海外であってもそうです。しかし、実際にはほとんどの場合、弊社は十分なレベルのサポートを遠隔地からでも、電話やスカイプや電子メールやチャットなどを使って提供することができます。詳しくは[弊社まで電子メールをお送りください](#)。

In addition, Code Architects offers migration services, so that you can let us migrate your VB6 application in your behalf. Read [here](#) for more information.

付け加えて言いますと、コードアーキテクト社はマイグレーションサービスを提供しておりますので、御社の VB6 アプリケーションを弊社にご依頼いただくことが可能です。詳しくは[こちら](#)をお読みください。

注意: 日本国内ではインフォーテック社がコードアーキテクト社に代わってマイグレーションサービスを提供しております。詳しくは[こちら](#)をお読みください。

- Why does VB Migration Partner emit more warnings than Microsoft Upgrade Wizard?／なぜ VB Migration Partner は Microsoft Upgrade Wizard に比べて多くの警告を出すのですか？

- In general VB Migration Partner emits more migration warnings than Upgrade Wizard (included in Visual Studio 2005 and 2008) or other VB6 migration tools from other vendors, a detail that might mistakenly bring you to the conclusion that VB Migration Partner is less precise than other tools on the market. The truth is just the opposite.

普通、VB Migration Partner は (Visual Studio 2005 と 2008 に含まれる) Upgrade Wizard や他社の VB6 変換ツールよりも多くの警告を出します。このことで VB Migration Partner が市場の他のツールよりも精度が悪いと誤解されるかもしれません。真実はまったく逆です。

In fact, just counting the number of warnings is simply misleading. The reason for this abundance of messages is that VB Migration Partner is able to spot many more potential errors than its competitors, which in turn allows you to anticipate many incorrect runtime behaviors that would bring to exceptions or just bogus results. To give an idea of how many potential problems Upgrade Wizard ignores, just browse [this page](#) and [this page](#).

実のところ、警告の数を数えることは誤解を招く恐れがあります。メッセージが多い理由は VB Migration

Partner が競合製品よりも多くのエラーとなる可能性のある場所を特定できるからです。逆に、これによってユーザーはエラーや不正な結果をもたらす多くの実行時の不正な動作を予測することができます。Upgrade Wizard が検出できない潜在的な問題の量に関する情報を見たいのなら、[こちら](#)と[こちら](#)をご覧ください。

Next, keep in mind that – unlike other VB6 migration tools – VB Migration Partner lets you decide which warnings should be emitted and which ones should be hidden – by means of the DisableMessage and DisableMessages pragmas – therefore you are always in control of the output from the migration process.

次に覚えておいていただきたいことは—他の VB6 変換ツールとは違って—VB Migration Partner を使えば、—DisableMessage プラグマと DisableMessage プラグマを使って—どの警告を表示させ、どの警告を表示させないかを定めることができるので、ユーザーは移行プロセスのアウトプットを常に管理できるということです。

Finally, consider that most warnings are related to code analysis features; these warnings highlight inefficiencies in the original VB6 code – methods that are never invoked, unreachable code, unused variables, ByRef parameters that can be safely transformed into ByVal parameters, etc. – and aren't really related to potential runtime errors. These warnings provide useful information to make the .NET code smaller and more efficient, however they can be safely ignored during the first phases of the migration. (Or you might just suppress them, as explained above.)

最後に、コード分析機能と関係のある警告の多くについて考察してみましょう。これらの警告は元の VB6 コードの非効率的な箇所—呼び出されないメソッドや実行されないコードや使用されない変数や ByVal 変数に安全に置き換えられる ByRef 変数など—に出ますが、潜在的なランタイムエラーとは実際のところ関連がありません。これらの警告は.NET のコードをより小さく、より効率的にするために役立つ情報を提供してくれますが、移行の初期段階では無視してもよいものです。(もしくは、上述したように削除することができます。)

Technical / 技術的な質問

- Which Visual Basic versions does VB Migration Partner convert from? / VB Migration Partner はどのバージョンの Visual Basic から変換できますか？
- VB Migration Partner converts applications written in VB6. If you have an application authored in VB5 or earlier version, you must convert it to VB6, ensure that it compiles and works correctly, and finally convert it to .NET using our tool.

