

APLIKACE Z DELPHI 6 POBĚŽÍ I NA LINUXU

Zanedlouho tomu budou už dva roky, kdy firma Borland dodala na trh vývojové prostředí Delphi ve verzi 5. Letos v březnu Borland dále rozšířil spektrum svých vývojových nástrojů projektem s pracovním názvem Kylix, který přenesl Delphi z platformy Windows do prostředí operačního systému Linux (kódové označení Kylix nakonec posloužilo i jako oficiální název výsledného produktu).

Velká odlišnost operačních systémů Windows a Linux si pochopitelně v produktu Kylix vynutila odlišné řešení některých funkčních oblastí, zejména přístupu do databáze a knihoven grafického uživatelského rozhraní. I když převedení aplikace vyvinuté v Delphi 5 na platformu Linux není nijak složité, musí dnes vývojář usilující o vývoj pro obě platformy udržovat některé zdrojové knihovny ve dvou verzích.

Podívejme se proto na některé novinky a vylepšení, které přináší testovací verze tohoto vývojového prostředí a které nepochybně najdeme i v "ostré" verzi produktu.

Souhrnně

Pokud bychom měli shrnout všechny významné novinky v Delphi 6 do jednoho seznamu, obsahoval by následující oblasti: zdokonalené vývojové prostředí, nové možnosti kompilátoru, rozšíření VCL, nová sada grafických komponent CLX, nový přístup k databázím dbExpress, vylepšená podpora tvorby třívrstevných aplikací, podpora protokolu SOAP, snadná tvorba webových aplikací komponentami SiteExpress a dalších internetových aplikací komponentami Indy, podpora práce s XML dokumenty a vylepšená podpora technologií COM+ a CORBA.

Vylepšené IDE

Začneme u novinek v integrovaném vývojovém prostředí (IDE). Při návrhu formuláře můžeme v Delphi 6 využít nového hierarchického pohledu na použité komponenty, který ve formě stromu přehledně znázorňuje logické vazby mezi komponentami – např. fakt, že tlačítko leží na panelu, nebo které sloupce jsou definovány v komponentě ListView. Tento pohled lze využít obdobně jako návrhový formulář, zvolené komponenty je možné editovat v inspektoru objektů a díky mechanismu drag & drop lze měnit i umístění určité komponenty ve zmiňované hierarchii.

Komponenty nabízené v paletě komponent již netvoří statickou množinu, ale v daný okamžik jsou vývojáři nabízeny jen ty komponenty, které lze umístit do právě navrhovaného kontejneru. Např. při editaci datového modulu v paletě komponent nenajdeme žádnou vizuální komponentu. Dalšího rozšíření doznal i inspektor objektů, který nyní umožňuje i editaci atributů a událostí objektů, na něž obsahuje určitá komponenta referenci. Např. při editaci vizuální databázové komponenty lze editovat i podřízené komponenty Table a Database bez nutnosti otevření datového modulu, ve kterém se tyto komponenty pro přístup k databázi nacházejí. Inspektor objektů může být do značné míry přizpůsoben představám uživatele, a to jak po obsahové, tak po vizuální stránce. Nově lze kupříkladu nastavit libovolné barevné schéma inspektora.

Efektivitu práce vývojáře při psaní kódu již tradičně zvyšuje sada nástrojů souhrnně označovaná jako Code Insight. Automatické doplňování kódu v Delphi 6 funguje již i v úvodní "interface" části zdrojové knihovny. Okno pro výběr z přípustných dokončení výrazu dostalo novou tvář a jeho velikost lze kdykoliv změnit myší. Seznam výrazů nabízených v okně je filtrován řetězcem zadaným uživatelem.

