

JBuilder 4.0 – Мощна среда за разработка

© Светлин Наков, 2000
<http://www.nakov.com>

Интегрираната среда за разработка на приложения JBuilder е едно от най-съвременните и популярни средства за програмиране на JAVA. С удобния си редактор, интегрирания дебъгер и всички останали качества JBuilder е любимият продукт на много JAVA програмисти. Възможностите му за бързо визуално програмиране с използване на готови компоненти и лесен достъп до бази данни го поставят на водещо място и сред средствата за бърза разработка на приложения (Rapid Application Development). Чрез добрата поддръжка на JSP, EJB и всички останали съвременни JAVA технологии JBuilder позволява лесна разработка на мощни и сложни бизнес приложения, което го прави предпочитан продукт и при големите корпоративни разработки. Благодарение на това, че е базиран изцяло на JAVA, той работи на различни платформи – Windows, Linux, Solaris и т.н.

JBuilder обикновено се разпространява в три различни варианта:

- JBuilder Foundation – напълно безплатна версия, която включва само интегрираната среда за разработка
- JBuilder Professional – включва и средствата за визуална разработка на приложения, компоненти за достъп до бази от данни, средства за INTERNET и Web разработка
- JBuilder Enterprise – включва и пълен набор от средства за разработка на големи корпоративни приложения – разширена поддръжка на Web разработки с JSP, разширена поддръжка на бази от данни, пълна поддръжка на EJB и CORBA за разработка на разпределени системи, интеграция с различни Application Server-и, средства за работа в екип и още др. допълнителни възможности

В настоящата статия ще разгледаме в детайли последната версия на продукта – JBuilder 4.0 Enterprise и по-специално новостите, които тя предлага, без да се спираме на старите възможности, които са добре известни за повечето читатели.

JBuilder 4.0 поддържа напълно J2SE (Java 2 Standard Edition) и J2EE (Java 2 Enterprise Edition – <http://java.sun.com/j2ee/>) технологиите на Sun. Нормално работи с JDK 1.3, но позволява разработка и за по-стари версии на JDK – JDK 1.1, JDK 1.2, JDK 1.2.2. Както знаем JDK 1.3 е последната версия на JAVA 2 технологията, която се отличава с множество разширения и увеличено бързодействие, особено в приложенията със Swing-базиран user interface. За подобряване на производителността на разработваните приложения се поддържа и новата HotSpot виртуална машина, която ускорява изпълнението на JAVA програмите до няколко пъти пъти спрямо остарелия вече JIT компилатор.

Новостите в интегрираната среда за разработка разширяват възможностите на редактора, като го правят още по-удобен. Средата на JBuilder 4.0 е базирана изцяло на JAVA с XML, зарежда се по-бавно, но работи значително по-бързо от тази на JBuilder 3.5. Добавени са нови wizard-и за автоматизиране и ускоряване на работата.

Разширена е поддръжката на JAVA архиви (JAR, ZIP, WAR). Приготвянето на програмите за разпространение (deployment) е подобро с новия Archive Builder. За свикналите с други среди за разработка и други клавиши за основните операции е предвиден специален визуален редактор за клавишните комбинации (visual keypad editor), чрез който можете да си настроите клавиатурата по ваше желание. Проектите се записват вече в специален XML формат (.JPX). Новата среда поддържа и вграден браузър за HTML4 с CSS2 и XML. Редакторът е разширен да поддържа Syntax Highlighting, CodeInsight и StructureInsight функциите не само за JAVA програми, но и за широко използваните в програмирането HTML, CSS, WML, JSP, XML, DTD, XSL и IDL файлове. Едно много важно подобрене е направено в help-системата. Клавишът F1 е вече context-sensitive, т.е. при натискането му по време на работа се показва документацията за текущия метод или поле на текущия клас без да е необходимо да се прави търсене на ръка, както беше в предишните версии. Това всъщност не е нещо супер ново, защото в Delphi, Visual C++, Kawa и дори в Turbo Pascal за DOS тази функция винаги е работела. Не е ясно защо в JBuilder тя е добавена едва сега. Сериозно е подобрена функцията за автоматично дописване на започнат код (CodeInsight), която най-сетне започва да прилича на това, което програмистът би искал – това, което го има в Delphi, Visual C++ и Visual Basic от няколко години насам. И въпреки това неудобството, че автоматичното дописване на код спира да работи при първата синтактична грешка в програмата и не започва да работи отново докато тя не се отстрани си остава (докато в Delphi например такъв проблем никога не е съществувал).

