

KỸ THUẬT VỀ CÁC MÔ HÌNH LẬP TRÌNH ỨNG DỤNG, CÁC KIẾN TRÚC COM/DCOM VÀ CÁCH HIỆN THỰC ỨNG DỤNG DỰA TRÊN CÁC KIẾN TRÚC NÀY

I. Giới thiệu mô hình lập trình ứng dụng:

1. Giới thiệu mô hình client/server :

Mô hình client/server là mô hình có ảnh hưởng lớn nhất đến ngành công nghệ thông tin. Mô hình client/server giúp cho những máy tính riêng lẻ, khả năng xử lý thấp có thể chạy được những ứng dụng rất lớn, phức tạp. Mô hình client/server giúp phát triển một phương thức phát triển ứng dụng mới là việc phân chia ứng dụng thành nhiều lớp thực hiện những chức năng chuyên biệt. Một ứng dụng thông thường được chia làm 3 lớp:

- o Giao diện (Presentation logic): lớp này là cầu nối giữa người dùng với ứng dụng, cung cấp những chức năng ứng dụng cho người dùng và nhận lệnh từ người dùng cho ứng dụng. Lớp này được thiết kế sao cho càng thân thiện với người dùng càng tốt.
- o Chức năng (Business logic): đây là phần lõi của một chương trình, cung cấp tất cả những chức năng có thể có của chương trình cho lớp giao diện bên trên.
- o CSDL (Data Access logic): lớp này cung cấp khả năng truy xuất đến CSDL cho lớp chức năng nếu cần.

Việc phân chia ứng dụng thành nhiều lớp còn giúp cho ứng dụng trở nên dễ dàng thay đổi, cập nhật và đáng tin cậy hơn.

2. Mô hình hai mức (2 – tier) client/server:

Ở mô hình này khi client yêu cầu một thông tin gì đó thì client sẽ gửi yêu cầu đến cho server, server xử lý các yêu cầu từ client gửi về đúng những thông tin mà client cần.

Mô hình này được chia làm hai loại:

- o Tính toán tập trung ở client hay còn gọi là fat client – thin server: ở loại này, phía client phải đảm nhận cả 2 lớp là giao diện và chức năng, còn server chỉ đảm nhận nhiệm vụ của lớp CSDL.

Ưu điểm của mô hình này là đơn giản, thích hợp cho những ứng dụng nhỏ. Tuy nhiên nó cũng có một số nhược điểm sau:

- + Do tính toán nghiêng về phía client nhiều nên đòi hỏi client phải có cấu hình đủ mạnh.
- + Do lớp chức năng của chương trình nằm ở phía client nên khi chương trình cần nâng cấp thì sẽ rất khó khăn vì phải cập nhật lại chương trình nằm ở toàn bộ các client.
- + Do mọi thao tác nằm trên client nên thao tác yêu cầu dữ liệu giữa client và server nhiều dẫn đến tốc độ chương trình chậm.

- o Tính toán tập trung ở server (fat server – thin client): ở loại này, client đảm nhận phần giao diện còn server thực hiện nhiệm vụ lớp chức năng và lớp CSDL.

Ngược lại với loại trên, loại này giảm sự lưu thông giữa client và server. Tuy nhiên do công việc tập trung nhiều tại server nên đòi hỏi cấu hình server đủ mạnh, đặc biệt là khi có nhiều client truy xuất cùng một lúc.

3. Mô hình ba mức (3 – tier) client/server:

Trong mô hình 3 mức, lớp chức năng của chương trình được tách ra thành một mức tạo thành 3 mức riêng biệt. Việc tách lớp này làm cho các phần của chương trình độc lập hơn, đáng tin cậy hơn, chương trình trở nên linh động hơn trong việc thay thế, nâng cấp và do đó mô hình này rất thích hợp với những ứng dụng có yêu cầu thay đổi thường xuyên.

II. Giới thiệu sơ lược về COM:

COM (Component Object Model) là một cách thức để viết các thành phần phần mềm (software component) cung cấp các chức năng phục vụ cho các ứng dụng, các thành phần khác. Viết một thành phần COM cũng như viết một hàm API động, hướng đối tượng. Các thành phần COM có thể được nối kết với các ứng dụng cũng như các thành phần COM khác.

a. COM là gì ?

COM chỉ là một sự đặt tả. Nó chỉ ra làm thế nào để xây dựng các thành phần có thể thay thế một cách “động”. COM cung cấp một “chuẩn” để các ứng dụng và các thành phần phải tuân theo để chúng có thể hoạt động được với nhau.

b. Thành phần COM là gì ?

Thành phần COM là những đoạn mã hiện thực các hàm, các chức năng được chứa trong file .DLL hoặc .EXE. Thành phần COM cần phải thỏa mãn được những “chuẩn” của COM. Các thành phần COM có thể được liên kết động.

Các thành phần COM có thể được đóng bao vì nó thỏa mãn các ràng buộc:

- o Thành phần COM hoàn toàn độc lập với ngôn ngữ hiện thực nó. Thành phần COM có thể được hiện thực bằng nhiều ngôn ngữ như Visual C++, Visual Basic, Java
- o Thành phần COM được sử dụng dưới dạng mã nhị phân.
- o Thành phần COM có thể được nâng cấp mà không làm hỏng chương trình đang chạy.
- o Thành phần COM trong suốt về vị trí.

c. Giao diện (Interface)

Giao diện là tất cả đối với COM. Các ứng dụng chỉ có thể nhìn thấy và truy xuất đến các thành phần COM thông qua giao diện của nó.

Giao diện làm tăng mức độ độc lập giữa ứng dụng và các thành phần COM. nhờ có giao diện mà một thành phần COM có thể được thay thế động mà không ảnh hưởng đến ứng dụng đang chạy.

Giao diện không bao giờ thay đổi. Nếu muốn nâng cấp một giao diện thì giao diện mới phải tồn tại song song với giao diện cũ.

