

Borland C++ Builder 5

Pro nové požadavky

Neuplynul ani rok od uvedení Borland C++ Builderu 4 a nová verze vizuálního vývojového nástroje pro C++ od firmy Inprise přichází na trh – nyní tedy s číslem 5.

Podobně jako u předchozích verzí je Borland C++ Builder 5 (BCB 5) dostupný ve třech provedeních, která se liší především rozsahem současně dodávané knihovny VCL, tedy počtem komponent, a množstvím dodatečných nástrojů.

Provedení Standard obsahuje překladač, vizuální vývojové prostředí, běžné ladicí nástroje a asi 85 komponent pro vývoj uživatelského rozhraní a jiných důležitých součástí aplikací. Firemní materiály toto provedení charakterizují jako nástroj určený pro seznámení s jazykem C++, s vývojovým prostředím a s programováním pro Windows.

V provedení Professional najdeme přes 150 komponent; navíc jde především o komponenty pro práci s databázemi a pro internetové aplikace. Přibyly také některé databázové a ladicí nástroje. Součástí dodávky je i lokální databázový server Interbase. Toto provedení lze již využít k vývoji profesionálních aplikací.

Provedení Enterprise obsahuje cca 200 komponent. Navíc přináší mj. nástroje pro podporu týmové práce (TeamSource) a pro vývoj distribuovaných aplikací podle standardů CORBA (VisiBroker 4.0) i COM. Součástí dodávky je plnohodnotný databázový server Interbase 5.6. Podobně jako v předchozích verzích zde najdeme také komponenty MIDAS 3 pro vytváření aplikací střední vrstvy. Pro recenzi jsme měli k dispozici právě provedení Enterprise.

Novinky

Změny, které BCB 5 přinesl, ukazují, že hlavním předmětem zájmu firmy Inprise se stává internet a tvorba distribuovaných aplikací. Další – a neméně významné – novinky se týkají přístupu k databázím a ladění. Vedle toho tu najdeme i změny v prostředí, zcela nové nástroje atd. Za hlavní novinky lze označit:

Součástí dodávky je nová verze VisiBrokeru, nástroje pro tvorbu distribuovaných aplikací podle standardu CORBA.

V BCB 5 lze vyvíjet aplikace podle standardu COM+.

K dispozici jsou komponenty Internet Express, usnadňující vytváření aplikací střední vrstvy, které se chovají jako aplikace pro WWW server a zároveň jako klient ve vícevrstvé aplikaci.

Komponenty ADO Express umožňují přístup k databázím prostřednictvím objektů ADO.

Komponenty Interbase Express zapouzdřují nativní rozhraní serveru Interbase.

Stránka palety Servers obsahuje komponenty pro ovládání OLE serverů, jako je Word, Excel atd., prostřednictvím automatizace (automation).

Nový návrhář datových modulů usnadňuje vytváření databázových aplikací.

Při hledání chyb v práci s pamětí a ostatními prostředky lze využít CodeGuard.

BCB 5 umožňuje nastavovat různé volby překladače pro jednotlivé uzly projektu.

Pro snadný vývoj aplikací, u kterých lze předpokládat lokalizaci (převod do cizojazyčného prostředí), slouží Borland Translation Suite.

Další drobná vylepšení se týkají správy projektů, některých nových komponent (rámy), nových "šamanů" (wizards) v zásobníku objektů aj.

Programovací jazyky

Pokud znáte implementaci jazyků C, C++ a Object Pascal z BCB 4, stačí říci, že jejich implementace ve verzi 5 je prakticky stejná. BCB implementuje programovací jazyk C, odpovídající standardu ISO 9899 z r. 1990 s pozdějšími technickými dodatky (funkce pro práci se širokými znaky ap.). Navíc obsahuje běžná rozšíření, z nichž asi nejdůležitější jsou strukturované výjimky navržené firmou Microsoft pro prostředí Win32; BCB umožňuje používat je i v C++.

