

# Obsah

## Kapitola 1

# Architektura aplikací

**1**

## 1.1 Komponentový software

**1**

|                                                   |    |
|---------------------------------------------------|----|
| 1.1.1 Dynamicky linkované knihovny                | 3  |
| 1.1.2 Co získáme použitím komponentového softwaru | 5  |
| 1.1.3 Historie a vývoj                            | 8  |
| 1.1.4 Od COM ke COM+                              | 9  |
| 1.1.4.1 Windows DNA                               | 13 |

## 1.2 Rozhraní čili Interface

**16**

|                                                    |    |
|----------------------------------------------------|----|
| 1.2.1 Deklarace rozhraní                           | 17 |
| 1.2.2 Skutečné COM rozhraní                        | 20 |
| 1.2.3 Jazyk IDL                                    | 21 |
| 1.2.4 Přehled jazyka IDL                           | 22 |
| 1.2.5 Rozhraní IUnknown                            | 24 |
| 1.2.5.1 Počítání referencí a životnost             | 28 |
| 1.2.5.2 Přístup k rozhraní – QueryInterface        | 33 |
| 1.2.6 Pravidla pro tvorbu a použití QueryInterface | 33 |
| 1.2.6.1 Jednoznačnost rozhraní IUnknown            | 33 |
| 1.2.6.2 Objekty mají pevnou množinu rozhraní       | 34 |
| 1.2.6.3 Reflexe metody QueryInterface              | 35 |
| 1.2.6.4 Symetrie metody QueryInterface             | 35 |
| 1.2.6.5 Tranzitivnost metody QueryInterface        | 35 |
| 1.2.7 První komponentová aplikace                  | 36 |
| 1.2.7.1 Klient v jazyce C++                        | 36 |
| 1.2.7.2 Založení instance komponenty v jazyce C++  | 37 |
| 1.2.7.3 Klient v jazyce Visual Basic               | 41 |
| 1.2.7.4 Klient ve skriptovacím jazyce              | 43 |
| 1.2.7.5 Klient v jazyce Java                       | 45 |
| 1.2.8 První komponenta                             | 50 |
| 1.2.8.1 Komponenta v jazyce C++                    | 50 |
| 1.2.8.2 Metoda AddRef                              | 53 |
| 1.2.8.3 Metoda Release                             | 53 |
| 1.2.8.4 Metoda QueryInterface                      | 54 |
| 1.2.8.5 Metody rozhraní IGenerator                 | 55 |
| 1.2.8.6 Class Factory                              | 56 |

|                                                               |            |
|---------------------------------------------------------------|------------|
| 1.2.8.7 Rozhraní IClassFactory                                | 58         |
| 1.2.8.8 Použití CoCreateInstance a CoGetClassObject           | 59         |
| 1.2.8.9 Implementace třídy Class Factory                      | 60         |
| 1.2.8.10 Exportované funkce in-process (DLL) serveru          | 63         |
| 1.2.8.11 Shrnutí založení komponenty v paměti (in-process)    | 65         |
| 1.2.8.12 Dokončení projektu komponenty                        | 67         |
| 1.2.9 Registrace komponent                                    | 71         |
| 1.2.9.1 Základní informace v Registry                         | 71         |
| 1.2.9.2 Další informace v registry                            | 71         |
| 1.2.9.3 ProgID komponenty                                     | 72         |
| 1.2.9.4 Jak použít ProgID v programovacích jazycích           | 73         |
| 1.2.9.5 Způsoby registrace komponent                          | 74         |
| 1.2.9.6 Registrace serverů pomocí registračního souboru       | 74         |
| 1.2.9.7 Automatická registrace in-process serverů             | 75         |
| 1.2.10 Dědičnost, polymorfismus a opakované použití komponent | 78         |
| 1.2.10.1 Containment a Aggregation                            | 81         |
| <b>1.3 Technologie ve vývojových prostředích</b>              | <b>91</b>  |
| 1.3.1 Univerzální propojení – Type Library                    | 92         |
| 1.3.1.1 Vytvoření typové knihovny                             | 92         |
| 1.3.1.2 Distribuce typové knihovny                            | 94         |
| 1.3.1.3 Registrace typové knihovny                            | 94         |
| 1.3.1.4 Datové typy v IDL                                     | 98         |
| 1.3.2 Komponenty v jazyce Visual Basic                        | 98         |
| 1.3.2.1 Návrh a implementace rozhraní                         | 102        |
| 1.3.3 Komponenty v jazyce Visual J++                          | 105        |
| 1.3.3.1 Jak začít?                                            | 105        |
| 1.3.3.2 Začínáme od nuly                                      | 105        |
| 1.3.3.3 Začínáme s typovou knihovnou                          | 107        |
| 1.3.4 Komponenty v jazyce Visual C++                          | 110        |
| 1.3.4.1 Použití typové knihovny                               | 110        |
| 1.3.4.2 Chytrý ukazatel (Smart pointer)                       | 112        |
| 1.3.4.3 Podpora třídy chytrých ukazatelů v jazyce C++         | 112        |
| <b>1.4 Apartmenty, místa kde komponenty žijí</b>              | <b>115</b> |
| 1.4.1.1 Krátká exkurze do threadového modelu ve Win32         | 116        |
| 1.4.1.2 COM thready                                           | 118        |
| 1.4.1.3 Typy Apartmentů                                       | 118        |
| 1.4.2 Třídy komponent a apartmenty                            | 119        |
| 1.4.3 STA apartment                                           | 122        |
| 1.4.4 MTA apartment                                           | 130        |
| 1.4.5 Threadový model Both in-process komponent               | 135        |
| 1.4.6 Free-Threaded Marshaler                                 | 136        |
| 1.4.7 Problémy s nasazením FTM                                | 139        |
| 1.4.8 Global Interface Table                                  | 141        |
| 1.4.9 Neutrální apartment                                     | 143        |
| 1.4.10 Vyberte si threadový model                             | 144        |
| 1.4.11 Podpora apartmentů v jazyce Visual Basic               | 145        |