Editor kódu zdrojových knihoven byl rozšířen o podporu různých typů pohledů na obsah editovaných knihoven. Této nové vlastnosti je využíváno např. při budování webových aplikací pomocí technologie SiteExpress (viz dále), kdy každá knihovna reprezentuje jednu výslednou dynamickou HTML stránku. Vývojář má možnost nahlížet na knihovnu jako na zdrojový kód v Pascalu, výsledný zdrojový kód v jazyce HTML, náhled v integrovaném prohlížeči, a pokud je stránka generována např. pomocí standardu XSLT, pak je k dispozici i náhled na zdrojový XML a

XSL kód.

Nové možnosti kompilátoru

Kompilátor v Delphi 6 zavádí novou direktivu \$IF, která na rozdíl od direktivy \$IFDEF umožňuje ve výrazech vyhodnocovat i konstanty definované v jazyce Pascal. Pro testování existence předchozí definice určitého symbolu slouží funkce Defined(), existenci deklarace pascalovské konstanty detekuje funkce Declared(), např. {\$IF Defined(WIN32) and Declared(MyConst)} ... {\$IFEND}. Integrovaný překladač assembleru byl kompletně přepracován a poskytuje podporu nových instrukcí pro MMX, SIMD, Pentium Pro, Pentium III, AMD K7, Enhanced MMX, Intel SSE a AMD Enhanced 3D.

Rozšíření VCL

Jako v každé předchozí verzi i v Delphi 6 najdeme řadu vylepšení ve knihovně vizuálních komponent VCL. Mnoho komponent přibýlo a mnoho komponent bylo dále vylepšeno. Celkově Delphi 6 v edici Enterprise nabízí vývojáři přes 430 komponent zařazených v paletě komponent do 30 záložek.

Nejdříve se stručně podíváme na některé nové vizuální komponenty. Komponenta LabeledEdit představuje jednořádkové vstupní pole s popiskem vzniklé spojením komponent Edit a Label do jediné. Její výhodou je jednodušší manipulace, instance komponenty Label je dostupná přes atribut EditLabel komponenty LabeledEdit. Komponenta ComboBoxEx rozšiřuje možnosti klasického kombinovaného seznamu o jednoduché použití obrázků a odsazení ve vizuálních prvcích seznamu. Pro výběr libovolné barvy z kombinovaného seznamu slouží nově připravený kombinovaný seznam zapouzdřený v komponentě ColorBox. Pokud od uživatele požadujeme editaci kolekce typu TStringList, ve které uchováváme dvojice klíč/hodnota, můžeme použít nové komponenty ValueListEditor, která pracuje podobně jako inspektor objektů.

V záložce Dialogs přibyla komponenta PageSetupDialog reprezentující dialogové okno pro nastavení vzhledu stránky před tiskem, příp. pro nastavení vlastností tiskárny.

Pro komfortnější správu akcí Delphi 6 přináší komponentu ActionManager rozšiřující starší komponentu ActionList a dále nové komponenty MainMenuActionBar a ToolActionBar, které nabízejí alternativu ke stávajícím komponentám představujícím menu aplikace a lištu nástrojů. Jejich předností je, že jejich obsah (tj. položky menu a tlačítka na liště nástrojů) lze nastavit pomocí mechanismu drag & drop tak, že z editoru akcí přetahujeme jednotlivé akce na pozici, kam chceme odkaz na příslušnou akci umístit. Tento způsob nastavení menu a panelů je dostupný nejen v návrhu, ale i za běhu aplikace, tedy i každému uživateli aplikace. Nastavení těchto komponent se může volitelně automaticky ukládat do určeného souboru a automaticky obnovovat při příštím spuštění aplikace.