Трябва се отбележи, че разработката на INTERNET и Web приложения с JBuilder 4.0 е наистина силно улеснена. Предоставя се пълна поддръжка на Servlets 2.2 и Java Server Pages (JSP) 1.1 стандартите, като създадените Web-приложения могат да бъдат изпълнявани на вградения в JBuilder сървър (Apache Tomcat 3.1), както и на външен Servlet/JSP сървър. Редакторът вече поддържа и JavaScript за улесняване на DHTML разработчиците. Много съществено подобрене е възможността да се дебъгват сървлети и JSP с интегрирания в средата дебъгер, като се използват пълните му възможности. Специални wizard-и и пакети от компоненти (ServletExpress и JSPExpress) ускоряват и автоматизират създаването на JSP приложения. Поддържат се стандартно сървлети за извличане на информация от бази данни в XML формат. Новите компоненти InternetBeans Express, позволяват лесно създаване на Internet-ориентирани приложения с достъп до бази от данни. Има и възможност за връзка на Web-базирани front-end приложения с EJB.

Новият JBuilder 4.0 има и функции за поддръжка на WAP, което го отваря като средство за разработка на Web-приложения за HandHeld мобилни устройства. Специални wizard-и създават сървлети, които обслужват WAP заявките. Разбира се, има и редактор за WML.

Като става дума за мобилни устройства, трябва да споменем, че JBuilder има и добавка за разработка на приложения за HandHeld устройства, с която могат да се създават визуално приложения за Palm OS, които да се изпълняват от специалната JAVA виртуална машина за Palm. Повече информация можете да намерите от: <http://www.borland.com/jbuilder/hhe/>.

Относно управлението на бази от данни, трябва да споменем, че са постигнати някои подобрения, но понеже базите данни се управляваха изключително лесно и удобно още в по-старите версии на продукта, не може да се забележи значителна

промяна. Новите неща, които JBuilder 4.0 ни дава са вградена поддръжка на Connection pool-ове, възможности за връзка с отдалечени бази данни през JNDI, JSQL за интерактивно създаване на SQL-заявки (таблиците се свързват с drag-n-drop), JDBC Explorer за визуално управление на data source-ове и таблици и още няколко полезни инструмента. Многобройните компоненти за достъп до бази данни и визуализация на данните позволяват бърза разработка на DB-приложения. Заедно с JBuilder се предоставя и лиценз за разработка и пълна интеграция с мощната обектно-релационна база данни Inprise JDataStore, която е силно разширена и подобрена. Новата версия на JDataStore поддържа кодиране на данните, автентикация на потребителите, синхронизация с JDBC, разширения на SQL интерпретатора и още много нови функции, които я правят удобна за по-малкия бизнес, където мощта и надеждността на ORACLE не са необходими.

Едно от най-големите постижения на JBuilder 4.0 е насочено към поддръжката на Enterprise Java Beans (EJB). Разработката, дебъгването и внедряването на EJB-та винаги е било тежък проблем, защото до скоро нямаше подходящи инструменти за тази цел. С новата версия на JBuilder работата с EJB е вече доста по-лека работа. EJB могат да се създават визуално посредством специални Wizard-и за създаването на Entity Bean-ове, Session Bean-ове и техните “home” и “remote” интерфейси. Инструментът Entity Bean Modeler позволява автоматично пренасяне на релационната структура на базата данни в обектен вид чрез визуално пренасяне на таблици и полета, като генерира Entity Bean-овете, създава “home” и “remote” интерфейсите им и Primary Key класовете. За създаването на deployment-дескрипторите JBuilder 4.0 предлага специален XML редактор. При разработката с EJB постоянно се налага да се създават EJB- архиви, които да се deploy-ват на Application-сървър. В JBuilder 4.0 този процес вече е автоматизиран. Поддържа се автоматично създаване на EJB-архиви по време на компилирането на проекта и освен това deploy-ването се извършва също автоматично. В настоящата версия интеграция има само с Inprise Application Server и WebLogic Application Server, но вероятно ще се поддържат и други EJB контейнери. Като добавка към тези възможности има и възможност и за тестване и дебъгване на EJB локално и отдалечено с пълната мощ на дебъгера. За CORBA разработчиците също са предвидени специални wizard-и за създаване на CORBA компоненти, които могат автоматично да се deploy-ват на ORB сървър. Предвидени са начини за тестване и дебъгване на CORBA обекти. Има интеграция с ORB сървърите VisiBroker и OrbixWeb. Редакторът поддържа IDL за редактиране на CORBA интерфейсните дефиниции.