VB Migration Partner doesn't convert other Visual Basic “dialects”, such as VBA and VBScript. Active Server Pages (ASP) applications aren't supported either, because they use VBScript.

We firmly believe that Active Server Pages applications should be redesigned and re-written from scratch as ASP.NET applications. Being VBScript a subset of VB6, converting ASP code would be a simple task; however, we don't offer this option because we don't want to encourage migration practices that we don't recommend.

- Which Visual Basic .NET versions does it convert to? / どのバージョンの Visual Basic .NET に変換できますか？

- VB Migration Partner can generate Visual Basic 2005, 2008 and 2010 projects.
- Do I need to run VB Migration Partner on a computer where VB6 is installed?／VB6 がインストールされた PC で VB Migration Partner を実行する必要がありますか？
- It isn't strictly necessary, but we recommend it for two reasons. First, the ability to edit and re-compile the VB6 application while fixing some recurring migration issues makes the migration process easier and speedier. Second, running VB Migration Partner on the same computer used to develop the VB6 application ensures that all file paths and registry entries are correct, that the right version of all type libraries is found, and so forth. In fact, VB Migration Partner might exhibit a few issues when converting a VB6 project that has been copied from another computer.

Notice, however, that during the conversion process VB Migration Partner does use components written in VB6, therefore it is required that the VB6 runtime be installed on the computer where you install VB Migration Partner. If this requirement isn't met, the setup process might fail.

- Do I need to run VB Migration Partner on a computer where Visual Studio .NET is installed?／Visual Studio .NET がインストールされた PC で VB Migration Partner を実行する必要がありますか？
- VB Migration Partner uses a few utilities that come with Microsoft Visual Studio, therefore we consider Visual Studio as a prerequisite for successfully carrying out the migration process. If you are perplexed about installing VB6 and Visual Studio .NET on the same computer, please read this [Knowledge Base article](#) from Microsoft.
- Can I test VB Migration Partner even if I haven't purchased Microsoft Visual Studio yet?／Microsoft Visual Studio をまだ購入していなかったとしても VB Migration Partner をテストすることはできますか？
- VB Migration Partner works well with both Microsoft Visual Studio and Microsoft Visual Basic Express, so you can test VB Migration Partner with the less powerful Visual Basic Express. Visual Basic Express is free and can be downloaded from [Microsoft web site](#).
- Which Windows versions does the tool support?／このツールはどのバージョンの Windows をサポートしていますか？
- VB Migration Partner run on all Windows versions on which .NET Framework 2.0 and Microsoft Visual Studio 2005, 2008 or 2010 can be installed (Both 32-bit and 64-bit versions are supported, where applicable). The same holds true for .NET applications that are produced by VB Migration Partner. Please check Microsoft web site for more information.
- Which VB6 controls does VB Migration Partner support?／VB Migration Partner はどの VB6 コントロールをサポートしていますか？
- All the 60+ controls that come in the Visual Basic 6 package, with only two exceptions: the OLE and the Repeater control. The so-called "windowless" controls (e.g. WLText and WLCommand) are also supported, as are a few controls and type libraries that aren't shipped with Visual Basic 6 but are often used by VB6 developers, such as the WebBrowser, ScriptControl, the Scripting library, and the VBScript Regex engine. VB Migration Partner even supports

some controls from third-party vendors, such as the Active ThreeD Plus suite from Sheridan (now Infragistics).

Unlike what other conversion tools do, *supporting a control* means that **all** major features of the control are correctly converted and behave as expected in the .NET application. For example, by stating that VB Migration Partner supports the PictureBox control we mean that it supports all its graphic methods, all drag-and-drop events (in both “classic” VB3-style and OLE flavors), data-binding, DDE, the ability to work as a container for other controls, and a lot more. No other conversion tool on the market supports all these features.

