

# Tối ưu mã nguồn C/C++

## Tại sao phải tối ưu mã lệnh?

Sự ra đời của các trình biên dịch hiện đại đã giúp lập trình viên cải thiện đáng kể thời gian và công sức phát triển phần mềm. Một vấn đề đáng quan tâm là xu hướng phát triển phần mềm theo hướng trực quan nhanh và tiện dụng dần làm mất bằng kỹ năng viết mã lệnh của các lập trình viên giảm rõ rệt vì họ trông cậy hoàn toàn vào sự hỗ trợ của trình biên dịch. Khi phát triển một hệ thống phần mềm có tần suất xử lý cao, ví dụ các sản phẩm có chức năng điều phối hoạt động dây chuyền sản xuất trong nhà máy, thì bên cạnh sự hỗ trợ của một trình biên dịch mạnh còn cần đến kỹ năng tối ưu mã lệnh của lập trình viên. Kỹ năng tốt sẽ biến công việc lập trình khô khan, với các đoạn code tưởng chừng lạnh lùng trở nên sinh động. Một đoạn mã lệnh tốt sẽ tận dụng tối đa ưu điểm của ngôn ngữ và khả năng xử lý của hệ thống, từ đó giúp nâng cao đáng kể hiệu suất hoạt động của hệ thống.

Để chương trình hoạt động tối ưu, điều đầu tiên là tận dụng những hỗ trợ sẵn có của trình biên dịch thông qua các chỉ thị (directive) giúp tối ưu mã lệnh, tốc độ và kích thước chương trình. Hầu hết các trình biên dịch phổ biến hiện nay đều hỗ trợ tốt việc tối ưu mã khi biên dịch. Tuy nhiên, để đạt được hiệu quả tốt nhất, lập trình viên cần tập cho mình thói quen tối ưu mã lệnh ngay từ khi bắt tay viết những chương trình đầu tay. Bài viết này trình bày một số gợi ý rất cơ bản và kinh nghiệm thực tế tối ưu trong lập trình bằng ngôn ngữ C/C++.

## Tinh giản các biểu thức toán học

Các biểu thức toán học phức tạp khi được biên dịch có thể sinh ra nhiều mã dư thừa làm tăng kích thước và chậm tốc độ thực hiện của chương trình. Do đó khi viết các biểu thức phức tạp lập trình viên cần nhớ một số đặc điểm cơ bản sau để giúp tinh giản biểu thức:

- CPU xử lý các phép tính cộng và trừ nhanh hơn các phép tính chia và nhân.

*Ví dụ:*

+ Biểu thức  $Total = (A*B + A*C + A*D)$  cần 2 phép cộng và 3 phép nhân. Ta có thể nhóm các phép cộng và viết thành  $Total = A*(B+C+D)$ , tốc độ tính nhanh hơn vì giảm đi một phép tính nhân.

+ Biểu thức  $Total = (B/A + C/A)$  cần 2 phép chia có thể viết thành  $Total = (B+C)/A$ , giúp giảm đi một phép chia.

- CPU xử lý tính toán với các số nguyên (integer) chậm hơn với số thực (float, double), và tốc độ xử lý float nhanh hơn double.

- Trong một số trường hợp nhân hoặc chia số nguyên, sử dụng toán tử dời bit (bit shifting) sẽ nhanh hơn toán tử nhân chia.

*Ví dụ:*

Biểu thức  $(A * 128)$  có thể tận dụng toán tử dời bit sang trái thành  $(A << 7)$ .

Một số trình biên dịch có khả năng tối ưu mã khi biên dịch như Visual C++ 6 hoặc .Net 2003, biểu thức  $(A * 128)$  và  $(A << 7)$  đều được biên dịch thành:

```
mov eax, A
```

```
shl eax, 7 (toán tử shl được dùng thay vì mul/imul)
```

```
mov A, eax
```

Ta có thể tối ưu bằng cách sử dụng mã assembly trực tiếp trong mã C/C++ như sau (xem thêm thủ thuật tận dụng thế mạnh của C/C++ bên dưới):

```
__asm shl A, 7;
```

### Tối ưu việc sử dụng biến tạm

Đối với một số biểu thức tính toán số học phức tạp, trình biên dịch thường tạo các biến tạm trong bộ nhớ để chứa kết quả tính toán và cuối cùng mới gán giá trị này cho biến kết quả. Việc sử dụng biến tạm làm giảm tốc độ tính toán do phải cấp phát vùng nhớ, tính toán và thực hiện việc gán kết quả cuối cùng. Để tránh việc sử dụng biến tạm, ta có thể thực hiện việc tách các biểu thức phức tạp thành các biểu thức nhỏ hơn, hoặc sử dụng các mẹo cho việc tính toán.

