

# Re: Neuer VB Compiler

---

*Source:* <http://www.tech-archive.net/Archive/German/VB/microsoft.public.de.vb/2006-06/msg00580.html>

---

- *From:* Ulrich Korndoerfer <[ulrich\\_wants\\_nospam@xxxxxxxxxxxxx](mailto:ulrich_wants_nospam@xxxxxxxxxxxxx)>
  - *Date:* Thu, 29 Jun 2006 02:46:26 +0200
- 

Hallo Harald!

Harald M. Genauck schrieb:

Hallo Ulrich,

Das Projekt ist noch nicht mal fertig :-). Soll nächstes Jahr in einer ersten Version öffentlich verfügbar sein.

Soll... hmmm... weiß jemand, wie stabil Sun im Allgemeinen mit derartigen Ankündigungen ist?

Keine Ahnung. Aber es geht hierbei wohl auch um die künftige Positionierung von Java gegenüber .NET, und Sun möchte wohl zum einen das Multilanguagefeature von .NET nun auch in Java ernsthaft und offiziell unterstützen und zum anderen Entwickler zum Umstieg von .NET-Tools zu Java-Tools bewegen. Das läßt hoffen, daß dabei auch die Wünsche umstiegs- oder einstiegswilliger Entwickler verstärkt berücksichtigt werden.

Wieso sollte man allein aus der Tatsache, dass Sun Entwickler zum Umsteigen bewegen will, schließen können, dass sie auf Wünsche eingehen werden, und das auch noch "verstärkt"? Schließlich will auch Microsoft Entwickler zum Umsteigen bewegen. Da brauchts doch irgendwie noch ein wenig mehr an Fakten, um einen solchen Schluss ziehen zu können, oder?

Um jemanden zum Umstieg zu bewegen, muß ich ihm etwas anbieten. Das, was MS anbietet, reicht oder gefällt vielen nicht. Wenn Sun hier dieses Fehlende anbietet, wäre das doch ein Argument zum Umstieg auf

## Re: Neuer VB Compiler

Sun, oder?

Es gab eine erste Demo der geplanten Fähigkeiten. Dabei wurde unter anderem in der Netbeans-IDE eine neue (minimale) VB-App erstellt: zwei Buttons, ein Textfeld und Eventhandlercode. Die Erstellung lief so wie unter VB gewohnt: neues Form anlegen, aus dem Toolkit die Controls auf die Form ziehen, per Klick auf die Controls die Eventhandlergerüste anlegen und dann im Codeeditor mit Code füllen. Das Toolkit soll alle VB-Standardcontrols zur Verfügung stellen (incl. zB FileListBox etc).

Klingt ja vielversprechend... Hm... nur die Standard-Controls? Und was ist mit den Common- und den Dialog-Controls? Und was ist mit 3rd-Party-OCXen?

Geht alles, so ist es zumindest annonciert: Zugriff auf Java-Framework per VB Code soll möglich sein, damit auch Zugriff auf die Möglichkeiten, die Java bietet, damit Zugriff auf COM-Server jeglicher Art und Standard-Dlls. Dann ist natürlich Anpassungsaufwand fällig.

...

Nein. UI-Programmierung wird 1:1 übernommen. Da, wo VB intern COM verwendet, wird die gleiche Funktionalität über die Java-Runtime zur Verfügung gestellt. zB wenn man mit New eine VB-Instanz einer VB-Klasse erstellt, verwendet VB dafür intern COM, VBforJava intern die Java-Runtime-Methoden (wobei die VB-Syntax aber bestehen bleibt). Das soll auch für Dlls funktionieren, wenn diese als Source vorliegen. Es wird dann keine Windows-COM-Dll erzeugt, sondern entsprechende kompilierte Java-Klassen. Diese sind aber wie gewohnt unter VB anzusprechen (mit New etc). Die interne Umsetzung, die ja COM nicht verwendet, bleibt dabei verborgen.

Und was ist mit 3rd-Party-COM-DLLs, ganz zu schweigen von Standard-DLLs?

Siehe oben.

Wenn man also eine VB-Applikation hat, die zudem einen Teil ihrer Funktionalität in VB-Dlls ausgelagert hat, und für alles die Sourcen vorliegen, soll diese App inklusive der "Dlls" 1:1, ohne ein Bit am Sourcecode ändern zu müssen (inklusive der Forms), unter VBforJava lauffähig sein. Und zwar plattformunabhängig. Seit wann kann .NET das?