Implementace jazyka C++ téměř odpovídá standardu ISO 14882 z r. 1998. Asi nejdůležitější odchylky od tohoto standardu jsou:

Konstrukci try/catch nelze dosud použít jako tělo funkce. (Obvykle to neznamena nic jiného, než že musíme psát o dvě složené závorky navíc. Skutečný význam to ale má v případě konstruktorů, neboť zde můžeme do bloku try uzavřít i inicializační část, ve které voláme konstruktory předků a složek, a tak zachytit výjimky, které se z nich rozšíří.)

BCB 5 nepodporuje exportní šablony; klíčové slovo export je "vyhrazeno pro budoucí použití". (Jde o způsob řízení vzniku instancí šablon, zařazený do standardu až na poslední chvíli.)

Nicméně ve srovnání s předchozí verzí tu přece jen najdeme několik drobných novinek. Jde především o modifikátory `__msreturn` a `__msfastcall`, specifikace `__declspec(nothrow)` a některé další; nové jsou i některé přepínače překladače. K tomu se vrátíme v samostatném oddílu věnovaném kompatibilitě s MS Visual C++.

Podobně jako v předchozích verzích tu najdeme také rozšíření nezbytná pro kompatibilitu s Object Pascalem z Delphi a zejména s knihovnou VCL. Implementace jazyka Object Pascal odpovídá Delphi 5; součástí instalace je překladač DCC32, verze 13. To umožňuje mj. začlenit do projektu v BCB moduly vytvořené v Delphi.

Knihovny

Neodmyslitelnou součástí každého dnešního vývojového nástroje je řada programových knihoven. To se týká i BCB 5.

Implementace standardní šablonové knihovny jazyka C++ pochází od firmy Rogue Wave. Je stejná jako v předchozí verzi.

Aktuální verze knihovny vizuálních (a nevizuálních) komponent má číslo 5.0 a stejně jako v předchozích verzích je napsána v Object Pascalu. Prošla lehkými změnami, které by neměly působit problémy z hlediska zpětné kompatibility. Přibylo několik nových komponent, především ve verzích Professional a Enterprise; budeme o nich hovořit v odstavcích věnovaných databázím, internetu a distribuovaným aplikacím. Z ostatních novinek VCL stojí za zmínku ráma (frame). Rám se v době návrhu chová podobně jako okno; představuje kontejner na komponenty, který lze později buď využít jako samostatné okno, nebo vložit jako celek do jiných oken (i opakovaně). Celý rám i jednotlivé komponenty v něm lze po vložení dále upravovat.

Tradiční borlandská knihovna OWL (Object Windows Library) pro vytváření aplikací pro Windows již není součástí instalace BCB 5. Nezmizela ovšem úplně; pokud ji potřebujete, najdete na doprovodném CD s označením Companion Tools instalaci verze 5.5.

Spolu s BCB 5 se dodává knihovna Microsoft Foundation Classes (MFC), verze 4.2. Najdeme tu jak zdrojové texty (jejich instalace je volitelná), tak přeložené moduly. V zásobníku objektů je navíc k dispozici šaman, který umí vytvořit kostru aplikace založené na této knihovně.

Z dílny Microsoftu pochází i knihovna ATL, která slouží k programování aplikací založených na standardu COM. Využívají ji šamani, kteří vytvářejí programové kostry řídicích prvků ActiveX, objektů COM+ atd. Najdeme je na kartě ActiveX v zásobníku objektů.

Knihovna kontejnerů (BIDS), která provázela borlandské překladače C++ od počátku devadesátých let (přesněji od Borland C++ 2.0), již také není součástí instalace; najdeme ji na doprovodném CD v oddílu OWL. Poznamenejme ale, že STL obsahuje přenositelné implementace podobných datových struktur, takže v nových aplikacích je BIDS nejspíš zbytečná.

Internet

Novinky v programování pro internet považuje firma Inprise nepochybně za stěžejní; svědčí o tom mj. palcové titulky na krabici. Podívejme se na ně tedy podrobněji.

Už v přehledu novinek jsme se zmínili o komponentách Internet Express. Najdeme je i ve verzi Professional a slouží k vytváření programů, které se chovají jako aplikace pro WWW server a zároveň jako klient ve vícevrstvé aplikaci. Komponenta XMLBroker získává z aplikačního serveru datové pakety v XML, zpřístupňuje je komponentám, které generují webové stránky, a zprostředkovává aktualizace. Komponenta MidasPageProducer slouží k vytváření webových stránek s databázovými informacemi získanými z aplikačního serveru. Vytvořená stránka obsahuje datové pakety v XML a vnořený javaskript, který umožňuje s daty pracovat.

