

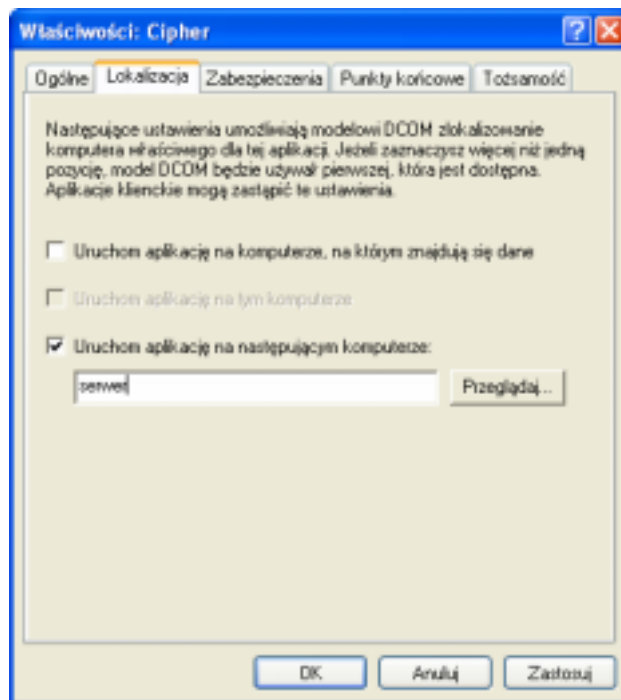
ĆWICZENIE 4. DCOM (DISTRIBUTED COM)

DCOM rozszerza COM o możliwość wykorzystywania i udostępniania komponentów poprzez sieć komputerową.

TWORZENIE KOMPONENTÓW NA KOMPUTERZE ZDALNYM

Najprostszą metodą uruchomienia komponentu na komputerze zdalnym jest określenie domyślnego komputera, na którym ma być on uruchamiany (domyślnie jest to komputer lokalny). Informacja ta przechowywana jest w rejestrze systemowym. Zmian najprościej jest dokonać z wykorzystaniem narzędzia konfiguracji DCOM – DCOMCNFG.EXE.

Wykorzystanie komponentów uruchamianych na komputerach zdalnych odbywa się na dokładnie tej samej zasadzie jak komponentów na komputerze lokalnym – jest więc całkowicie przezroczyste dla aplikacji. Biblioteka COM na podstawie wpisów w rejestrze tworzy komponent lokalnie bądź zdalnie. Dzięki temu z zalet DCOM mogą korzystać nie tylko te aplikacje, które zostały napisane z myślą o tej technologii.



OKREŚLANIE KOMPUTERA, NA KTÓRYM MA BYĆ UTWORZONY KOMPONENT

W przypadku, gdy w aplikacji konieczne jest określenie komputera, na którym ma być utworzony komponent, opisany powyżej mechanizm staje się niewystarczający.

Biblioteka COM udostępnia funkcje tworzące komponenty, które jako jeden z argumentów przyjmują identyfikator komputera, na którym komponent ma zostać utworzony.

Jedną z takich funkcji jest *CoCreateInstanceEx*. Funkcja ta, prócz określania nazwy komputera, pozwala na pobieranie wielu interfejsów w jednym wywołaniu. Jest to szczególnie ważne w środowisku sieciowym, w którym każde wywołanie funkcji zabiera cenny czas.

```
MULTI_QI mqi[]={ {IID_IInterfejs1, NULL, 0}, {IID_IInterfejs2, NULL, 0} };
```

```
COSERVERINFO srvinfo = {0, L"nazwaserwera", NULL, 0};
```

```
// Otworzenie obiektu i odpytanie o dwa interfejsy
```

```
HRESULT hr=CoCreateInstanceEx(
```

```
    CLSID_Komponent,
```

```
    NULL,
```

```
    CLSCTX_SERVER,
```

```
    &srvinfo,
```

```
    sizeof(mqi)/sizeof(mqi[0]), // liczba interfejsow (2)
```

```
    mqi);
```

```
    // interfejsy
```

```
if (SUCCEEDED(hr)) {
```

```
    if (SUCCEEDED(mqi[0].hr)) {
```

```
        IInterfejs1* i1= (Interfejs1*) mqi[0].pItf;
```

```
        i1->Metoda();
```

```
        // użycie interfejsu
```

```
        i1->Release();
```

```
        // zwolnienie interfejsu
```

```
    } // analogicznie dla I2...
```

```
}
```

Wykorzystanie *CoCreateInstanceEx* do utworzenia komponentu na komputerze zdalnym