- Which VB6 features aren't supported? / サポートされない VB6 の機能はどれですか？
- VB Migration Partner doesn't support WebClass components, DHTML Page classes, DataReport controls, a few undocumented functions (VarPtr, StrPtr, and ObjPtr), and minor control properties such as DrawMode, ClipControls, and Palette.

UserDocument and PropertyPage classes aren't supported directly: they are converted to .NET UserControls, therefore developers can easily reuse them in a different context.

- Why these features aren't supported? / これらの機能はなぜサポートされないのですか？
- The actual reasons may vary. For example, WebClass and DHTML Page components have been used in relatively few VB6 commercial applications, thus we decided to focus our efforts on the features that have been used more extensively. In addition, we believe that server-side applications using WebClass components should be converted to ASP.NET applications, because only ASP.NET ensures that you take full advantage of the .NET Framework.

The DataReport hasn't been used in many business applications because most VB6 developers preferred more powerful 3rd-party reporting tools such as Seagate's Crystal Report, thus we decided that supporting DataReport classes wasn't essential. We don't support UserDocument and PropertyPage classes because they don't fit the .NET Framework architecture. Remaining unsupported features were simply impossible or way too difficult to implement in a robust way under .NET.

- Do converted .NET applications depend on COM and ActiveX components? / 変換された .NET アプリケーションは COM コンポーネントと ActiveX コンポーネントに依存しますか？
- The majority of controls used by converted .NET applications are fully managed controls and don't depend on any ActiveX control or COM-based technology. The only controls that depend on the original ActiveX control are: DataGrid, MSHFlexGrid, MS Char, MM Control, MSComm, INet, MAPISession, MAPIMessage, Winsock, and ScriptControl.
- Are converted applications “true” .NET Framework executables? / 変換されたアプリケーションは「本当に」.NET Framework 上で実行可能ですか？

- Yes, absolutely. VB Migration Partner generates native .NET applications that can be loaded inside Microsoft Visual Studio and modified or extended as needed. You can add new forms, add standard .NET controls to form migrated from VB6, use any class in the .NET Framework, make the application Internet-enable by adding support for Web Services, and so forth.
- Do converted .NET applications require a support library?／変換された.NET アプリケーションはサポートライブラリを必要としますか？
- All applications generated by VB Migration Partner require that the CodeArchitects.VBLibrary.dll file be deployed with the .NET executable. The support library is essential to support features that have no counterpart in .NET – DataEnvironment classes, for example – or that are implemented in a completely different way (graphic methods or data binding, just to name a few).

It should be noted that also applications converted with the Upgrade Wizard may require a companion library – more precisely, the Microsoft.VisualBasic.Compatibility.dll library – to support features such as control arrays, resource files, and a few controls that have no counterpart under .NET – namely, DriveListBox, DirListBox, and FileListBox. We just decided to improve on this concept with a support library that can account for virtually *all* the VB6 features that .NET doesn't support natively.

- Do converted .NET applications depend on Visual Basic 6 runtime?／変換された.NET アプリケーションは Visual Basic 6 のランタイムに依存しますか？
- Converted .NET projects have no dependency on Visual Basic 6 and its runtime.
- Can VB Migration Partner convert 3rd-party ActiveX controls?／VB Migration Partner はサードパーティ製の ActiveX コントロールを変換できますか？
- Yes, it can. We provide a command-line tool named **AxWrapperGen**, which you can use to create wrapper classes for ActiveX controls that VB Migration Partner doesn't support out of the box. You can fine-tune, modify, and extend these wrapper classes – if necessary – and control how properties of the original ActiveX controls are imported or translated. Finally, you just need to compile the wrapper classes into a DLL and drop it in a directory recognized by VB Migration Partner (VB Migration Partner recognizes DLLs located in its installation folder or in any directory mentioned in an AddLibraryPath pragma), so that the next time you run the tool it will recognize the additional controls. (An individual DLL can provide support for multiple ActiveX controls.)