## Re: Neuer VB Compiler

..NET kann das nicht, klar. Aber ist das wirklich so entscheidend? Denn dieser Idealfall einer Pur-VB-Code-Anwendung ohne jeden Griff unter die Haube ist meiner Einschätzung nach zumindest bei professionell( geschrieben)en Anwendungen eher eine Rarität. Zumindest in meinem Bestand gibt es keine.

Direkte API-Aufrufe und Aufrufe von Klassen in binären COM-Dlls bleiben natürlich plattformabhängig und der aufrufende Code muß auch angepaßt werden.

Eben...

Bezüglich des Engagements der VB-Community bei der Einbringung von Wünschen sehe ich die Gefahr, dass womöglich unter dem Wunsch auch hier "endlich mal alte Zöpfe abzuschneiden" und einer möglicherweise vorhandenen Unerfahrenheit auf Seiten Suns mit tatsächlichen VB-Gegebenheiten die versprochene Kompatibilität doch mehr oder weniger leiden könnte...

Eines der Ziele laut Sun ist es, VB-Entwicklern zu ermöglichen, ihre VB\*classic\* Kenntnisse verwenden zu können, um damit unter einer Java-Runtime Programme zu entwickeln. Es wird also nicht erwartet, daß diese sich erst neue Kenntnis aneignen müssen. Das geht natürlich nur, wenn VB classic 1:1 unterstützt wird. Erweiterungen sind trotzdem möglich und auch angedacht. Diese sollen aber \*wahlfrei\* sein. Also auf Wunsch strikt VB classic oder mit Erweiterungen. Und jetzt sag nicht, daß das nicht geht. Das geht sehr wohl.

Sicher geht das.

Und mit dem "endlich mal alte Zöpfe abzuschneiden" zeigst Du deutlich, daß Du immer noch nicht kapiert hast, worum es vielen VB-Entwicklern inklusive meiner Wenigkeit geht.

## Re: Neuer VB Compiler

Da hast Du mich aber missverstanden. Das "alte Zöpfe abschneiden" habe ich lediglich als Gefahr gesehen, dass eine zu eingehende Berücksichtigung von Entwicklerwünschen Sun dazu verleiten könnte, letztlich doch auch die angestrebte 1:1-Kompatibilität platzen zu lassen.

Wie gesagt, angedacht ist es, VB classic 1:1 anzubieten und \*wahlfrei\* Erweiterungen. Also auf Wunsch kann 1:1 gearbeitet werden, auf Wunsch aber auch unter Einbeziehungen etwaiger Erweiterungen.

Ich habe nichts dagegen, alte Zöpfe abzuschneiden. Ich hätte auch nichts dagegen, wenn .NET ohne VB.NET ausgeliefert werden würde. Die so geschätzten RAD-Features sind auch mit C# möglich. Was mich dagegen aufregt, ist, daß man sich nicht die Mühe gemacht hat, für den \*Übergang\* VB classic unter .NET zu ermöglichen.

Vermutlich hätte Microsoft das besser machen können. Das denke ich auch, aber ich rege mich deswegen nicht auf – das ist wohl der wesentliche Unterschied in unseren Standpunkten. Und ich vermute, dass man sich Microsoft heute im Rückblick darauf, dass erst jetzt mit der 2005er Version ernsthaft das Migrations-"Marketing" angegangen worden ist, auch denkt, dass man vielleicht doch eine Zwischenlösung hätte anbieten sollen.

Hurra! Der Beginn einer Einsicht? ;-)

Was mich dagegen aber eher "aufregt", ist, dass nach so verhältnismäßig langer Zeit immer noch, obwohl faktisch wohl nichts mehr daran zu rütteln ist, immer noch Energie fürs "Aufregen" (um nicht zu sagen, "Jammern") verschwendet wird.

Ich jammer nun mal gerne :-)

Im Ernst, ich komme momentan noch gut mit VB Classic zurecht, ein etwaiger Umstieg auf was auch immer steht erst in zwei, drei Jahren im Raum. Bis dahin habe ich also noch genügend Zeit, mir das dann beste Tool für den Umstieg auszusuchen. Das könnte hier VB for Java als Einstieg in dann Programmierung gegen das Java-Framework sein.

– Und all diejenigen, deren Vertrauen gegenüber MS gewaltig gelitten hat (wg. des gewaltsamen Abwürgens von VB) und deshalb zwar sehen, daß langfristig eine Umstellung nötig ist, dies aber ungern wieder mit MS-Tools tun möchten, haben nun eine

## Re: Neuer VB Compiler

ernstzunehmende Alternative.