Další novinkou je šaman, který vytvoří základ aplikace pro WWW server; najdeme ho v zásobníku objektů na stránce New. Umí vytvářet aplikace využívající rozhraní ISAPI/NSAPI (dynamické knihovny spouštěné WWW serverem), CGI skripty (v podstatě samostatné konzolové aplikace, které přijímají požadavky prostřednictvím standardního vstupu a zapisují výsledky – vytvořené HTML stránky – do standardního výstupu) a Win-CGI skripty, což jsou aplikace pro Windows, které dostávají požadavky v souboru .ini a výsledky zapisují do souboru.

Vedle toho zde najdeme komponenty, se kterými jsme se mohli setkat už v předchozích verzích a které zapouzdřují nejrůznější aspekty programování pro internet – např. WWW browser, FTP, sokety atd. Palety s těmito komponentami byly ovšem poněkud přeorganizovány.

Databáze

Podobně jako předchozí verze nabízí i BCB 5 dva základní druhy databázových komponent: komponenty pro přístup k datům (data access components) zapouzdřují spojení z databázemi, dotazy, tabulky, uložené procedury apod., zatímco vizuální databázové komponenty (mřížka, databázové editační pole ap.) získaná data zobrazují a umožňují je editovat.

V předchozích verzích BCB využívaly komponenty pro přístup k datům borlandský databázový stroj (BDE), který bylo nutno šířit s aplikací, konfigurovat ho atd. V BCB 5 s ním můžeme samozřejmě pracovat také; vedle toho ovšem máme i další možnosti.

První z nich představují komponenty ADO Express, které zpřístupňují microsoftské databázové objekty ADO (ActiveX data objects). Zřejmou výhodou tohoto přístupu je, že objekty ADO jsou součástí řady běžně užívaných programů, a proto se zpravidla nemusíme starat o jejich šíření.

V aplikacích, které používají server InterBase, můžeme použít komponenty Interbase Express. Jde opět o komponenty pro přístup k datům, které ovšem využívají vlastního rozhraní serveru InterBase, a proto mohou nabízet širší paletu služeb než univerzální komponenty založené na BDE.

Automatizace

Automatizace (automation) představuje volání aplikací, které se chovají jako COM server; potřebné informace o takovémto serveru jsou uloženy v tzv. typové knihovně (type library). Stejně jako v BCB 4 i zde najdeme šamana, který umí typovou knihovnu importovat. Navíc na paletě v BCB 5 najdeme stránku Servers, která obsahuje komponenty zapouzdřující běžné COM servery (Word, Excel a další).

Distribuované aplikace

BCB 5 podporuje tvorbu aplikací podle standardu COM+. V zásobníku objektů na stránce ActiveX najdeme mj. šamany pro vytvoření objektu události COM+ nebo pro vytvoření transakčního objektu. Součástí verze Enterprise jsou, podobně jako v předchozích verzích, komponenty MIDAS, které slouží k vytváření střední vrstvy aplikací.

Stejně jako v předchozí verzi i zde najdeme také podporu pro vytváření distribuovaných aplikací podle standardu CORBA. IDE nabízí několik šamanů, kteří umějí vytvořit soubor IDL, CORBA server, CORBA klient atd. Součástí dodávky je VisiBroker 4, což je nejen nástroj pro vývoj CORBA aplikací, ale i běhový systém (implementace ORB).

Ve srovnání s předchozí verzí je zde jedna významná novinka: VisiBroker 4 implementuje POA (portable object adaptor), zatímco předchozí verze implementovala pouze BOA (basic object adaptor). Připomeňme si, že objektový adaptér je součástí, která zabezpečuje komunikaci CORBA serveru s objektovou sběrnici (ORB). Zatímco BOA je nepřenositelná implementace objektového adaptéru (specifikace nebyla dostatečná), POA je plně přenositelná. (Poznamenejme, že rozhraní BOA z předchozích verzí VisiBrokeru zůstalo zachováno, funguje nad POA, a proto aplikace vyvinuté v předchozí verzi lze spustit i v prostředí VisiBrokeru 4.)