We have prepared two whitepapers on this topic: [A smart approach to 3rs-party ActiveX control conversion](#) explains the basics of AxWrapperGen, whereas [ActiveX controls and wrapper classes](#) is a step-by-step tutorial that guides you through the conversion of a control.

- What about ActiveX controls authored in VB6?／VB6 で署名された ActiveX コントロールについてはどうですか？
- VB Migration Partner can convert UserControl controls authored in VB6. Nearly all features are supported, including advanced one such as the Extender object, Ambient properties, graphic methods, and more. VB Migration Partner even generates XML comments and Description attributes for properties and supports UserControl controls that work as data

sources (for example, a custom Data control). Our knowledge base includes [a few tips](#) for improving UserControl classes after converting them to .NET and we even provide a special VB Migration Partner extender to generate a more “polished” code for UserControl components.

- What is the best migration strategy if the VB6 applications uses *a lot* of ActiveX controls? / VB6 アプリケーションが多数の ActiveX コントロールを使用している場合、最も良い移行方法は何ですか？
- VB Migration Partner replaces unrecognized 3rd-party ActiveX controls with placeholder controls (i.e. picturebox controls with red background); for obvious reasons the tool can't correctly migrate references to the control's properties and methods. To complete the migration process, you can choose among three different strategies.
 1. If the control isn't used in each and every form of the application, it makes sense to migrate the application and then manually replace placeholder controls with the .NET control that more closely resemble the original ActiveX control. This strategy delivers a “native” .NET Framework application, but doesn't support the convert-test-fix methodology. (In other words, your manual fixes are lost if you re-convert the application.)
 2. You can use the **AxWrapperGen** tool to generate a .NET control that wraps the original ActiveX control. VB Migration Partner then uses the .NET control to replace all occurrences of the ActiveX control. This approach does support the convert-test-fix cycle, but delivers a .NET Framework application that still depends on ActiveX controls. Worse, a small number of ActiveX controls occasionally crashes when used via COM Interop, thus this approach might not be applicable to all controls in a real-world business application.
 3. in a more radical approach, you write a 100% managed control that has same members and behavior as the original ActiveX control. (This is the approach we adopted for nearly all the controls in our support library.) While wrapping hundreds of properties and methods of an ActiveX is a complex task, in practice most VB6 applications use only a small fraction of these members, thus this task is often easier than expected. (Of course, Code Architects can help in this effort.). In practice, you can modify the wrapper class generated by AxWrapperGen so that it derives from a real .NET control rather than the original ActiveX control.

Interestingly, mixing the first two strategies can often be a reasonable approach: you use the ActiveX wrapper during the migration and test process (to leverage the convert-test-fix cycle) and then replace the ActiveX control with a “native” .NET Framework control once all other issues have been solved.

- Can VB Migration Partner convert MTS/COM+ applications? / VB Migration Partner は MTS/COM+アプリケーションを変換することができますか？
- VB Migration Partner converts MTS/COM+ components written in VB6 into .NET classes that inherit from System.EnterpriseServices.ServicedComponents, with the proper Transition attribute. In addition, calls to the most important and widely used objects in the COMSVCSLib type library – namely ContextInfo,ObjectContext, SecurityProperty – are converted into calls to .NET methods. If a VB6 app uses only these objects, then the reference to COMSVCSLib typelib is dropped and the .NET project has no direct dependency on it; else, this typelib will be referenced also by the converted .NET application.

Additionally, VB Migration Partner recognizes the IObjectConstruct interface and generates the correct code that allows an MTS/COM+ component to retrieve the construction string passed to it by the COM+ infrastructure.