Setzt man hier nun auf Sun, begibt man sich u.U. vom Regen in die Traufe. Sich aus Frust über die Abhängigkeit von Microsoft nun in die Abhängigkeit von Sun begeben zu wollen, halte ich für wenig rational.

Quark. Sobald Du Software entwickelst und dazu notwendigerweise irgendwelche Tools verwendest, bist du schon abhängig. Das ist rational, sich darüber im klaren zu sein.

Wenn Du Dir darüber im klaren bist – schön. Viele Stimmen in den Diskussionen bisher lassen aber eher vermuten, dass es bei weitem nicht allen so klar ist, bzw. gewesen war, als sie sich auf VB–Classic eingelassen haben.

Na, das man sich abhängig macht, sollte einem schon klar gewesen sein. Aber man hat sich halt nicht ohne vermeintlich gute Gründe auf VB eingelassen: zB war MS bis vor einiger Zeit bekannt dafür (und das war sicher auch einer der Gründe für den MS Erfolg), Kompatibilität bis zum Erbrechen zu bewahren.

Es gibt zB für das Betriebssystem einen schönen Blog eines MS–Angestellten, in dem des öfteren krumme Lösungen auf den unbedingten Willen zur Aufrechterhaltung der Kompatibilität zurückgeführt wurden.

Dies ging sogar soweit, das für APIs fehlerhaft anwendende, aber verbreitete Apps von Drittanbietern das Betriebssystem dies überwacht hat, und, wenn die entsprechende API von der fehlerhaften App aufgerufen wurde, dieses sich dann so verhalten hat, daß wieder alles gut wurde.

Da gabs ein Kompatibilitätslabor, die nichts anderes zu tun hatten, als ständig alle möglichen Apps von Drittanbietern zu untersuchen, ob sie unter der nächsten OS–Version auch laufen werden. Die GPFs wurden notfalls die Apps disassembliert, um die Ursache festzustellen. War diese ein fehlerhafter API–Aufruf, wurde notfalls das API–Verhalten korrigiert (speziell angepaßt)!

Es gab allerhand kopfschüttelnde Kommentare zu diesem Eintrag im Blog, nach dem Motto: warum um Himmels willen hat man sich nicht direkt an den Hersteller gewendet und diesem einen dezenten Wink gegeben?

Wie Du sicher auch weist, gibt es sogar API–Funktionen, die nie öffentlich gemacht werden sollten, aber durch irgendwelche Zufälle entdeckt und fleißig verwendet wurden. Die hat MS dann beibehalten, um nur ja keine Kompatibilitätsprobleme zu bekommen. Man hat also sogar die Kompatibilität für undokumentierte APIs gehalten! Das waren noch Zeiten.

## Re: Neuer VB Compiler

Rational ist es auch, sich die Tools zu suchen, die einem am besten behagen und produktiv sind (zB möglichst wenig Aufwand für eine initiale Codeumstellung benötigen). Und rational ist es, dabei auf ein langfristig unterstütztes Tool zu setzen und nicht auf Tools, die morgen schon vom Markt verschwinden können oder deren Entwickler nicht die Kapazitäten haben, ihr Tool laufend zB gegen die sich ständig wandelnden Plattformen upzudaten. Wer das nun liefert, ist mir, mit Verlaub, sch\*egal.

Und bei Sun bist Du also rational der Meinung, dass die die Ankündigung so wie eben angekündigt wahr machen werden, dass es langfristig unterstützt werden wird, dass es nicht morgen (... d.h. übermorgen, da es ja erst "morgen" erscheinen soll...) schon vom Markt verschwindet, dass es laufend gegen die sich wandelnden Plattformen upgedatet werden wird?

VB classic ändert sich nicht mehr. Da gibts nicht upzudaten. Der Rest ist Java-Runtime, und die updatet Sun schon ziemlich lange. Ich sehe keinen Grund, warum sie es auch nicht weiter tun sollten. Für JavaSE 6 ist zB Vista Unterstützung vorgesehen.

Die Frage ist auch, wie stabil dann irgendwann im nächsten Jahr die erste Version sein wird

Würde genauso stabil wie .NET1 reichen?

Wenn die das ohne die 2 Beta-Jahre, die .NET 1 vor der Veröffentlichung 2002 hinter sich hatte, hinbekommen sollten, wäre das bewunderswert – also Antwort: ja.

Wahrscheinlich stabiler und in kürzerer Zeit als zwei Jahre. Die Basis (das Framework und sämtliche Tools) ist ja vorhanden und erprobt. Es kommt "nur" ein neuer Compiler dazu.