Kompatibilita s Visual C++

Převádění projektů mezi různými vývojovými nástroji je běžnější, než by se mohlo zdát; příčiny mohou být různé, ale potíže, se kterými se pak vývojář musí vyrovnat, jsou vždy stejné. Firma Inprise se rozhodla vývojářům maximálně usnadnit přechod od Microsoft Visual C++ (MSVC) k BCB; proto již předchozí verze obsahovala šamana, který uměl převést projekt z MSVC do BCB. Setkáme se s ním i v současné verzi, stejně jako s prepínači umožňujícími přeložit některé microsoftské nestandardní konstrukce.

Nově přidaná klíčová slova `__msfastcall` a `__msreturn` emulují konvence používané pro předávání parametrů a pro vracení malých struktur hodnotou používané v MSVC. Specifikace `__declspec(nothrow)` vlastně nahrazuje standardní specifikaci `throw()`, určující, že se z dané funkce nesmí rozšířit žádná výjimka; v BCB je pouze kvůli kompatibilitě zdrojového kódu s MSVC. (Připomeňme si, že MSVC 6.0 ještě neimplementuje specifikaci výjimek v deklaraci funkce.) Také další deklarace `__declspec` mají podobný účel.

Ovšem vedle kompatibility ve směru k BCB je třeba také uvažovat o kompatibilitě ve směru k MSVC, neboť migrace probíhá v obou směrech. Při přechodu k MSVC není standardně k dispozici utilita pro převod projektů; navíc se v BCB 5 změnil formát projektového souboru.

Další problémy si můžeme přivodit, budeme-li v BCB používat nejnovější vlastnosti C++, které jsou sice ve standardu, ale které MSVC dosud neobsahuje – například vnořené šablony nebo už zmíněné specifikace výjimek v deklaraci funkce.

Svéráznou kapitolu představuje standardní knihovna C++ (STL). Ta totiž podstatným způsobem využívá pokročilé konstrukce jazyka C++, mj. právě vnořené šablony. Pokud je překladač nepodporuje, musí je implementace STL různým způsobem obcházet, ale tak vzniknou konstrukce, které nebudou přenositelné do standardní implementace STL (a programy využívající standardní implementace STL budou obtížně přenositelné do tohoto prostředí). Navíc starší verze návrhu normy, z níž MSVC 6 vychází, obsahovala některé konstrukce, které se do standardu nedostaly – například operátor `new`, který má jako dodatečný parametr alokátor – a to může také zkomplikovat přenos programů.

Poznamenejme ale, že problémy při přenosu programů užívajících STL mezi MSVC a BCB se objeví, až když začneme využívat pokročilejších vlastností této knihovny, jako je možnost řízení alokace kontejnerů pomocí alokátorů.

Vývojové prostředí

Celkový vzhled integrovaného vývojového prostředí (IDE) ani způsob práce s ním se nezměnil, nová verze přinesla jen několik drobných vylepšení. Přibyli noví šamani, zejména v zásobníku objektů (object repository). Jmenujme alespoň šamana pro vytvoření aplikace pro webový server, pro vytvoření nového zdrojového souboru v C, hlavičkového souboru atd. Zmizel šaman pro generování aplikací založených na OWL.

Při vývoji databázových aplikací poslouží návrhář datových modulů. Ten nyní umí zobrazovat hierarchii databázových komponent a vztahy mezi nimi; tyto vztahy v něm lze i graficky editovat. Za zmínku stojí, že soubory `.dfm` obsahující popis formulářů se nyní ukládají v textové podobě; lze si ovšem předepsat, že je chceme ukládat jako binární soubory, stejně jako v předchozích verzích. Příjemná je i možnost zobrazovat v inspektoru objektů hodnoty nejen textově, ale i graficky – ikonou.

Seznam úkolů (To Do List) nabízí možnost vést si přímo v prostředí přehled toho, co je ještě potřeba udělat, co již je hotovo apod. Vedle centrálního seznamu lze takto spravovat i seznamy vložené jako komentáře do zdrojových souborů.