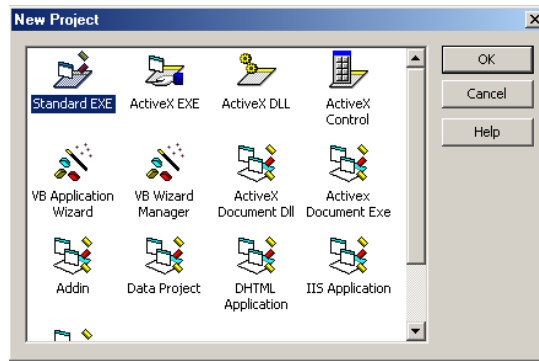
d. COM trong Windows:

Trong Windows, các thành phần COM cũng cũng như giao diện của COM đều được gán một ID. ID của thành phần COM được gọi là CLSID(class identifier); của giao diện COM là IID(interface identifier). Các ID này được gọi chung là GUID(globally unique identifier) là một cấu trúc 16 byte và được xác định duy nhất trên toàn thế giới. Tất cả các ID này đều được chứa trong registry của windows cùng với thông tin về các thành phần hoặc giao diện mà nó đại diện.

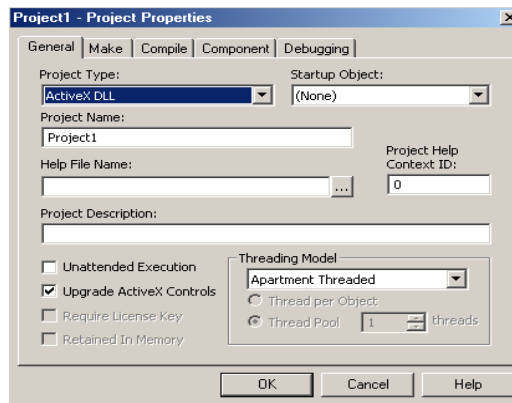
Khi ứng dụng muốn sử dụng COM, nó sẽ gọi một hàm API của windows là CoCreateInstance với thông số truyền vào là CLSID và IID tương ứng với thành phần và giao diện mà thành phần đó hỗ trợ để có được giao diện mong muốn.

e. Cách tạo COM(.dll) và COM(.exe) trong Visual Basic

Khi tạo một project mới trong visual basic, ta có thể chọn các dạng mẫu có sẵn để tạo một project mới. Để tạo thành phần COM ta có hai lựa chọn dạng COM(.dll) hoặc dạng COM(.exe) trong hộp thoại New Project tùy theo ứng dụng của chúng ta.



Khi tạo một dự án mới dạng ActiveX (.dll) hay ActiveX (.exe) Visual Basic sẽ tự động thiết lập các thuộc tính mặc định cho dự án đó. Ta có thể thay đổi những thuộc tính đó để chương trình thực hiện tốt hơn. Sau đây là các thuộc tính cơ bản của một dự án thành phần COM.



+ *Startup Object* : Startup Object cho phép ta chỉ định liệu code có được thực thi khi thành phần được load lên hay không. Đa số thì ta thiết lập thuộc tính này là None, code được thực thi khi visual basic tạo một thể hiện của class.

+ *Project Name* : là tên của dự án mà ta đang tạo. Tên này cũng là ID của thành phần COM trong windows registry. Tên của dự án kết hợp với tên của class tạo nên ProgID. Client sẽ dùng tên này để tạo một đối tượng COM.

+ *Threading Model (ActiveX .dll)*: Trong dự án ActiveX (.dll), Thread Model cho phép ta lựa chọn Single Threaded hoặc Apartment Threaded. Giá trị mặc định là Apartment Threaded.

- Apartment Threaded về hình thức thực hiện chức năng multithreading giống như Windows 95, Windows 98, and Windows NT. Đa phần các thành phần COM thực hiện với Visual Basic đều thực hiện với Apartment Threaded.

- Single Threaded: khi lựa chọn dạng này thì mọi client sẽ chạy trên một thread duy nhất, khi một client gửi một yêu cầu tạo thread đến, mà thread đang phục vụ cho một client khác thì client vừa gửi yêu cầu phải đợi cho đến khi thread phục vụ xong client kia mới đáp ứng lời gọi của mình. Với dạng này thì lợi điểm là dễ kiểm soát việc tạo thread, nhưng ngược lại khả năng phục vụ kém.

- Những lợi điểm của sự lựa chọn Apartment Threaded.

. Xử lý được vấn đề có nhiều yêu cầu tạo thread từ phía nhiều client. Tự động quản lý việc cung cấp thread phục vụ cho nhiều client, nhưng tại một thời điểm chỉ có một client được phục vụ.

+ *Threading Model (ActiveX .exe)*: Trong dự án ActiveX Exe ta có những lựa chọn cho Threading Model như sau:

- Chỉ có một thread phục vụ: Chỉ cho tất cả client dùng chung một thread
- Chỉ định một con số cụ thể thread dùng chung, và số thread này được quản lý để phục vụ cho nhiều client theo phương pháp round-robin
- Mọi yêu cầu tạo phục vụ từ nhiều client sẽ được đáp ứng bằng cách tạo cho mỗi client một thread mới. Điều này gây khó khăn cho việc quản lý các đối tượng do client tạo ra.

f. Cách đăng ký, gỡ bỏ đăng ký COM(.dll) và COM(.exe) trong Windows

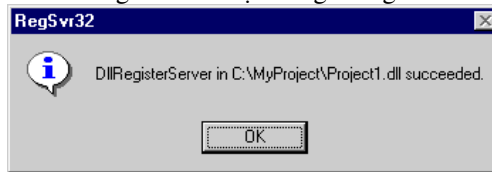
Sau khi biên dịch một dự án COM thành một file dạng dll hay dạng exe ta phải đăng ký các thành phần COM này để có thể sử dụng được có nhiều cách để đăng ký sau đây là cách tiêu biểu để đăng ký:

+ *Cách đăng ký COM DLL* : Để đăng ký được thành phần COM (.dll) ta thực hiện tuần tự các bước sau:

- o Nhấn vào nút *start* của Windows
- o Trên *Start* menu, click *Run*.
- o Trong *Run* dialog box, ta gõ vào như sau:

Regsvr32.exe <DLLPath>\<DLL name>.dll

- o Click vào nút OK. Nếu đăng ký thành công sẽ hiển thị dialog thông báo sau



Cách gỡ bỏ đăng ký của thành phần COM (.dll) cũng thực hiện các bước như trên, nhưng trong hộp thoại *Run* ta gõ vào như sau:

Regsvr32.exe /u <DLLPath>\<DLL name>.dll

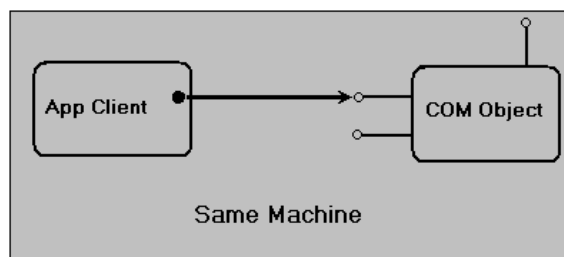
+ *Cách đăng ký COM (.exe)* :

Cách đăng ký hay gỡ đăng ký của COM Exe cũng thực hiện các bước tương tự như việc đăng ký COM Dll. Nhưng trong hộp thoại *Run* ta gõ vào như sau

- . <Đường dẫn của file exe đã biên dịch>\<tên file Exe> /regserver trong trường hợp muốn đăng ký.
- . <Đường dẫn của file exe đã biên dịch>\<tên file Exe> /unregserver trong trường hợp muốn gỡ bỏ đăng ký.