(so es denn nicht bloß bei der Ankündigung und

Hoffnung darauf bleibt), oder wie lange es dann noch dauern wird, bis eine tatsächlich taugliche, stabile Version verfügbar sein wird.

## Re: Neuer VB Compiler

So wie .NET2 nach 5 Jahren?

2005 – 2002 macht bei mir 3 Jahre... aber sei's drum – Antwort: ja.

Macht bei mir 4 Jahre, aber egal ;-)

Das hieße aber auch, dass dieses VBforJava so etwa erst 2010 wirklich produktiv nutzbar wäre.

Nee. Ich gehe davon aus, daß es in spätestens 2 Jahren produktiv nutzbar sit.

Sollte Microsoft die Ankündigung wahr machen, alle 18 Monate eine neue VS-Generation mit neuen .NET-Essentials auf den Weg zu bringen, dürften dann aber Welten hinsichtlich der Produktivität eines stabilen VB-Classic-Nachbaus und dem dann aktuellen .NET liegen...

Wer vergleicht jetzt mit .Net? Zumindest muß man hier zwischen .Net und dem Java-Framework vergleichen. VB classic hat damit nichts zu tun. VBforJava soll erst mal ermöglichen, VB classic Code zu verwenden. Damit habe ich schon mal die Möglichkeit, alten Code 1:1 auf einer unterstützten Plattform laufen zu lassen. Neuentwicklungen dann unter Java, mit allen Möglichkeiten, die das Framework bietet.

Als direkte, echte Alternative zu VB.NET sehe ich das aber nicht – jegliche vergleichende Diskussion zu VB.NET halte ich daher auch für überflüssig.

Es soll ja auch nicht mit VB.NET verglichen werden, da es nicht vergleichbar ist. Es ist ein anderer Ansatz.

Warum argumentierst Du dann ständig mit .NET vergleichend?

Ich machs halt so wie Du ;-)

## Re: Neuer VB Compiler

Grundsätzlich finde ich es aber interessant, die Idee, unter verschiedenen Sprachen gegen ein umfassendes Framework zu programmieren, aufzugreifen – hier nun unter einem Basic gegen das Java-Framework, also eine interessante Option einer zusätzlichen Plattform, die man dann ebenfalls mit grundlegenden Basic-Skills bedienen kann.

Ja. Zuerst hat Sun mit einer modernen, pointerlosen Sprache und einem mächtigen, plattformunabhängigem Framework Maßstäbe gesetzt. Dann hat MS nachgezogen und die Fähigkeit, das Framework unter verschiedenen Sprachen zu bedienen, zusätzlich (schon by design) vorgesehen. Dagegen ist auch nichts einzuwenden, leider fiel die Plattformunabhängigkeit dabei weg (und jetzt bitte nicht Mono erwähnen, das ist nicht von MS). Und nun zieht Sun seinerseits nach und schwenkt auf die multiple languages – one framework Geschichte ein (offiziell, mit Sun Unterstützung). Damit hätte Sun wieder die Nase vorn.

Seit wann bedeutet ein Einschwenken/Gleichziehen/Nachahmen "Nase vorn"? Das kann ich so jetzt nicht ganz nachvollziehen. Wegen der Plattformunabhängigkeit?

Genau deswegen.

Nun ja... warum werden dann nicht die

verbreitetsten (und populärsten) plattformübergreifend verfügbaren Anwendungen bisher schon noch nicht in Java geschrieben? Mozilla/Firefox, Opera, Open Office, Apache, PHP-Runtime, Perl-Runtime u.v.v.v.v.v.m.? (Nein, ich will jetzt sicher keine pro-/kontra Java-Diskussion anfangen...)

Da mußt Du schon deren Ersteller selbst fragen. Ebenso könntest Du aber Andere Fragen, warum sie ausgerechnet Java für ihr plattformübergreifendes, weit verbreitetes Programm verwendet haben. Webserver (unter Java sind das eher ApplicationServer): zb TomCat, WebLogic, etc. Für Perl, Python u.v.v.v.v.v.m. gibt es Java-Ports.

Ich weiß wirklich nicht, was Du mir damit sagen willst. Das Java ungeeignet ist? Dann aber bitte Argumente.

--

Ulrich Korndoerfer

VB tips, helpers, solutions -> <http://www.proSource.de/Downloads/>

---

Plea for a bright future for VB classic.

Sign the petition to Microsoft: -> <http://classicvb.org/petition/>

Re: Neuer VB Compiler