Sada komponent CLX

Portace Delphi do prostředí operačního systému Linux a následná snaha o kompatibilitu zdrojových kódů aplikace pro obě platformy si vynutila vznik nové kompletní sady vizuálních komponent označovaných jako CLX (čteno klix). Stávající implementace VCL je příliš svázaná s WinAPI a ovládacími prvky Windows, než aby bylo možné ji přenést na Linux. Za společný, na platformě nezávislý základ posloužila grafická knihovna Qt firmy Trolltech, kterou komponenty CLX využívají a která musí být pochopitelně šířena s vyvinutou aplikací. Komponenty CLX nesou stejná jména jako jejich protějšky ze sady VCL, jsou však implementovány zcela odlišným způsobem v knihovnách začínajících písmenem Q (např. QGraphics, QControls, QForms, QDialogs, QStdCtrls, atd.). Vývoj aplikací založených na CLX komponentách se nijak neliší od vývoje stávajících vizuálních aplikací, vývojář pouze při založení projektu specifikuje odlišný typ aplikace. Při následném vytváření formulářů již Delphi generují formuláře z knihovny CLX a v paletě komponent nabízí pouze CLX vizuální komponenty.

Nové knihovny

Kromě množství nových komponent najdeme v Delphi 6 i několik nových knihoven. Knihovna ConvUtils obsahuje množství obecných funkcí pro konverze a porovnávání hodnot v libovolných měrných soustavách. Standardní měrné soustavy pro kvantifikaci vzdálenosti, plošné míry, objemu, váhy, teploty a času jsou definovány v knihovně StdConv. Pro úplné pokrytí potřeb počítání s proměnnými typu čas a datum (TDateTime) slouží skoro 150 nových funkcí a procedur v knihovně

DateUtils. Práci s řetězci nám v Delphi 6 kromě funkcí definovaných v knihovně SysUtils usnadní i mnoho dalších funkcí v nové knihovně StrUtils. Pro práci s binárně kódovanými desetinnými čísly byla zavedena knihovna FMTCBCD. Matematická knihovna Math definuje nové konstanty pro kladné a záporné nekonečno a sadu nových funkcí pro trigonometrické výpočty, převody stupňových jednotek a další. Za zmínku stojí zejména funkce IfThen, usnadňující podmíněné zpracování. Funkce má tři parametry. První parametr představuje logický výraz – pokud je pravdivý, vrací funkce hodnotu druhého parametru, pokud je nepravdivý, vrací hodnotu třetího parametru. Funkce je přetížena pro typy integer, double, int64 a string.

Implementace typu Variant byla přesunuta ze základní knihovny System do dvou nových knihoven Variants and VarUtils, kde byly definovány i některé nové funkce pro práci s tímto typem. Nová implementace Variantu umožňuje velmi jednoduchým způsobem – odvozením potomka od obecné třídy TCustomVariantType definovat pro Variant vlastní datové typy a přetěžovat aritmetické operace nad nimi prováděné. Příkladem může být implementace komplexních čísel jako zvláštního podtypu typu Variant v knihovně VarCmplx, kde se nachází i mnoho funkcí podporujících počítání s komplexními čísly.

dbExpress – nová cesta k datům

Mnoha vývojářům je už řadu let trnem v oku technologie BDE (Borland Database Engine) využívaná pro přístup k libovolným databázovým systémům. Delphi 5 přineslo alternativní způsob přístupu k databázím pomocí ADO komponent Microsoftu a pomocí nativních komponent označovaných jako InterBaseExpress pro práci s databázovým systémem InterBase. Při portaci Delphi na operační systém Linux však nebylo možné využít ani ADO komponent založených na COM a prostředí Windows, ani nemělo smysl portovat dnes už zastaralou technologii BDE. Hledal se proto nový obecný způsob práce s libovolnou databází, který bude integrován jak v produktu Kylix, tak v Delphi 6 a umožní přenositelnost zdrojového kódu databázových aplikací mezi oběma platformami. Tímto novým mechanismem přístupu k datům se stala technologie dbExpress, skládající se z množiny velmi lehkých databázových ovladačů. Pro každý podporovaný databázový systém existuje ovladač, který transformuje specifické rozhraní knihoven pro přístup k databázi dodávaný producentem databáze (tzv. databázového klienta) na jednotné rozhraní dbExpress. Na rozdíl od technologie BDE, která plně řídila komunikaci s databází a vývojáři umožnila jen přístup k záznamům ve své mnohdy obrovské vyrovnávací paměti, je dbExpress koncipován jako velmi tenká vrstva minimalizující komunikaci s databází. Pro zpracování datových množin vrácených z databáze po provedení SQL příkazu poskytuje dbExpress pouze jednozáznamovou vyrovnávací paměť a umožňuje procházet záznamy pouze jednosměrně. Díky tomu lze vytvářet extrémně rychlé databázové aplikace s minimálními nároky na operační paměť. Pokud chce vývojář pracovat nad datovou množinou obousměrně jako dříve a udržovat všechny záznamy v nějaké vyrovnávací paměti, může využít komponentu SQLClientDataSet, která načítá data přes dbExpress a spravuje je v paměťové tabulce.