III. Các kiến trúc của COM/DCOM:

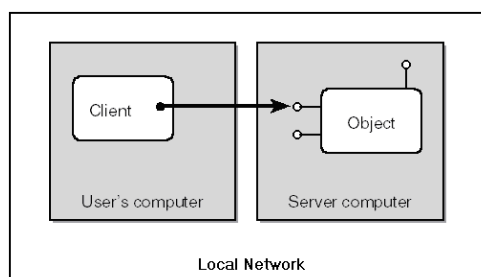
1. Ứng dụng Client và ứng dụng COM chạy trên cùng một máy:



Mục đích xây dựng thành phần đối với mô hình này là :

- . Ứng dụng COM có thể được tái sử dụng cho nhiều ứng dụng Client khác nhau.
- . Làm cho chương trình dễ viết, dễ kiểm tra lỗi, nâng cấp mà không ảnh hưởng đến những ứng dụng Client đang chạy nó.

2. Ứng dụng Client chạy một máy, Ứng dụng COM chạy một máy. Nhưng hai máy này cùng một mạng local.



Lúc này thành phần COM được gọi là DCOM. Có 3 cách để giao tiếp giữa ứng dụng Client và Ứng dụng COM:

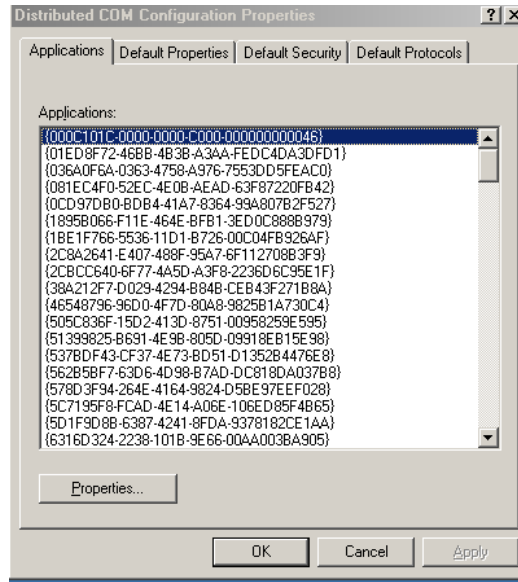
- ActiveX (.exe)
- Dùng MTS (Microsoft Transaction Server) để triển khai ActiveX DLL từ xa
- RDS (Remote Data Access)

a. Giới thiệu về ActiveX (.exe)

Cũng là một ứng dụng COM nhưng được xây dựng cho phép chia sẻ ứng dụng cho nhiều ứng dụng Client ở các máy khác nhau (trong cùng một mạng) sử dụng. ActiveX (.Exe) chính là hình thức dạng DCOM.

Sau khi xây dựng và biên dịch một ứng dụng COM dạng ActiveX (.exe) để những ứng dụng Client có thể liên kết và sử dụng được ta phải chạy tập tin Dcomcnfg.exe của Windows để cấu hình cho ứng dụng COM này.

❖ Giới thiệu Dcomcnfg.exe để cấu hình ứng dụng COM (ActiveX Exe)



Trong giao diện chính của DCOM Config có 3 tabs chính sau đây:

- Applications.
- Default Properties.
- Default Security.

+ Applications Tab:

Applications Tab hiển thị những đối tượng đã đăng ký vào trong khóa registry:

HKEY_CLASSES_ROOT\AppId\

Mỗi đối tượng được hiển thị trong Applications Tab, đặt tính của mỗi ứng dụng có thể được hiển thị bằng cách chọn đối tượng và nhấn vào nút "Properties..." hay nhấn đúp vào đối tượng cần xem.

+ Default Properties Tab:

Những giá trị được hiển thị trong Default Properties Tab có thể được tìm thấy trong registry của Windows ở vị trí sau :

HKEY_LOCAL_MACHINE\Software\Microsoft\OLE

Tiêu đề đầu tiên trong Default Properties Tab là một check box :

"Enable Distributed COM on this computer"

khi check box này được đánh dấu sẽ cho phép tạo đối tượng DCOM. Ngược lại không tạo được đối tượng DCOM.

Lưu ý: tất cả những thiết lập có hiệu lực khi ta khởi động lại hệ thống.

Thành phần thứ 2 của Default Properties Tab là Default Distributed COM Communication Properties thành phần này được chia làm 2 mục sau:

- Default Authentication Level (Packet Level)

Mục này có nhiều giá trị chọn như sau:

Name	Description
None	No authentication.
Connect	Authentication occurs when a connection is made to the server. Connectionless protocols do not use this.

Call	The authentication occurs when a RPC call is accepted by the server. Connectionless protocols do not use this.
Packet	Authenticates the data on a per-packet basis. All data is authenticated.
Packet Integrity	This authenticates that the data has come from the client, and checks that the data has not been modified.
Packet Privacy	In addition to the checks made by the other authentication techniques, this encrypts the packet.
Default	May vary depending upon operating system.

- Default Impersonation Level

<i>Name</i>	<i>Description</i>
Anonymous	The client is anonymous. This setting is not currently supported by DCOM.
Identify	The server can impersonate the client to check permissions in the ACL (Access Control List) but cannot access system objects.
Impersonate	The server can impersonate the client and access system objects on the client's behalf.
Delegate	In addition to the Impersonate level, this level can impersonate the client on calls to other servers. This is not supported in the current release of DCOM.