Нещо, което липсваше на предходните версии и което е неразделна част от процеса на разработка на софтуер най-сетне е направено в JBuilder 4.0. Става дума за възможността за работа в екип. Досега обикновено за работа в екип се използваша външни програми за контрол на версиите и синхронизация между разработчиците. В новият JBuilder вече има вградена в средата система за контрол на версиите, която поддържа два режима на работа – с локален архив на данните и с външен архив на данните. При използване на локален архив всички версии на всеки от файловете, който принадлежи на разработвания проект се съхраняват локално на хард-диска на разработчика. Дава се възможност за сравнение на версиите, визуално показване на разликите, връщане до стара версия, копиране на код от стара версия и т.н. При разработка на големи проекти, обаче, локалният архив не е удобен поради редица

причини. За това се използват специални програми за контрол на версиите (VCS - version control systems), които представляват сървъри, до които има достъп целият екип, който разработва проекта. Работата е така организирана, че програмистите могат да вземат файлове от проекта за да работят върху тях, да ги променят и след това да ги връщат обратно, при което сървърът автоматично запазва старите версии. В зависимост от настройката на сървъра той или не позволява върху един и същ файл да работят няколко души за да не настъпи бъркотия, или когато това се случи, се предоставя възможност различните версии на един и същ файл, получени от различни разработчици внимателно да се слоят в едно цяло, като това става или автоматично или с намесата на разработчика. Дава се добра възможност на хора, които физически се намират на големи разстояния да работят заедно в екип върху един и същ проект. JBuilder 4.0 се интегрира много добре с широко разпространената програма за контрол на версиите "CVS", а също и със системата "Borland Team Development". Допълнително може да се инсталира и поддръжка на популярният продукт "Visual SourceSafe", както и на други VCS системи. Повечето такива системи поддържат история на файловете, която описва промените на всеки файл, като за всяка от тях се пази кога и от кого е направена. Има възможност за добавяне, взимане, промяна, връщане и изтриване на файлове в архива, а също за разрешаване на конфликти, ограничаване на достъпа и т.н. По този начин се синхронизира работата при разработка в екип.

Относно средствата за дебъгване (отстраняване на грешки чрез следене на работата на програмата) трябва да споменем, че новите неща по-скоро разширяват старите функции и създават допълнителни удобства за програмиста, отколкото внасят нещо революционно ново. Понеже дебъгерът на JBuilder още от първите му версии си беше много добър, не е чудно, че и в новата версия той си е все такъв. Хубавите неща са че можем да дебъгваме JSP скриптове с пълните възможности на дебъгера, можем също да дебъгваме EJB, можем да отваряме няколко дебъг сесии за един и същ проект, можем да дебъгваме програми на отдалечени сървъри, можем да дебъгваме няколко нишки на програмата паралелно и всичко това без много усилия. Разбира се, можем да използваме и всичките ни стари познати техники и инструменти за дебъг – условни и безусловни breakpoint-и, watches, постъпково изпълнение, изчисляване на изрази, наблюдаване на променливите и стека по време на изпълнение на програмата, автоматично прекъсване при не уловен exception и още много други.

До момента разгледахме главно добрите страни на JBuilder 4.0. Нека сега да хвърлим поглед и върху и проблемите, които възникват с продукта по време на работа. По време на тестовете, които проведохме с JBuilder 4.0 установихме, че при дебъгване командата "Resume Program" не работи след използване на командата "Step Into" или "Step Over". Това прави почти невъзможно по-задълбоченото дебъгване. С помощта на мои колеги успяхме да се преборим с проблема, който се оказва, че произлиза не от JBuilder, а от използването на Hot-Spot виртуалната машина, която компилира кода по време на изпълнение и премахва breakpoint-ите от компилирания код. За да дебъгваме нормално е достатъчно да пуснем виртуалната машина в класически режим (Project Properties | Run | VM parameters = "-classic"). Другият по-малък проблем е че често след като сложим breakpoint се получава така, че е невъзможно да го премахнем и трябва "по метода на Windows" да излизаме от JBuilder и да го стартираме отново. Още един бъг в дебъгера е, че не винаги работи