V oblasti třívrstevných aplikací došlo k přejmenování technologie MIDAS nabízející služby pro přenos databázových informací mezi aplikačním serverem a tenkým klientem na DataSnap. K podporovaným protokolům Sockets, COM/DCOM, CORBA a HTTP určeným pro komunikaci mezi distribuovanými komponentami v Delphi 6 přibyl protokol SOAP. V záložce DataSnap dále najdeme několik nových komponent usnadňující vytváření třívrstevných databázových aplikací, např. komponentu SharedConnection umožňující přistupovat k více datovým modulům na aplikačním serveru přes jediné připojení či komponentu ConnectionBroker centralizující skupinu komponent ClientDataSet používajících stejné připojení k aplikačnímu serveru. Ke komponentě ClientDataSet přibýly její varianty BDEClientDataSet, SQLClientDataSet a ADOClientDataSet pro vývoj dvouvrstevných aplikací obsahující i prostředky pro práci nad příslušnou databázovou technologií a zajišťující uložení dat v operační paměti.

Webové aplikace snadno a rychle

Jednou z oblastí, které byla při přípravě Delphi 6 věnována mimořádná pozornost, je oblast vývoje aplikací pro internet. Technologie WebBroker, která je primárním prostředkem pro budování webových aplikací v Delphi, doznala od třetí do páté verze tohoto nástroje jen malých změn a vývoj aplikací tohoto typu byl poměrně pomalý. Nová verze Delphi přináší přepracovanou podobu technologie WebBroker umožňující vývoj CGI, WinCGI, ISAPI/NSAPI a Apache webových modulů, které lze efektivně provozovat na dnes nejrozšířenějších webových serverech. Testování a ladění vyvíjených aplikací si může programátor značně ulehčit použitím testovacího webového serveru označeného jako "TestSvr", který je součástí instalace vývojového prostředí. Pokud chce vývojář

tohoto serveru využít, vyvíjí svou aplikaci jako speciální webový modul, který pro komunikaci s webovým serverem používá COM rozhraní. Po úplném odladění takové aplikace se drobnou změnou kódu určí cílový webový server, na kterém bude aplikace nasazena.