+ Default Security Tab

Có 3 sự lựa chọn ở Default Security tab. Những giá trị của tab này được lưu trữ trong Windows Registry tại vị trí:

HKEY_LOCAL_MACHINE\Software\Microsoft\OLE

❖ Cách cấu hình Visual Basic DCOM dùng Dcomcnfg.exe

Việc cấu hình được thực hiện theo những bước sau:

- Run Dcomcnfg.exe
- Tại Applications Tab chọn ứng dụng DCOM cần cấu hình.
- Nhấn đúp vào ứng dụng DCOM vừa chọn.
- Chọn Security Tab
- Thực hiện những thay đổi sau.

- CUSTOM ACCESS PERMISSIONS:

Everyone	-allow access
System	-allow access
Interactive	-allow access

CUSTOM LAUNCH PERMISSIONS:

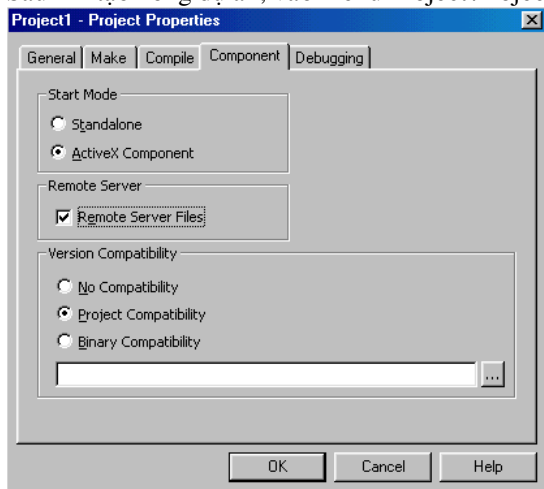
Everyone	-allow launch
System	-allow launch
Interactive	-allow launch

- Chọn Identity Tab, chọn "The Interactive User."
- Nhấn OK, nhấn OK một lần nữa.

❖ **Cách hiện thực một ứng dụng DCOM dạng ActiveX(.exe) dùng Visual Basic.**

◆ **Tạo một dự án DCOM dạng ActiveX(.exe)**

1. Mở Visual Basic tạo một dự án dạng ActiveX EXE
2. Sau khi tạo xong dự án, vào menu Project\Project Properties thiết lập như sau:



- Việc ở Start Mode chọn ActiveX Component sẽ cho phép ta chạy ứng dụng server dưới môi trường phát triển Visual Basic. Nếu không chọn thì ứng dụng server sẽ thoát ra ngay khi ta chạy nó trong Visual Basic.
- Remote Sever đánh dấu vào Remote Server Files sẽ làm cho Visual Basic tạo ra VBR(Visual Basic Registry) và TLB (Type Library) ngay khi ta xây dựng xong ứng dụng server.
- Cuối cùng Version Compatibility chọn Project Compatibility, sau này ta sẽ đổi sang lựa chọn Project Compatibility.

3. Biên dịch ứng dụng server

◆ Trước khi phân bố ứng dụng, ta mở lại ứng dụng server, vào project properties của ứng dụng server thiết lập lại Version Compatibility chọn Binary Compatibility, tại text box dùng browser dẫn tới tập tin .exe của ứng dụng server. Save ứng dụng lại, biên dịch lại ứng dụng server.

◆ Triển khai ứng dụng server để ứng dụng Client có thể giao tiếp được.

Để triển khai ứng dụng phân bố vừa tạo ra ta phải sử dụng công cụ "Package and Deployment Wizard" của Visual Basic Tools bằng cách chọn Start\Program\Microsoft Visual Basic 6.0\Microsoft Visual Basic Tools.

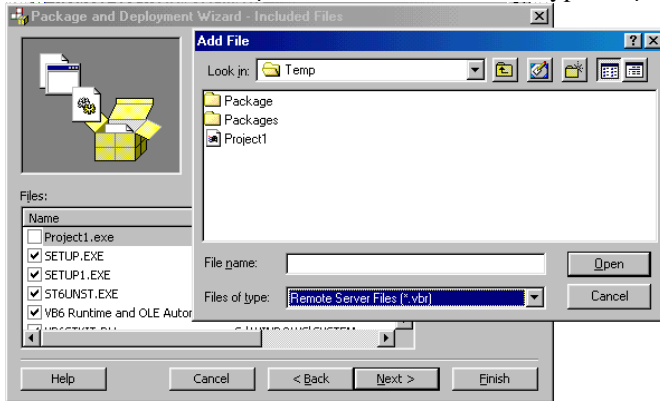
Tiếp theo thực hiện tuần tự theo các bước sau:

1. Dùng Browser chọn dự án server mới vừa tạo, nhấn vào nút Package
2. Chọn Dependency File trong Package Type.
3. Nhấn Next để chuyển qua bước tiếp theo.
4. Nhấn Next tiếp.
5. Chọn No khi được hỏi về hỗ trợ Remote Automation
6. Nhấn Next trên màn hình Included Files.
7. Nhấn Next trên màn hình Cab Information.
8. Nhấn Next trên Install Locations.
9. Nhấn Finish.

◆ **Tạo chương trình Setup cho Client.**

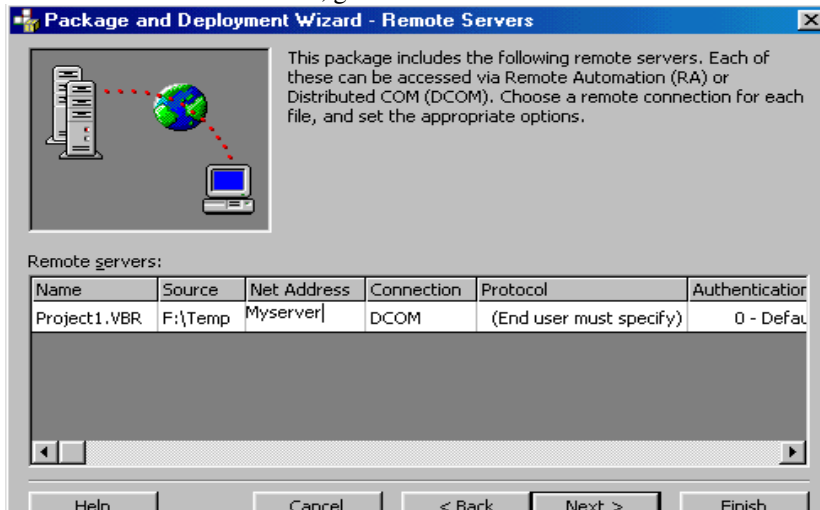
1. Khởi động chương trình Package and Deployment một lần nữa.
2. Nhấn vào nút Package
3. Chọn Standard Setup Package trong danh sách Package Type.
4. Chọn một folder mà nó sẽ được truy cập qua mạng.
5. Trong màn hình Included Files, Bỏ đánh dấu không chọn dự án server(.exe).