Xem một ví dụ cộng các số nguyên sau:

```
A = B + C
```

Về cơ bản, khi thực hiện biểu thức này trình biên dịch tạo một biến tạm rồi thực hiện cộng 2 giá trị B, C vào biến tạm này, cuối cùng sẽ gán kết quả cho A.

Ta có thể viết lại biểu thức trên như sau để tránh sử dụng biến tạm làm chậm việc tính toán:

```
A = B;
```

```
A += C;
```

Trong lập trình hướng đối tượng, theo thói quen đôi khi lập trình viên sử dụng các biến tạm không cần thiết như trong ví dụ sau:

```
int MyFunc(const MyClass &A)
{
    MyClass B;
    B = A;
    return B.value;
}
```

Trong hàm trên, khi biến tạm B kiểu MyClass được khởi tạo thì constructor mặc định sẽ được thực hiện. Sau đó B được gán giá trị của biến A thông qua việc sử dụng toán tử =, khi đó copy constructor sẽ được gọi. Tuy nhiên với yêu cầu của bài toán thì việc này không cần thiết, ta có thể viết lại như sau:

```
int MyFunc(const MyClass &A)
{
    return A.value;
}
```

```
}
```

Dưới đây là một ví dụ khác cho bài toán hoán vị giá trị 2 số nguyên A và B. Thông thường, yêu cầu này sẽ được viết như sau:

```
int A = 7, B = 8;

int nTemp; //biến tạm

nTemp = A;

A = B;

B = nTemp;
```

Tuy nhiên, bạn có thể sử dụng mẹo sau để tránh sử dụng biến tạm và tăng tốc tính toán:

+ Sử dụng toán tử XOR:

```
A = A^B;

B = A^B;

A = A^B;
```

+ Sử dụng phép cộng, trừ

```
nX = nX + nY;

nY = nX - nY;

nX = nX - nY;
```

Bạn hãy chạy thử đoạn mã lệnh trên sẽ thấy điều bất ngờ thú vị khi bài toán hoán vị được giải quyết hết sức đơn giản.

Thuật toán tránh sử dụng biến tạm cần áp dụng linh động tùy thuộc kiểu dữ liệu, đặc biệt các loại dữ liệu phức tạp như kiểu structure, string... có cơ chế lưu trữ và xử lý riêng. Đối với các trình biên dịch hiện đại, việc tối ưu theo cách này đôi khi không cần thiết vì trình biên dịch đã hỗ trợ sẵn cơ chế tối ưu này khi biên dịch mã lệnh.

### Tối ưu các biểu thức điều kiện và luận lý

Biểu thức điều kiện là thành phần không thể thiếu ở hầu hết các chương trình máy tính vì nó giúp lập trình viên biểu diễn và xử lý được các trạng thái của thế giới thực dưới dạng các mã lệnh máy tính. Những điều kiện dư thừa có thể làm chậm việc tính toán và gia tăng kích thước mã lệnh, thậm chí có những đoạn mã có xác suất xảy ra rất thấp. Một trong những tiêu chí quan trọng của việc tối ưu các biểu thức điều kiện là đưa các điều kiện có xác suất xảy ra cao nhất, tính toán nhanh nhất lên đầu biểu thức.

Đối với các biểu thức luận lý, ta có thể linh động chuyển các biểu thức điều kiện đơn giản và xác suất xảy ra cao hơn lên trước, các điều kiện kiểm tra phức tạp ra sau.

*Ví dụ:* Biểu thức logic ((A || B) && C) có thể viết thành (C && (A || B)) vì điều kiện C chỉ cần một phép

kiểm tra TRUE, trong khi điều kiện  $(A \parallel B)$  cần đến 2 phép kiểm tra TRUE và một phép OR ( $\parallel$ ). Như vậy trong trường hợp C có giá trị FALSE, biểu thức logic này sẽ có kết quả FALSE và không cần kiểm tra thêm giá trị  $(A \parallel B)$ .