**2.1 Automation****149**

|                                                   |     |
|---------------------------------------------------|-----|
| 2.1.1 Technologie automation komponent            | 150 |
| 2.1.2 Vše co potřebuje je rozhraní IDispatch      | 150 |
| 2.1.2.1 Duální rozhraní                           | 152 |
| 2.1.2.2 Datové typy Automation                    | 153 |
| 2.1.2.3 Datový typ BSTR                           | 154 |
| 2.1.2.4 Datový typ SAFEARRAY                      | 155 |
| 2.1.2.5 Datový typ Variant                        | 157 |
| 2.1.2.6 Návrh rozhraní IDispatch v komponentě     | 160 |
| 2.1.2.7 Implementace rozhraní IDispatch           | 161 |
| 2.1.2.8 Properties rozhraní                       | 166 |
| 2.1.2.9 Automation klient v jazyce Visual Basic   | 167 |
| 2.1.2.10 Automation klient ve skriptovacím jazyce | 169 |
| 2.1.2.11 Automation klient v jazyce C++           | 170 |

**2.2 Detekce chyb****172**

|                                                          |     |
|----------------------------------------------------------|-----|
| 2.2.1 Základní chybové kódy                              | 173 |
| 2.2.2 Dekódování HRESULT kódu v jazyce C++               | 175 |
| 2.2.3 Rozšířené chybové informace                        | 175 |
| 2.2.4 Generování chyb v komponentách Visual Basic        | 177 |
| 2.2.5 Zpracování chybových objektů v jazyce C++          | 177 |
| 2.2.6 Zpracování chybových objektů v jazyce Visual Basic | 179 |

**2.3 Connection Points****179**

|                                                             |     |
|-------------------------------------------------------------|-----|
| 2.3.1 Myšlenka Connection Points                            | 180 |
| 2.3.2 Implementace Connection Points                        | 181 |
| 2.3.3 Rozhraní IConnectionPointContainer a IConnectionPoint | 182 |
| 2.3.4 Klient objektu Connectable                            | 187 |
| 2.3.5 Klientská aplikace v jazyce Visual Basic              | 189 |
| 2.3.6 Klientská aplikace v jazyce Visual J++                | 189 |

**2.4 Persistence****191**

|                                                    |     |
|----------------------------------------------------|-----|
| 2.4.1 Rozhraní pro podporu persistence             | 191 |
| 2.4.2 Rozhraní IStream                             | 192 |
| 2.4.3 Rozhraní IPersistStream a IPersistStreamInit | 193 |
| 2.4.4 Klient v jazyce C++                          | 195 |
| 2.4.5 Persistence v jazyce Visual Basic            | 198 |
| 2.4.6 Klientská aplikace v jazyce VB               | 200 |
| 2.4.7 C++ klient pro komponentu Visual Basic       | 201 |

**2.5 Monikery****203**

|                                                 |     |
|-------------------------------------------------|-----|
| 2.5.1 Inicializace objektu a aktivační rozhraní | 204 |
| 2.5.2 Class objekt                              | 205 |
| 2.5.2.1 Uživatelská aktivace objektu            | 205 |

|                                                      |     |
|------------------------------------------------------|-----|
| 2.5.3 Aktivace monikerem                             | 206 |
| 2.5.3.1 Architektura pojmenování a vytváření objektů | 207 |
| 2.5.4 Class Moniker                                  | 211 |
| 2.5.5 New Moniker                                    | 212 |
| 2.5.6 Running Object Table                           | 213 |