UDOSTĘPNIANIE KOMPONENTÓW PRZEZ SIEĆ

Technologia DCOM pozwala na wykorzystanie w sieci lokalnej zarówno komponentów wewnątrz jak i zewnątrzprocesowych. Aby możliwe było ich użycie konieczna jest rejestracja zarówno po stronie klienta jak i niektórych serwera. W niektórych przypadkach może być konieczna rejestracja bibliotek DLL *proxy* i *stub* udostępnianych komponentów. Składniki wewnątrzprocesowe oraz biblioteki *proxy* i *stub* mogą zostać zarejestrowane z wykorzystaniem narzędzia REGSVR32.EXE. Komponenty zewnątrzprocesowe rejestrowane są po uruchomieniu pliku EXE serwera z argumentem „/RegServer”.

Prawa dostępu do komponentów oraz tryb ich uruchamiania mogą być ustawiane przy pomocy narzędzia konfiguracji DCOM – DCOMCNFG.EXE. W zależności od konfiguracji komponenty mogą być uruchamiane z wykorzystaniem zawsze tego samego, konkretnego konta, konta systemowego (w przypadku serwisów), oraz z wykorzystaniem konta użytkownika, na którego rzecz tworzony jest komponent (domyślnie).

TWORZENIE SERWISÓW UDOSTĘPNIAJĄCYCH KOMPONENTY

Biblioteka ATL pozwala na tworzenie serwisów, które są serwerami COM. Mogą być one uruchamiane na koncie systemowym automatycznie podczas startu systemu. Aby stworzyć taki serwis należy w kreatorze projektu ATL wybrać „Service (EXE)”.

Po utworzeniu komponentów i skompilowaniu serwis może być uruchamiany jak najzwyczajniejszy zewnątrzprocesowy serwer COM – ułatwia to bardzo proces uruchamiania. Aby serwer COM pracował jako serwis i był widoczny w Panelu Sterowania, konieczne jest jednak jego zainstalowanie. Dokonuje się tego poprzez uruchomienie wygenerowanego pliku EXE z parametrem „/Service”.

WSPÓLDZIELENIE KOMPONENTU PRZEZ WIELU KLIENTÓW JEDNOCZEŚNIE

Często zachodzi potrzeba stworzenia komponentu, który mógłby być wykorzystany przez wielu klientów jednocześnie (np. do celów wymiany informacji). Stworzenie tego komponentu wiąże się z:

1. Taką konfiguracją serwer COM udostępniającej komponent, żeby uruchamiała się z wykorzystaniem zawsze tego samego konta użytkownika (może być zrealizowana w formie serwisu systemowego, ale nie jest to konieczne) – w ten sposób każdy klient połączy się z tą samą instancją serwera COM.
2. Stworzenie komponentu będącego singletonem, czyli takiego, który w ramach aplikacji (serwera COM) ma dokładnie jedną instancję (w bibliotece ATL singletony tworzy się z wykorzystaniem makrodefinicji DECLARE_CLASSFACTORY_SINGLETON) – zagwarantuje to, że każdy klient wykorzysta dokładnie ten sam komponent.

PROGRAMY PRZYKŁADOWE

Zestaw programów przykładowych zawiera:

- Zewnątrzprocesowy serwis COM, który udostępnia komponent pozwalający na pobranie nazwy komputera na którym jest uruchomiony.
- Aplikacje klienckie pozwalające na wykorzystanie komponentu (krótkie programy w C++/JScript).

PRZEBIEG LABORATORIUM

1. Zapoznać się z narzędziem DCOMCNFG.EXE.
2. Zapoznać się z programami przykładowymi (komponentem i klientami).
3. Uruchomić programy przykładowe w sieci lokalnej z wykorzystaniem dwóch komputerów.
4. Wykonać zlecone przez prowadzącego zadanie.

ZADANIA PRZYKŁADOWE

1. Rozproszona aplikacja umożliwiająca przeprowadzenie rozmów (chat). Aplikacja powinna się składać z komponentu implementującego funkcję serwera uruchamianego na jednym z komputerów oraz aplikacji klienckich łączących się z serwerem, umożliwiających wysyłanie i odbieranie wiadomości.
2. System umożliwiający monitorowanie ilości wolnego miejsca na dyskach twardych komputerów. Aplikacja powinna być złożona z komponentów uruchamianych na komputerach oraz z klienta umożliwiającego monitorowanie podanych przez użytkownika komputerów.

Sprawozdanie powinno zawierać dokumentację utworzonego oprogramowania oraz opis sposobu jego uruchomienia w środowisku sieciowym.

Opracowanie: Wojciech Pytlik