Đối với các biểu thức kiểm tra điều kiện phức tạp, ta có thể viết đảo ngược bằng cách kiểm tra các giá trị cho kết quả không thoả trước, giúp tăng tốc độ kiểm tra.

*Ví dụ:* Kiểm tra một giá trị thuộc một miền giá trị cho trước.

```
if (p <= max && p >= min && q <= max && q >= min)
{
    //thực hiện khi thoả miền giá trị
}
else //không thoả
{
    //thực hiện khi không thoả miền giá trị
}
```

Có thể viết thành:

```
if (p > max || p < min || q > max || q < min)
{
}
else
{
}
```

Tránh các tính toán lặp lại trong biểu thức điều kiện

*Ví dụ:*

```
if ((mydata->MyFunc() ) < min)
{
    // ...
}
else if ((mydata->MyFunc() ) > max)
{
    // ...
}
```

Ta có thể chuyển hàm MyFunc ra ngoài biểu thức điều kiện như sau:

```
int temp_value = mydata->MyFunc();

if (temp_value < min)

{

// ...

}

else if (temp_value > max)

{

// ...

}
```

Đối với biểu thức điều kiện dạng switch...case: nếu các giá trị cho case liên tục nhau, trình biên dịch sẽ tạo ra bảng ánh xạ (còn gọi là jump table) giúp việc truy xuất đến từng điều kiện nhanh hơn và giảm kích thước mã lệnh. Tuy nhiên khi các giá trị không liên tục, trình biên dịch sẽ tạo một chuỗi các phép toán so sánh, từ đó gây chậm việc xử lý:

Ví dụ sau cho kết quả truy xuất tối ưu khi sử dụng switch...case:

```
switch (my_value)
{
case A:
...
break;
case B:
...
break;
case C:
...
break;
case D:
...
default:
...
}
```

Trong trường hợp các giá trị dùng cho case không liên tục, ta có thể viết thành các biểu thức if...elseif...else như sau:

```
switch (my_value)
{
case A:
...
break;
case F:
...
break;
case T:
}
```

```
Có thể viết thành:
if (my_value == A)
{
// xử lý cho trường hợp A
}
else if (my_value == F)
{
// xử lý cho trường hợp F
}
else
{
// các trường hợp khác
}
```

## Ø Tối ưu vòng lặp

Vòng lặp cũng là một thành phần cơ bản phản ánh khả năng tính toán không mệt mỏi của máy tính. Tuy nhiên, việc sử dụng máy móc vòng lặp là một trong những nguyên nhân làm giảm tốc độ thực hiện của chương trình. Một số thủ thuật sau sẽ giúp lập trình viên tăng tốc vòng lặp của mình:

- Đối với các vòng lặp có số lần lặp nhỏ, ta có thể viết lại các biểu thức tính toán mà không cần dùng vòng lặp. Nhờ vậy tiết kiệm được khoảng thời gian quản lý và tăng biến đếm trong vòng lặp.

Ví dụ cho vòng lặp sau:

```
for( int i = 0; i < 4; i++ )
{
array[i] =MyFunc(i);
}
```

Có thể viết lại thành:

```
array[0] = MyFunc(0);
array[1] = MyFunc(1);
array[2] = MyFunc(2);
array[3] = MyFunc(3);
```

- Đối với các vòng lặp phức tạp có số lần lặp lớn, cần hạn chế việc cấp phát các biến nội bộ và các phép tính lặp đi lặp lại bên trong vòng lặp mà không liên quan đến biến đếm lặp.