- Can VB Migration Partner convert NT service applications?／VB Migration Partner は NT サービスアプリケーションを変換することができますか？
- VB6 developers can run their applications as NT services by using the [NTSVC.OCX control](#). At this time we don't support this control, therefore the migration of Windows service applications requires some manual labor.
- Do converted .NET applications run under Microsoft Vista and Windows 7?／変換された.NET アプリケーションは Microsoft Vista や Windows 7 で動作しますか？
- Yes, with two caveats. First, a few .NET and Windows API methods might require you to disable User Access Control (UAC) features under Microsoft Vista, else the call would fail. (This action would be necessary also to run the original VB6 application under Vista.)

Second, if your VB6 application uses 3rd-party ActiveX controls, the converted .NET application uses a wrapper for that ActiveX control and we can't guarantee that the ActiveX control works well under Microsoft Vista or Windows 7. If in doubt, you should either ask the manufacturer of the ActiveX control or thoroughly test the original VB6 application under recent versions of the Windows operating system.

Please notice that VB Migration Partner uses wrappers for all the ActiveX controls that come with VB6, for example DataGrid and MS Chart. Microsoft [made a statement](#) that these controls work well under Microsoft Vista.

- Do I need administrative privileges to run VB Migration Partner?／VB Migration Partner を実行するのに管理者権限は必要ですか？
- VB Migration Partner requires access to areas of the registry and the file system – most notably, the System32 directory where most ActiveX controls are installed – therefore it is required that it runs under administrative privileges and – if running under Microsoft Vista – that User Access Control (UAC) be disabled.
- Can VB Migration Partner generate ASP.NET applications?／VB Migration Partner は ASP.NET アプリケーションを生成することができますか？
- No, because the ASP.NET programming model is too different from VB6 model for an automated conversion tool to achieve a significant success ratio. In our opinion, all existing tools that claim to offer such a feature don't generate .NET code that meets the quality standards that are required for real-world software, in terms of efficiency and maintainability.

Purchasing VB Migration Partner／購入に関する質問

- How much does VB Migration Partner cost?／VB Migration Partner はいくらですか？
 - The price of VB Migration Partner depends on a number of factors, including the total size of the applications to be migrated, the number of user licenses, and whether you are interested in any optional modules.

Please download our free [VB6 Bulk Analyzer](#) tool, run it over your VB6 source code, and [send us](#) the generated report. VB6 Bulk Analyzer can process all the files in one or more directory trees, therefore you

can create a single report for all the VB6 projects that make up your application. In return you'll receive a price quotation, a summary of all the challenges you're going to face in your migration process, which ones can be easily solved by VB Migration Partner, and which ones might require some additional work.

- What are the terms of the end user license agreement (EULA)? / エンドユーザー使用許諾契約条項はどうなっていますか？
 - Not counting the usual legal stuff, by purchasing our tool you basically agree on the following terms: You can't convert someone else's VB6 projects, unless they also own a copy of VB Migration Partner. To clarify: VB Migration Partner consists of two distinct pieces, the actual converter and the support library. If you own an application that has been converted from VB6 using our tool (and therefore requires the support library) then you must purchase a VB Migration Partner's license, even if the actual conversion job was done by an external consultant or a 3rd party company. This restriction holds true regardless of whether you have converted the VB6 project for internal use or if you are reselling it to end users.

- Is the support library's source code available for purchase? / サポートライブラリのソースコードを購入することはできますか？
 - The support library is provided as compiled, optimized VB.NET code and there's no technical reason for purchasing the support library's source code. From a conceptual point of view, the support library has the same role of the .NET Framework library: you can extend it by inheriting from its classes or by adding new classes, but you never need to modify its source code. Moreover, even a minor change in the library's code might introduce subtle bugs and we wouldn't be able to support any .NET application that uses a modified version of the support library.

However, we are willing to license the support library's source code, if your company policy requires that you can access the source code of any 3rd party software. For more details and price, please [contact us](#).

- Is it possible to get a Trial? / トライアル版を入手することはできますか？
 - Yes, you can have a fully-working, time-limited Trial Edition of VB Migration Partner that allows you to test it against your actual code. [Contact us](#) to receive the personalized link to download it.