6. Nhấn vào nút Add, Chọn VBR ở combo box file type, chọn file .vbr



7. Nhấn Next.

8. Ở màn hình Remote Servers, gõ vào Net Address của Server.



9. Nhấn chọn Next đến hết. Nhấn Finish.

◆ Cài đặt trên máy chạy ứng dụng DCOM và trên máy chạy ứng dụng Client.

+ Chạy chương trình Setup.exe được tạo bởi Package and Deployment Wizard cho ứng dụng server trên máy chạy ứng dụng COM

+ Chạy chương trình Setup.exe được tạo bởi Package and Deployment Wizard cho ứng dụng Client trên máy chạy ứng dụng Client.

b. Dùng MTS để triển khai ActiveX DLL từ xa

Đối với Windows NT thì MTS được tách riêng ra thành một ứng dụng riêng biệt. Muốn sử dụng MTS thì phải cài đặt Option Pack 4 cho Windows NT. Còn đối với Windows 2000 thì MTS được tích hợp vào IIS 5.0, đưa COM và MTS tích hợp lại trong COM+ Applications.

❖ Giới thiệu MTS (Microsoft Transaction Server)

MTS là một dịch vụ trên Windows giống như các dịch vụ khác như IIS, File hay Print ..., có nghĩa là ta có thể chạy (start) hoặc ngừng chạy (stop) chúng khi cần và MTS chạy background để phục vụ cho các ứng dụng cần đến nó

Trong Windows NT 4.0, MTS được coi như là một phần phụ (add-on), được cài thông qua NT4 Option Pack. Còn trong Windows 2000 nó trở thành một phần mặc định của chính hệ điều hành.

Ngoài chức năng chính là quản lý giao dịch, MTS còn quản lý nhiều thứ khác “MTS quản lý cách mà ứng dụng sử dụng các thành phần” hay nói cách khác là MTS quản lý các thành phần giúp cho các ứng dụng sử dụng các thành phần đó được hiệu quả hơn.

Các bước thông thường để sử dụng một thành phần:

1. Tạo một instance của thành phần.
2. Khởi động giá trị của các thuộc tính của thành phần.
3. Sử dụng thành phần.
4. Giải phóng thành phần ra khỏi bộ nhớ.

Thông thường, việc khởi tạo và giải phóng thành phần nằm ở đầu và ở cuối một thủ tục hay chương trình. Nếu sử dụng tốt hơn thì khởi tạo thành phần ngay khi cần đến và giải phóng sau khi chắc chắn không còn cần dùng đến thành phần đó nữa. Tuy nhiên trong khoảng thời gian sử dụng thành phần đó thường có rất nhiều thao tác không sử dụng thành phần hoặc không có thao tác nào sử dụng đến thành phần, nhưng vẫn phải giữ thành phần đó trong bộ nhớ chờ được sử dụng sau ở các thao tác khác

mà không thể giải phóng thành phần vì khi giải phóng thì các trạng thái bên trong của thành phần cũng bị xóa đi. Điều đó tạo nên một sự lãng phí rất lớn tài nguyên của hệ thống.

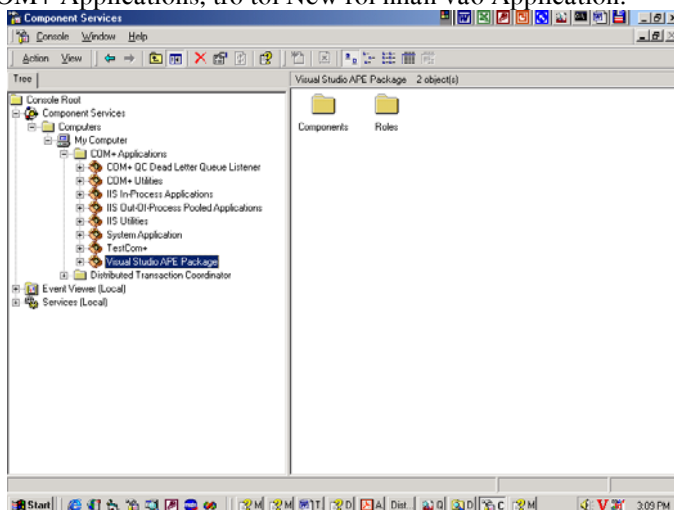
MTS giải quyết vấn đề trên bằng cách: khi một ứng dụng cần một thành phần thì MTS sẽ cung cấp 1 instance của thành phần đó cho ứng dụng. Nếu một ứng dụng thứ 2 muốn sử dụng thành phần đó, trong khi ứng dụng 1 không thực sự sử dụng nó (tuy nhiên ứng dụng thứ 1 vẫn chưa giải phóng thành phần) thì MTS sẽ đưa instance mà ứng dụng thứ 1 đang giữ cho ứng dụng thứ 2 sử dụng. Nếu ứng dụng thứ 1 đột nhiên muốn sử dụng tiếp thành phần thì MTS sẽ tìm trong các instance của thành phần đó có instance nào không thực sự được sử dụng bởi ứng dụng khác không. Nếu có thì MTS sẽ đưa instance đó cho ứng dụng 1 sử dụng. Nếu tất cả các instance của thành phần đó đều đang được sử dụng thì MTS sẽ tạo một instance mới và giao nó cho ứng dụng thứ 1 sử dụng.

Quản lý giao dịch trong MTS: MTS cung cấp môi trường giao dịch cho các thành phần. Các thành phần trong cùng một môi trường sẽ thực hiện một giao dịch, có nghĩa là nếu toàn bộ các thành phần trong môi trường cùng thực thi thành công nhiệm vụ của mình thì giao dịch coi như là thành công. Nếu một trong các thành phần vì lý do nào đó không thể thực thi nhiệm vụ của mình thì toàn bộ các công việc của các thành phần khác sẽ bị hủy bỏ và hệ thống sẽ trở lại trạng thái ban đầu, trạng thái trước khi giao dịch được thực hiện.