Ví dụ cho vòng lặp sau:

```
int students_number = 10000;
for( int i = 0; i < students_number; i++ )
{
//hàm MyFunc mất nhiều thời gian thực hiện
double sample_value = MyFunc(students_number);
```

```
CalcStudentFunc(i, sample_value);  
  
}
```

Trong ví dụ trên, biến `sample_value` được tính ở mỗi vòng lặp một cách không cần thiết vì hàm `MyFunc` có thể tốn rất nhiều thời gian, ta có thể dời đoạn mã tính toán này ra ngoài vòng lặp như sau:

```
int students_number = 10000;  
  
double sample_value = MyFunc(students_number);  
  
for( int i = 0; i < students_number; i++ )  
{  
    CalcStudentFunc(i, sample_value);  
}
```

- Đối với vòng lặp từ 0 đến n phần tử như sau:

```
for( int i = 0; i < max_number; i++ )
```

Nên thực hiện việc lặp từ giá trị `max_number` trở về 0 như sau:

```
for( int i = max_number - 1; i >=0 ; -- i )
```

Vì khi biên dịch thành mã máy, các phép so sánh với 0 (zero) sẽ được thực hiện nhanh hơn với các số nguyên khác. Do đó phép so sánh ở mỗi vòng lặp là `( i >=0 )` sẽ nhanh hơn phép so sánh `( i < max_number )`.

- Trong vòng lặp lớn, các toán tử prefix dạng `--i` hoặc `++i` sẽ thực hiện nhanh hơn toán tử postfix `(i--` hoặc `++)`. Nguyên nhân là do toán tử prefix tăng giá trị của biến trước sau đó trả kết quả về cho biểu thức, trong khi toán tử postfix phải lưu giá trị cũ của biến vào một biến tạm, tăng giá trị của biến và trả về giá trị của biến tạm.

## Tối ưu việc sử dụng bộ nhớ và con trỏ

Con trỏ (pointer) có thể được gọi là một trong những "niềm tự hào" của C/C++, tuy nhiên thực tế nó cũng là nguyên nhân làm đau đầu cho các lập trình viên, vì hầu hết các trường hợp sụp đổ hệ thống, hết bộ nhớ, vi phạm vùng nhớ... đều xuất phát từ việc sử dụng con trỏ không hợp lý.

- Hạn chế pointer dereference: pointer dereference là thao tác gán địa chỉ vùng nhớ dữ liệu cho một con trỏ. Các thao tác dereference tốn nhiều thời gian và có thể gây hậu quả nghiêm trọng nếu vùng nhớ đích chưa được cấp phát.

Ví dụ với đoạn mã sau:

```
for( int i = 0; i < max_number; i++ )  
{  
    SchoolData->ClassData->StudentData->Array[i] = my_value;  
}
```

Bằng cách di chuyển các pointer dereference nhiều cấp ra ngoài vòng lặp, đoạn mã trên có thể viết lại như sau:

```
unsigned long *Temp_Array = SchoolData->ClassData->StudentData->Array;

for( int i = 0; i < max_number; i++ )

{

Temp_Array[i] = my_value;

}
```

- Sử dụng tham chiếu (reference) cho đối tượng dữ liệu phức tạp trong các tham số hàm. Việc sử dụng tham chiếu khi truyền nhận dữ liệu ở các hàm có thể giúp tăng tốc đáng kể đối với các cấu trúc dữ liệu phức tạp. Trong lập trình hướng đối tượng, khi một đối tượng truyền vào tham số dạng giá trị thì toàn bộ nội dung của đối tượng đó sẽ được sao chép bằng copy constructor thành một bản khác khi truyền vào hàm. Nếu truyền dạng tham chiếu thì loại trừ được việc sao chép này. Một điểm cần lưu ý khi sử dụng tham chiếu là giá trị của đối tượng có thể được thay đổi bên trong hàm gọi, do đó lập trình viên cần sử dụng thêm từ khóa const khi không muốn nội dung đối tượng bị thay đổi.

*Ví dụ:* Khi truyền đối tượng dạng giá trị vào hàm để sử dụng, copy constructor sẽ được gọi.

```
void MyFunc(MyClass A) //copy constructor của A sẽ được gọi

{

int value = A.value;

}
```

Khi dùng dạng tham chiếu, đoạn mã trên có thể viết thành:

```
void MyFunc(const MyClass &A) //không gọi copy constructor

{

int value = A.value;

}
```

- Tránh phân mảnh vùng nhớ: Tương tự như việc truy xuất dữ liệu trên đĩa, hiệu năng truy xuất các dữ liệu trên vùng nhớ động sẽ giảm đi khi bộ nhớ bị phân mảnh. Một số gợi ý sau sẽ giúp giảm việc phân mảnh bộ nhớ.