възможността за показване стойността на променливи чрез поставяне курсора на мишката над тях. Надяваме се от Borland (името на компанията вече не е Inprise) скоро да пуснат някакъв patch за отстраняване на тези проблеми, заради които аз лично все още пиша на JBuilder 3.0 и поне за момента не смятам да преминавам на новата версия. Като говорим за лошите страни на JBuilder, не можем да не споменем и лошите неща, които идват от това че продуктът е написан изцяло на JAVA. Става въпрос за това, че JBuilder е доста по-бавен, от старите версии, които бяха чисти Windows приложения (версии 2.0 и 3.0). За това има две причини: Първата е, че библиотеката за графичен user interface javax.Swing е проектирана и реализирана не достатъчно добре и затова работи доста тромаво. Въпреки опитите за оптимизиране, които прави монополистът Sun (<http://java.sun.com>) в JDK 1.3, тя си остава едно от най-слабите места на JAVA. Втората причина е, че JAVA по принцип си е бавна, понеже се интерпретира и спецификацията на езика не позволява да се създаде бърза виртуална машина. Други проблеми с JAVA графичната среда, на която е базиран новият Jbuilder, са че не могат да се използват предимствата на операционната система – настройка на шрифтовете (не можем да променим шрифта на менютата, редакторът разполага е ограничен брой шрифтове), управление на хардуера (например колелцето на мишката не работи), интеграцията с операционната система и други програми е значително затруднена (JAVA няма registry) и т.н. Друго нещо, което го имаше в JBuilder 3.0, а в новата версия го няма е възможността за изпълнение на проекта в отделен конзолен прозорец, което си има някои предимства пред execution log-a, който ползва новата версия. Следващият неприятен проблем (не е ясно дали е бъг, по-вероятно е да е направено нарочно) е че по време на работа клавиатурата постоянно се превключва в режим на кирилица. Това се случва при всяка поява на нов Swing прозорец и води до доста неприятни неща. Проблемът най-вероятно е че от Sun са се престарали и са добавили автоматичната смяна на “keyboard layout”-а като стандартен feature в JDK 1.3. Едно от решенията е да се сложи “default locale”-а на Windows да бъде “United States”, а не “Bulgaria”. Това обаче обърква останалите Windows приложения (например датите се изобразяват по американския стандарт) и не е много препоръчително. Ето защо аз лично се зах да реша проблема интелигентно и написах специален plug-in за JBuilder, който казва на JAVA виртуалната машина да използва американския locale. Можете да го изтеглите от <http://www.nakov.com/produkti.html>. Следващият проблем, с който се сблъскахме е че колелцето на мишката, което служи за скролиране на активния прозорец не работи. Това отново е проблем на JAVA, но и за него си има решение. Понеже JBuilder поддържа plug-ins, (наричат ги “Open Tools”), имаме възможност да подобряваме сами средата, в която работим. Всеки може да напише Open Tool на JAVA и да си го инсталира. Позволява се доста мощно взаимодействие със средата, като добавяне на собствени менюта, обработка на сорс кода по време на редактиране, прихващане на клавишни комбинации и още много други. Готови Open Tool-ове, за разширяване на средата на JBuilder можете да намерите на адрес <http://www.borland.com/jbuilder/resources/jbopentools.html>. На посочения адрес ще намерите Open Tool за поддръжка на колелцето на мишката (за съжаление не работи при разглеждане на help-a), за интеграция с Visual SourceSafe, както и още много други. Друг досаден проблем е, че при натискане на F1, помощната информация се появява като диалог в текущия прозорец, като скрива работния екран, а не в отделен прозорец, както беше в предходните версии. Не е ясно дали това е направено

нарочно, но със сигурност е изключително неудобно. Решение за този проблем засега нямаме. Друго нещо, което не е точно проблем, но е интересно да се знае е как да накараме wizard-ите за EJB и CORBA да станат достъпни, понеже те по подразбиране са забранени и ги няма по менютата. Едно решение е да инсталираме WebLogic или Inprise Application Server и да настроим по подходящ начин опциите за EJB и CORBA от Tools | Enterprise Setup.

Безплатната версия на JBuilder 4.0 – JBuilder 4.0 Foundation можете да изтеглите от адрес: <http://www.borland.com/jbuilder/foundation/>. Повече информация за продукта можете да намерите на сайта на Borland – <http://www.borland.com>. Презентацията на продукта от Пловдивския панаир можете да изтеглите от <http://www.nakov.com/articles/>. Официален представител на Borland за България фирма PowerPartner. От нея също можете да получите актуална информация.

В заключение можем да кажем, че JBuilder 4.0 интегрира Web, Database, Mobile и Enterprise разработката на приложения в мощен пакет от инструменти, вградени в изключително удобна среда. От Borland полагат много усилия за да бъдат лидери на пазара за средства за разработка на приложения с JAVA и особено в средствата за Enterprise разработка и тази тяхна политика се оказва доста успешна. Това се потвърждава и от десетките награди, които е спечелил JBuilder. И все пак колкото и да е добър JBuilder 4.0, по-удобна среда за програмиране от тази на Delphi, все още няма, въпреки многобройните опити за създаване на такава.