❖ **Các bước để triển khai một thành phần từ xa trong Windows 2000 dùng MTS :**

◆ **Các bước thực hiện trên máy chạy thành phần COM DLL muốn triển khai từ xa.**

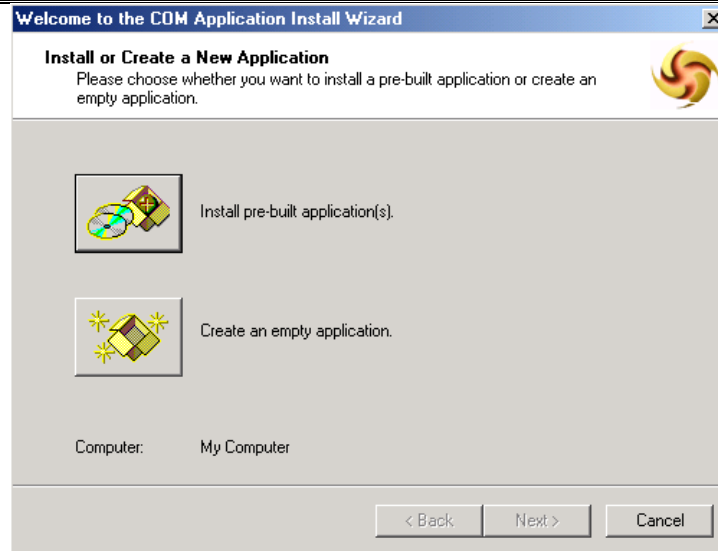
1. Tạo một thành phần COM dạng ActiveX DLL
2. Nhấn Start \ Settings \ Control Panel
3. Trong cửa sổ Control Panel mở folder Administrative Tools
4. Mở Component Services
5. Trong console tree của Component Services, Nhấn đúp vào Computers
6. Nhấn đúp vào My Computers, nhấn đúp vào COM+ Applications
7. Nhấn phải chuột vào COM+ Applications, trở tới New rồi nhấn vào Application.



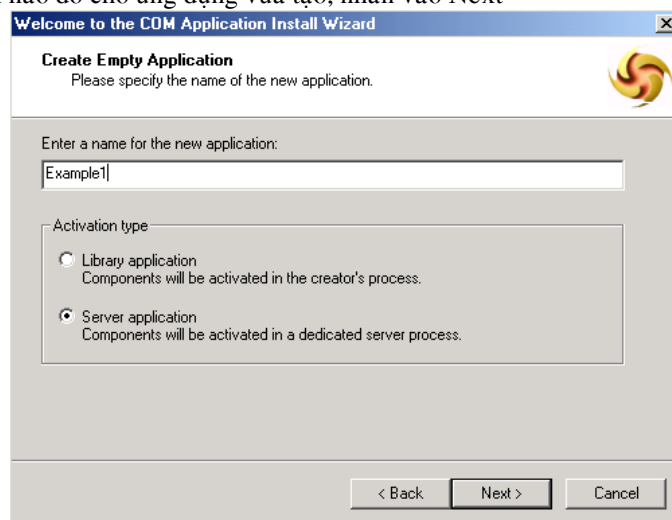
8. Xuất hiện hộp thoại COM Application Install Wizard, nhấn Next.



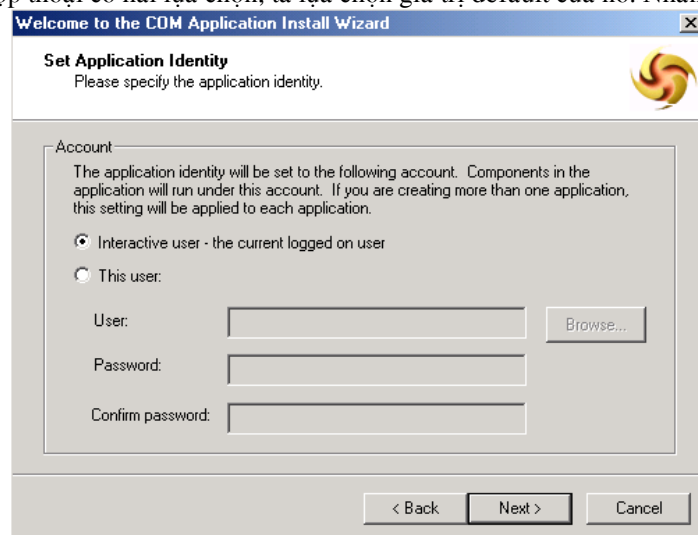
9. Tại đây sẽ có hai lựa chọn. Chọn *Create an empty Application*
 - *Install pre-built application(s)* Nếu muốn xây dựng lại ứng dụng mà ta đã tạo trước đó.
 - *Create an empty Application* : Nếu tạo một ứng dụng trống rồi import thành phần COM (.dll) vào.



10. Gõ vào vào một cái tên nào đó cho ứng dụng vừa tạo, nhấn vào Next



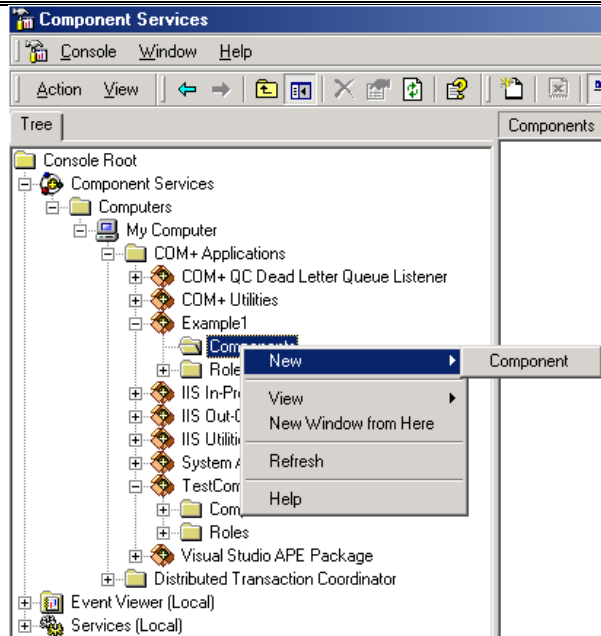
11. Tiếp theo xuất hiện hộp thoại có hai lựa chọn, ta lựa chọn giá trị default của nó. Nhấn tiếp vào Next.