Technologie WebBroker se opírá o centrální objekt dispečera, který přijímá příchozí požadavky a směřuje je na konkrétní objekty, od kterých se očekává transformace požadavků na konkrétní odpověď, typicky vygenerování příslušné HTML stránky. V předchozích verzích Delphi bylo generování výsledného HTML kódu vývojáři ulehčováno pouze několika komponentami, z nichž nejužitečnější byla komponenta TPageProducer umožňující vytvoření šablon s HTML kódem a uživatelskými značkami, které byly v okamžiku zpracování požadavku nahrazeny vývojářem definovanými výrazy. Vytváření rozsáhlejších webových aplikací a jejich údržba byly celkově velmi pracné. Delphi 6 obsahuje nadstavbu nad vrstvou WebBroker nazvanou SiteExpress. Vývojář při budování své aplikace používá pro vytvoření každé dynamické HTML stránky zvláštního typu webového modulu (potomka třídy TWebPageModule). Každý takový modul představuje kontejner definovaný v samostatné zdrojové knihovně, do něhož lze vkládat nevizuální komponenty ovlivňující zpracování příchozího požadavku. Při inicializaci aplikace dochází k registraci všech webových modulů definovaných v jednotlivých knihovnách projektu u centrálního dispečera. Dispečer při příchodu požadavku z jeho URL adresy pozná, který modul je za zpracování požadavku odpovědný, aktivuje ho a předá mu požadavek ke zpracování. Při větší zátěži systému, kdy přichází více požadavků na jeden modul současně, umožňuje dispečer vytvoření více instancí třídy určitého modulu a práci ve více vláknech současně. Vlastní vytvoření HTML stránky ve webovém modulu zajišťuje ve většině případů komponenta PageProducer či její nadstavba, která zpracovává HTML šablonu vytvořenou uživatelem. Tato šablona může být přiřazena modulu přímo direktivou \$R a lze ji snadno editovat v editoru zdrojového kódu. Dynamickou složku v šabloně reprezentuje vložený kód ve zvoleném skriptovacím jazyce interpretovaný na straně serveru a uživatelské značky. Skriptovací jazyk má přístup k několika obecným předdefinovaným objektům (např. objekty Application, Page, Session, EndUser...) a dále k objektům nazývaným adaptéry, které programátor k tomuto účelu definuje ve webovém modulu. Některé obecné typy adaptérů SiteExpress nabízí již připravené, např. komponentu DataSetAdapter, která zpřístupní jakoukoliv datovou množinu (potomka třídy TDataSet) skriptovacímu jazyku jak pro čtení, tak pro editaci. Sada komponent SiteExpress podporuje sledování komunikaci s jedním konkrétním uživatelem (session tracking), umožňuje vytvoření seznamu přístupových účtů s určitými právy, kontroluje přístupová práva ke stránkám a případně automaticky požaduje přihlášení, podporuje transformaci obrázků z databáze do po internetu přenositelné formy, umožňuje využití XSL transformace XML dokumentů atd.

Komponenty firmy NetMasters, které v minulých verzích Delphi zajišťovaly práci s protokoly rodiny TCP/IP byly v Delphi 6 nahrazeny sadou známých komponent označených jako Indy (dříve též Winshoes) vyvíjených již několik let jako Open Source.

Podpora XML

Na stále rostoucí uplatnění standardizovaného jazyka XML v praxi reagují Delphi 6 rozsáhlou podporou XML hned v několika směrech. Nová komponenta XMLDocument představuje zapouzdření externího XML parseru do komponenty v Delphi, typicky parseru firmy Microsoft, který je umístěn na většině počítačů (volitelně lze však použít i jiný parser). Přístup k prvkům XML dokumentu a jejich editaci zajišťuje komponenta XMLDocument pomocí množiny rozhraní definovaných standardem DOM. Pokud již při návrhu známe strukturu XML dokumentu, se kterým budeme pracovat, můžeme pomocí speciálního nástroje "XML Data Binding Wizard" vygenerovat soustavu rozhraní a tříd ušitých na míru určitému XML dokumentu, které budou použity pro procházení i editaci dokumentu. Zmiňovaný nástroj umožňuje libovolně pojmenovat definované třídy a rozhraní, jejich metody a atributy a určit typ všech dat obsažených v XML dokumentu. Takto vytvořené mapování XML dokumentu do hierarchické struktury pascalovských tříd lze uložit ve speciálních souborech s příponou .xdb a později znovu použít. Vývojář se díky automaticky generovaným třídám reprezentujícím uzly XML dokumentu nemusí zabývat standardem DOM a píše jednodušší a podstatně přehlednější kód.