IDE nyní také dovoluje pojmenovat a zapamatovat si různá rozložení oken a později je používat. Lze také předepsat, že určité rozložení se má používat implicitně při návrhu nebo při ladění.

Projekt

V této oblasti přináší verze 5 několik vylepšení. Prvním z nich je možnost nastavovat zvláštní volby pro překlad a ladění jednotlivých uzlů projektu (konečně...!). Také projektový soubor (`.bpr`) má nyní nový formát – ukládá se v jazyce XML.

Příjemnou novinkou je překlad na pozadí – nyní můžeme po dobu překladu editovat zdrojový text nebo jinak pracovat s prostředím. (Něco takového oceníme zejména u rozsáhlých projektů, kde překlad trvá i desítky minut nebo hodiny.) Potěší i možnost zadat při vytváření nového projektu již hotový soubor s funkcí `main()` – ani to kupodivu předchází verze neuměly a museli jsme si vypomáhat různými triky.

Ladění

BCB 5 nabízí obvyklý standard – krokování programu (i distribuovaného) jak na úrovni zdrojového textu, tak na úrovni disasemblovaného strojního kódu, sledování hodnot proměnných, používání zarážek atd. Vedle toho však přináší několik užitečných novinek.

Jednou z nich je možnost sdružování zarážek (breakpoints) do pojmenovaných skupin a definování akcí sdružených se zarážkami. Takovou akci může být povolení, nebo zakázání skupiny jiných zarážek, změna způsobu ošetřování výjimek při ladění atd. Tak lze třeba zajistit, že určitá zarážka bude aktivní, pouze když program předtím projde jistým místem.

Pro ladění distribuovaných aplikací na vzdáleném počítači slouží Remote Debug Server, který je nutno instalovat na cílovém počítači. BCB se také umí připojit k běžícímu procesu a začít ho ladit.

Při hledání chyb v práci s pamětí (zápis do nealokované paměti, opakované uvolnění téže části paměti apod.) můžeme využít CodeGuard. Možná si vzpomenete, že před několika lety uvedla firma Borland tento nástroj na trh jako samostatný produkt, který se dal používat spolu překladači s Borland C++ 4.5. Nyní se jeho pokročilejší verze stala součástí BCB 5.

Poslední drobností, o které se zde zmíním, je možnost sledovat při ladění hodnoty v registrech koprocesoru (a MMX).

Co a jak

Dodávka BCB 5 Enterprise obsahuje kromě instalačního CD i manuál Developer's Guide, CD označené Companion Tools, které obsahuje především další komponenty od třetích firem, a instalační CD pro Delphi 4 Professional a JBuilder 3 Professional.

Chceme-li instalovat BCB 5 Enterprise, potřebujeme PC s procesorem Intel Pentium/90 MHz (ale raději lepší) vybavený nejméně 32 MB RAM (doporučeno alespoň 64 MB), mechanikou CD-ROM, grafickou kartou VGA nebo lepší (karty VGA ještě existují?), odpovídajícím monitorem a myší. Nezbytný je operační systém MS Windows 2000 (případně 9x) nebo Windows NT 4.0 se Service Pack 3 nebo pozdějším. Úplná instalace BCB 5 Enterprise zabere na disku 388 MB, kompaktní 253 MB.

Nároky ostatních provedení se liší jen diskovým prostorem: pro BCB 5 Professional potřebujeme 360 MB, resp. 240 MB, pro BCB 5 Standard 185 MB, resp. 120 MB. (Tyto údaje se týkají pouze instalaci BCB, nezahrnují server InterBase, nástroj TeamSource, dodatečné knihovny ani další nástroje, které si lze z dodaných CD instalovat zvlášť.)

Dojem

Přiznám se, že už mě nebaví psát stále dokola, že je to "silný nástroj"; docela rád bych si na nějaký produkt zanádal. BCB 5 mi však v tomto směru mnoho možností nedává. Jistě, najdou se zde drobnosti, které nefungují tak, jak by měly, ne vždy jsou vlastnosti produktu dokumentovány dostatečně, ale to na věci nic nemění – je to silný nástroj.

Miroslav Virius