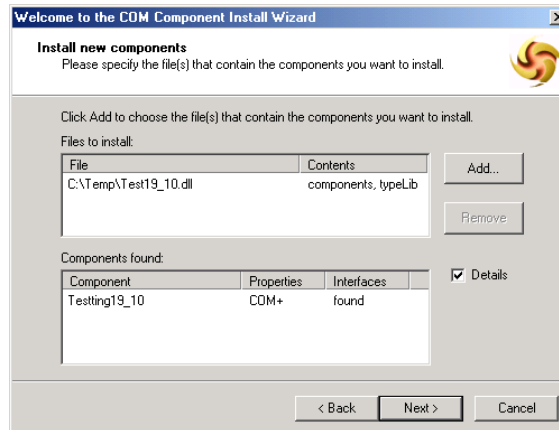
12. Nhấn Finish.

13. Nhấn đúp vào ứng dụng Example1 vừa tạo. nhấn đúp vào Components

14. Nhấn phải chuột vào Components, trở tới New, nhấn vào Component.



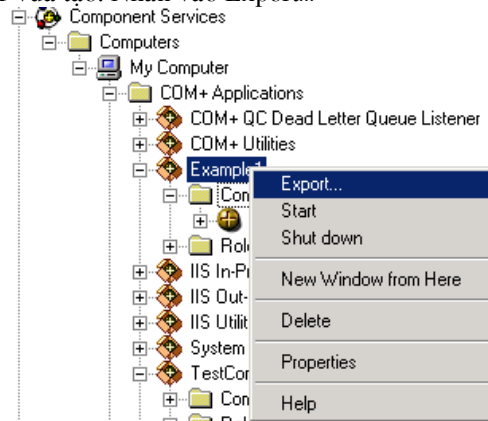
15. Nhấn vào Next, Hộp thoại Import or Install a Component xuất hiện, chọn Install new Component(s).
16. Dẫn đến chỗ lưu trữ file .dll, nhấn nút Open
17. Nhấn vào nút Next. Rồi nhấn vào Finish để kết thúc.



Để cho phép ứng dụng Client truy cập vào thành phần chạy dưới MTS, ta phải đăng ký vào Registry của máy Client để báo với nó rằng thành phần từ xa.

Để Client nhận ra thành phần MTS phân phát từ xa, ta làm như sau:

1. Nhấn phải chuột vào Example1 vừa tạo. Nhấn vào Export...



2. Dùng nút Browse để chọn một thư mục rỗng để export gói. Cho vào tên tập tin, nhấn Save.
3. Nhấn Next. Sau một lúc, hiển thị ra một hộp thoại thông báo sau:



4. Nhấn vào Finish để hoàn tất.

◆ **Các bước thực hiện trên máy chạy ứng dụng Client.**

1. Chép thư mục mà ta vừa Export được ở trên vào máy Client
2. Chạy tập tin .MSI trong thư mục này để đăng ký vào máy Client.

Muốn gỡ bỏ đăng ký này ra khỏi máy Client ta có thể vào Control Panel\Add/Remove Programs để gỡ bỏ ra.

c. Dùng RDS (Remote Data Access): Kỹ thuật này không nên sử dụng đối với mô hình này vì tốc độ chậm, kỹ thuật này được dùng cho trường hợp máy chạy ứng dụng và máy chạy ứng dụng COM không cùng mạng. Kỹ thuật này sẽ được giới thiệu sau.

Kết luận: Tóm lại đối với mô hình kiến trúc máy chạy ứng dụng Client và máy chạy ứng dụng COM cùng một mạng local thì ta chọn giải pháp “**Dùng MTS (Microsoft Transaction Server) để triển khai ActiveX DLL từ xa**”.

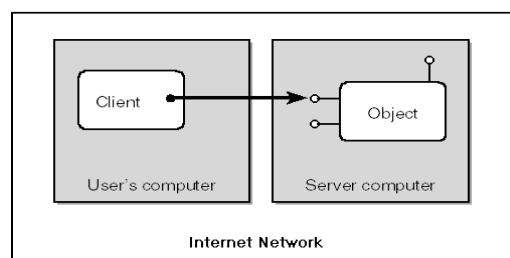
Ưu điểm của giải pháp này sau:

- So với dạng ActiveX (.exe) thì dạng này quản lý các thể hiện thành phần và quản lý giao dịch tốt hơn nhiều nhờ sự hỗ trợ của MTS.
- Về vấn đề quản lý security thì dạng này cũng hỗ trợ tốt hơn so với ActiveX (.exe).
- Còn so với RDS thì kỹ thuật này sẽ làm cho ứng dụng chạy nhanh hơn. Vì kỹ thuật RDS được dùng hỗ trợ cho việc truy cập từ xa đối với những máy không cùng mạng. Ngoài ra kỹ thuật này còn hỗ trợ cho những ứng dụng WEB.

Khuyết điểm:

- Máy chạy ứng dụng COM của giải pháp này chỉ có thể chạy trên hệ điều hành Windows NT, hoặc Windows 2000. Còn giải pháp dùng ActiveX(.exe) và giải pháp dùng RDS, máy chạy ứng dụng COM có thể chạy trên nền Windows 98, Windows NT, Windows 2000.
- So với ứng dụng dùng giải pháp dùng ActiveX(.exe) thì ứng dụng dùng giải pháp này sẽ chạy chậm hơn vì có kết hợp với MTS.

3. Máy chạy ứng dụng Client và máy chạy ứng dụng COM khác máy, 2 máy này không cùng mạng.



Để hiện thực việc giao tiếp giữa ứng dụng Client và ứng dụng COM đối với mô hình này thì phải dùng RDS(Remote Data Access).

❖ **Giới thiệu về RDS:**

RDS là một mô hình lập trình cho phép ứng dụng Client có thể tạo một thể hiện của một ứng dụng COM đặt từ xa thông qua mạng Internet bằng các giao thức http, https. RDS hỗ trợ mạnh cho những ứng dụng Web, cho phép những ứng dụng Web viết những đoạn Script thao tác dữ liệu trên Server và lấy dữ liệu về thông qua thành phần RDSServer.DataFactory, RDS.DataControl.

lưu ý: để dùng RDS để truy cập DCOM dưới đây thì ta phải dùng MTS triển khai thành phần COM đó thì ta mới có thể dùng `Set object = DataSpace.CreateObject("ProgId", "Myserver")` Còn nếu dùng trong dạng COM thì phải khai báo với hình thức `In-process`.