- What do I have to do to get your free advice for a successful migration? / うまく移行するためにどのような無償アドバイスを受けるべきでしょうか？

- To get our [free advice](#) and learn more about your VB6 applications, run [VB6 Analyzer](#) on your source code and [send us](#) the generated text file. You will receive a detailed report, prepared by a team of VB6 and .NET skilled developers led by VB guru Francesco Balena, about your VB6 applications and how VB Migration Partner can help you to quickly and effectively migrate it to .NET.

- Is technical support included in the license price? / 技術サポートはライセンス価格に含まれていますか？

- VB Migration Partner's license comes with a 6 month subscription period, during which you can submit any technical question related to our software *and* you will be able to download any new 1.xx version of VB Migration Partner, including any other ancillary component we have uploaded to our web site in the meantime.

At the end of the subscription period you will still be able to download any new version of the 1.xx support library, so that you won't have to fix your migrated code if a future version of Windows or the .NET Framework is partially incompatible with your migrated code.

- May I extend the subscription period? / ライセンスの有効期間を延長することはできますか？

- Yes, the subscription period can be further extended. Please [contact us](#) for more details.

- We provide migration services to 3rd party companies. Can we later transfer the license to our customers? / 弊社ではサードパーティベンダーに対する移行サービスを提供していますが、将来、弊社の顧客にライセンスを譲渡することはできますか？

- If you are an independent consultant or a company that provides migration services, you need to purchase a VB Migration Partner license to do the migration on your computer(s). However, once the migration is over you can freely transfer your license to your customer. However, it is essential that you inform us by email or fax, so that we can deactivate your license and activate theirs.

- What kind of guarantee do you offer that VB Migration Partner will correctly convert my VB6 application to .NET? / VB Migration Partner が VB6 アプリケーションを .NET へ正常に移行することについて御社ではどのように保証していますか？

- Unfortunately, no code conversion software – regardless of its source and target language – can be proved to work flawlessly in *all* circumstances. Just think that all competing VB6-to-.NET tools on the market don't even attempt to convert features that are fully supported by VB Migration Partner, for example graphic statements, drag-and-drop, and DDE. It is true, however, that your VB6 application may include portions that won't work well after the migration to .NET, for a variety of reasons. For example, your project might use one

or more ActiveX controls that don't work when hosted on a .NET form, or it might rely on one of the few VB6 features that VB Migration Partner doesn't support, such as UserDocument or HTML Pages.

The simplest and fastest way to understand whether VB Migration Partner supports a given feature or control is browsing and searching our massive online documentation. We put together a detailed manual and a huge collection of Knowledge Base articles that describe what VB Migration Partner support, and what it doesn't. This is much more than any other competing product currently offers.

But there is another, simpler way to check whether VB Migration Partner can fulfill your needs: just ask for a Trial Edition and see it for yourself. VB Migration Partner Trial Edition has no limitations in the size of VB6 projects it can convert and includes virtually all the features of the Full edition. The only limitation is that the Trial Edition has an expiration date, yet you have enough time to test it against the most challenging portions of your code. Best of all, Trial Edition users have full access to tech support, thus you can also test the quality of the team of developers behind VB Migration Partner.

- How many years will VB Migration Partner library be supported? / VB Migration Partner のライブラリは何年間サポートしてもらえますか？

- We plan to support it for at least 8 years, or until Microsoft will support VB.NET or Windows Forms (depending on which event will happen first). Of course, if for any reason Microsoft will not support anymore VB.NET or Windows Forms, the migrated application will not be supported; so the real problem will not be our support library.

At any rate, Microsoft always supports a technology for at least 10 years after it has been declared obsolete. As of 2010 VB.NET and Windows Forms are still supported, therefore we don't anticipate any problem in this area.

- There will never be a charge for the support library? / サポートライブラリに対する課金はまったく発生しないのですか？

- VB Migration Partner's support library is royalty free and it will always be, period.

In addition to being royalty-free, registered VB Migration Partner's user have the lifetime right to download any future 1.xx release of the library. Such future releases will include all bug fixes, performance optimizations, and other minor features.