V praxi se často setkáváme s požadavkem zpřístupnit jiné aplikaci data uložená v relační databázi v požadované XML formě nebo naopak pracovat s určitým XML souborem obdobným způsobem jako s datovou množinou z databáze. Tyto potřeby lze v Delphi 6 efektivně řešit pomocí dvou základních komponent, které jsou schopny provést transformaci z XML formátu využívaného komponentou ClientDataSet do libovolného jiného XML formátu a obráceně. Předpis pro takovou transformaci je poměrně složitý, a proto Borland dodává samostatný externí nástroj pro generování libovolného popisu transformace (ty se pak ukládají v souborech s příponou .xtr). Pokud chce vývojář

implementovat např. aplikaci umožňující editaci obsahu XML souboru ve standardních vizuálních databázových komponentách, může použít komponentu XMLTransformProvider. Vytváření libovolného XML kódu z údajů v databázi a zpětný zápis změn v tomto kódu do databáze umožňuje komponenta XMLTransformClient.

Vylepšená podpora technologie COM+

Další oblastí, která doznala některých vylepšení, je vývoj aplikací založených na technologii COM/DCOM, resp. COM+. Pro podporu vytváření objektů podporujících propagaci událostí podle technologie COM+ (tzv. Event Objects) nabízí Delphi 6 speciálního průvodce pro definici rozhraní používaného pro šíření a příjem událostí. Průvodce pro vytváření MTS objektů byl nahrazen průvodcem pro vytváření transakčních objektů, který je nyní dostupný už v edici Professional a umožňuje vývoj jak COM+ transakčních objektů, tak i objektů pracujících ve starším prostředí MTS.

Editor knihoven typů (Type Library) byl rozšířen o editaci množiny atributů, které umožňují každému COM objektu nastavit prostředí COM+.

Při vytváření nových COM objektů v prostředí Delphi 5 bylo vždy s třídou implementující COM objekt vygenerováno i nové rozhraní (interface). Průvodce vytvářením nových COM objektů v Delphi 6 umožňuje již v okamžiku zadávání jména nového objektu určit, zda bude objekt implementovat nové nebo již jinde definované rozhraní. Již definované rozhraní lze vybrat ze seznamu rozhraní zaregistrovaných v systémovém registru nebo výběrem příslušné knihovny typů.

Konečně CORBA jak má být

Nová verze vývojového prostředí Delphi přináší i značně vylepšenou podporu vývoje aplikací založených na technologii CORBA. Integrovaný kompilátor jazyka OMG IDL umožňuje na rozdíl od předchozích verzí Delphi vyvíjet aplikace CORBA způsobem obvyklým ve světě architektury CORBA, tj. deklarovat rozhraní distribuovaných komponent v popisném jazyce IDL (editor kódu nyní podporuje zvýrazňování klíčových slov jazyka OMG IDL), dále vygenerovat pro každé rozhraní v oddělených souborech vazební objekty (stub a skeleton) a vygenerovat kostru třídy implementující distribuovanou komponentu. I nadále je CORBA v Delphi implementována na dynamických knihovnách produktu VisiBroker for C++, které jsou tenkou vrstvou zpřístupněny prostředí Delphi a jazyku Object Pascal. Možnosti, které má vývojář k dispozici, jsou však daleko větší než v předchozích verzích – s Delphi 6 můžeme vytvářet plnohodnotné CORBA servery, generovat uživatelské výjimky na straně serveru, používat vícenásobnou dědičnost rozhraní, používat CORBA typy array a sequence atd. Vývoj CORBA serverů i klientů usnadňují též k tomuto účelu navržené nové průvodci.

A na samotný závěr už jen poslední a pro některé uživatele nepřijemná novinka – oproti předchozím verzím bude instalační CD s Delphi chráněno proti kopírování ochranou SafeDisc firmy Macrovision Corporation a instalační disk bude v periodických intervalech v okamžiku spuštění Delphi požadován ke kontrole.

Jindřich Zelený