Protocol	Syntax
HTTP	<code>Set object = DataSpace.CreateObject("ProgId", "http://awebsrvr")</code>

HTTPS	Set object = DataSpace.CreateObject("ProgId", "https://awebsrvr")
DCOM	Set object = DataSpace.CreateObject("ProgId", "computername")
In-process	Set object = DataSpace.CreateObject("ProgId", "")

Vì RDS cho phép ứng dụng Client có thể thao tác dữ liệu thông qua mạng Internet nên vấn đề bảo mật đường truyền là vấn đề quan trọng để tránh việc thất lạc dữ liệu hay việc đánh cắp dữ liệu thì người lấy được những dữ liệu này cũng không thể xem những dữ liệu này được. Vì vậy RDS có hỗ trợ việc tạo đối tượng qua mạng Internet dùng giao thức https. Giao thức này được sử dụng với công nghệ bảo mật đường truyền SSL(Secure Sockets Layer).

◆ Giới thiệu SSL(Secure Socket Layer)

Trên Web, SSL là giao thức thông dụng cho việc giao tiếp có mã hóa giữa browser và server. Sự kết hợp giữa một giải thuật mã/giải mã, một hàm giản lược thông điệp và một cách định danh được gọi là một bộ mật mã(cipher suite). SSL có thể hỗ trợ nhiều bộ mật mã.

Để sử dụng SSL, Web Server cần phải có bằng chứng nhận(Certificate) do một nhà cung cấp chứng nhận (CA – Certificate Authority) như Verisign cấp. Ngoài ra còn có thể sử dụng bằng chứng nhận do Certificate Server của Windows NT Server hay của Windows 2000 server cấp (cũng giống như của CA nhưng không có giá trị quốc tế).

+ Một quá trình SSL :

Mục đích của một quá trình SSL là định danh server(hoặc có thể định danh client) và cuối cùng có được một khóa session cho việc giao tiếp giữa client và server trong một phiên làm việc (session).

+ Các bước của một quá trình SSL:

1. Client mở một cầu nối đến server và gửi thông điệp “ClientHello”. Thông điệp còn kèm theo một số thông tin như version của SSL, các bộ mật mã và giải thuật nén dữ liệu mà client hỗ trợ.
2. Server trả lời bằng thông điệp “Server Hello”. Thông điệp kèm theo thông tin về bộ mật mã và giải thuật nén mà server chọn cùng với một SessionID tương ứng của Client. Nếu các bộ mã được hỗ trợ bởi client và server không hợp nhau thì server sẽ gửi một thông báo “handshake failure” và chấm dứt session.
3. Server gửi “bằng chứng nhận” của nó để định danh nó đối với Client.
4. Server yêu cầu “bằng chứng nhận” của client. Điều này không bắt buộc và ít khi xảy ra.
5. Nếu client có “bằng chứng nhận” thì nó sẽ gửi cho server. Nếu không, nó sẽ gửi một thông báo không có bằng.
6. Client gửi thông điệp “trao đổi khóa” cùng với một bao điện tử chứa khóa session do client tạo ngẫu nhiên và được mã hóa bằng khóa public của server được lấy từ bằng chứng nhận của server.
7. Cả Client và Server cùng gửi thông báo “ChangeCipherSpec” để xác nhận cả hai đã sẵn sàng giao tiếp.
8. Cả hai cùng gửi thông báo “chấm dứt” phần thủ tục và bắt đầu giao tiếp có bảo mật bằng khóa session ở trên. Ngoài ra thông điệp cuối cùng này còn có chứa hash của toàn bộ quá trình làm thủ tục trên để cả hai cùng kiểm tra xem quá trình có bị trục trặc gì không.

◆ Cấu hình RDS trong Windows 2000(Máy chạy ứng dụng COM)

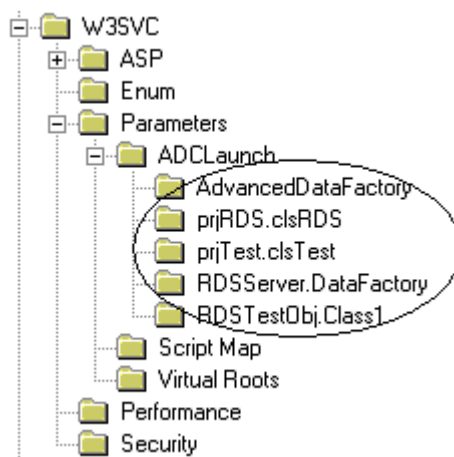
Có hai sự cấu hình liên quan đến Remote Data Access: RDS security và IIS security.

+ RDS security:

Để ứng dụng Client có thể dùng RDS để giao tiếp với ứng dụng COM thì trên thư mục ảo(wwwroot đối với windows 2000) của máy chạy ứng dụng COM phải có thư mục MSADC, trong đó có tập tin msadc.dll.

Ngoài ra cần phải khai báo bằng tay trong registry của máy chạy ứng dụng COM (mở ra bằng cách chạy tập tin regedit.exe) ProgId của ứng dụng COM (tên dự án.tên những class) tại vị trí sau:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\



+ *IIS security*:

Để thiết lập IIS security thực hiện lần lượt các bước sau:

1. Mở Control Panel, mở *Administrative Tools*, mở *Internet Services Manager*.
2. Nhấn đúp lên *Default Web Site*.
3. Nhấn phải chuột trên thư mục ảo MSADC, nhấn *Properties*.
4. Trên *Directory Security* tab, chỗ *IP Address and Domain Name Restrictions*, nhấn *Edit*.
 - Nếu muốn tất cả các Client có thể giao tiếp với ứng dụng chọn *Granted Access*.
 - Nếu muốn chỉ cho phép một số Client nào đó có thể truy cập đến ứng dụng, chọn Client rồi *Add* vào danh sách.

◆ Ngoài ra ta có thể dùng RDS để truy cập đến những ứng dụng COM được khai triển từ xa dùng MTS thông qua Internet. Dùng RDS để truy cập những ứng dụng DCOM dạng ActiveX EXE.