## Kapitola 3

# Od lokálních ke vzdáleným komponentám 217

|                                                       |            |
|-------------------------------------------------------|------------|
| <b>3.1 Marshaling</b>                                 | <b>218</b> |
| 3.1.1 Marshalování s typovou knihovnou                | 219        |
| 3.1.2 Standardní marshalování                         | 221        |
| 3.1.3 Custom marshaling                               | 225        |
| <b>3.2 Surrogates</b>                                 | <b>225</b> |
| 3.2.1 ApplD – další klíč v Registry                   | 226        |
| <b>3.3 EXE komponenty</b>                             | <b>227</b> |
| 3.3.1 Odlišný vstupní bod                             | 228        |
| 3.3.2 Registrace komponent                            | 228        |
| 3.3.3 Přístup ke Class Factory                        | 230        |
| 3.3.4 Řízení životnosti komponenty                    | 232        |
| 3.3.5 Uspaná komponenta                               | 234        |
| 3.3.6 Singleton                                       | 235        |
| <b>3.4 Vzdálené spouštění</b>                         | <b>236</b> |
| 3.4.1 Utilita DCOMCNFG                                | 237        |
| <b>3.5 Bezpečnost</b>                                 | <b>240</b> |
| 3.5.1 Principy bezpečnosti v COM+                     | 240        |
| 3.5.1.1 Access security – ochrana objektu komponenty  | 241        |
| 3.5.1.2 Launch Security – ochrana vzdáleného počítače | 241        |
| 3.5.1.3 Security Identity – bezpečnost objektu        | 241        |
| 3.5.1.4 Connection Policy – „kdopak to mluví?“        | 242        |
| 3.5.1.5 Autentikace – identifikace uživatele          | 242        |
| 3.5.1.6 Delegation a Impersonation                    | 242        |
| 3.5.2 Deklarativní bezpečnost                         | 243        |
| 3.5.2.1 Nastavení implicitní bezpečnosti počítače     | 243        |
| 3.5.2.2 Konfigurace bezpečnosti komponenty            | 246        |
| 3.5.3 Programový přístup                              | 250        |
| 3.5.4 Bezpečnost v prostředí Windows 95 a 98          | 251        |

**4.1 Atributové programování a kontext****254**

|                                           |     |
|-------------------------------------------|-----|
| 4.1.1 Konfigurované komponenty            | 255 |
| 4.1.2 Balíčky a aplikace                  | 257 |
| 4.1.3 Intercepce                          | 259 |
| 4.1.4 Kontext                             | 260 |
| 4.1.5 Relativita kontextu                 | 265 |
| 4.1.6 Kontext, apartment a multithreading | 266 |
| 4.1.7 Aktivita                            | 268 |

**4.2 Transakce****271**

|                                           |     |
|-------------------------------------------|-----|
| 4.2.1 Transakce a její vlastnosti         | 272 |
| 4.2.2 Implementace transakcí v MTS a COM+ | 273 |
| 4.2.2.1 Deklarace transakcí               | 273 |
| 4.2.3 Řízení transakcí                    | 273 |
| 4.2.4 Just-in-Time Aktivace               | 279 |
| 4.2.5 Object Pooling                      | 280 |

**4.3 Queued Components****283**

|                                                  |     |
|--------------------------------------------------|-----|
| 4.3.1 Oblasti nasazení Queued Component aplikací | 283 |
| 4.3.2 Architektura QC                            | 283 |
| 4.3.3 Vývoj komponent technologie QC             | 285 |
| 4.3.3.1 Návrh rozhraní                           | 286 |
| 4.3.3.2 Získání výsledků                         | 287 |
| 4.3.4 Instalace QC komponent v systému           | 288 |
| 4.3.5 Aktivace QC komponent                      | 290 |
| 4.3.6 Podpora transakcí s QC                     | 290 |
| 4.3.6.1 Interní transakce                        | 291 |
| 4.3.6.2 Externí transakce                        | 292 |

**4.4 COM+ události****293**

|                                      |     |
|--------------------------------------|-----|
| 4.4.1.1 Pooling                      | 294 |
| 4.4.1.2 Tightly-coupled event        | 294 |
| 4.4.1.3 Loosely-coupled events (LCE) | 294 |
| 4.4.2 Architektura COM+ událostí     | 295 |
| 4.4.3 Příklady aplikací              | 295 |
| 4.4.4 Filtrace událostí              | 297 |
| 4.4.4.1 Subscriber filtering         | 302 |
| 4.4.4.2 Publisher filtering          | 303 |