+ Tận dụng bộ nhớ tĩnh. Ví dụ: như tốc độ truy xuất vào một mảng tĩnh có tốc độ nhanh hơn truy xuất vào một danh sách liên kết động.

+ Khi cần sử dụng bộ nhớ động, tránh cấp phát hoặc giải phóng những vùng nhớ kích thước nhỏ. Ví dụ như ta có thể tận dụng xin cấp phát một mảng các đối tượng thay vì từng đối tượng riêng lẻ.

+ Sử dụng STL container cho các đối tượng hoặc các cơ chế sử dụng bộ nhớ riêng có khả năng tối ưu việc cấp phát bộ nhớ. STL cung cấp rất nhiều thuật toán và loại dữ liệu cơ bản giúp tận dụng tối đa hiệu năng của C++. Các bạn có thể tìm đọc các sách về STL sẽ biết thêm nhiều điều thú vị.



- Sau khi cấp phát một mảng các đối tượng, tránh nhầm lẫn khi sử dụng toán tử delete[] và delete: với C++, toán tử delete[] sẽ chỉ định trình biên dịch xóa một chuỗi các vùng nhớ, trong khi delete chỉ xóa vùng nhớ mà con trỏ chỉ đến, do đó có thể gây hiện tượng "rác" và phân mảnh bộ nhớ.

*Ví dụ:*

```
int *myarray = new int[50];  
  
delete []myarray;
```

Với delete[], trình biên dịch sẽ phát sinh mã như sau:

```
mov ecx, dword ptr [myarray]  
  
mov dword ptr [ebp-6Ch], ecx  
  
mov edx, dword ptr [ebp-6Ch]  
  
push edx  
  
call operator delete[] (495F10h) //gọi toán tử delete[]  
  
add esp, 4
```

Trong khi với đoạn lệnh:

```
int *myarray = new int[50];  
  
delete myarray;
```

Trình biên dịch sẽ phát sinh mã như sau:

```
mov ecx, dword ptr [myarray]  
  
mov dword ptr [ebp-6Ch], ecx  
  
mov edx, dword ptr [ebp-6Ch]  
  
push edx  
  
call operator delete (495F10h) //gọi toán tử delete  
  
add esp, 4
```

## **Sử dụng hợp lý cơ chế bẫy lỗi try...catch**

Việc sử dụng không hợp lý các bẫy lỗi có thể là sai lầm tai hại vì trình biên dịch sẽ thêm các mã lệnh kiểm tra ở các đoạn mã được cài đặt try...catch, điều này làm tăng kích thước và giảm tốc độ xử lý của chương trình, đồng thời gây khó khăn trong việc sửa chữa các lỗi logic. Thống kê cho thấy các đoạn mã có sử dụng bẫy lỗi thì hiệu suất thực hiện giảm từ 5%-10% so với đoạn mã thông thường được viết cẩn thận. Để hạn chế điều này, lập trình viên chỉ nên đặt bẫy lỗi ở những đoạn mã có nguy cơ lỗi cao và khả năng dự báo trước thấp

## **Tận dụng đặc tính xử lý của CPU**

Để đảm bảo tốc độ truy xuất tối ưu, các bộ vi xử lý (CPU) 32-bit hiện nay yêu cầu dữ liệu sắp xếp và tính toán trên bộ nhớ theo từng offset 4-byte. Yêu cầu này gọi là memory alignment. Do vậy khi biên dịch một đối tượng dữ liệu có kích thước dưới 4-byte, các trình biên dịch sẽ bổ sung thêm các byte trống để đảm bảo các

dữ liệu được sắp xếp theo đúng quy luật. Việc bổ sung này có thể làm tăng đáng kể kích thước dữ liệu, đặc biệt đối với các cấu trúc dữ liệu như structure, class...

Xem ví dụ sau:

```
class Test
{
    bool a;

    int c;

    int d;

    bool b;

};
```

Theo nguyên tắc alignment 4-byte (hai biến "c" và "d" có kích thước 4 byte), các biến "a" và "b" chỉ chiếm 1 byte và sau các biến này là biến int chiếm 4 byte, do đó trình biên dịch sẽ bổ sung 3 byte cho mỗi biến này. Kết quả tính kích thước của lớp Test bằng hàm sizeof(Test) sẽ là 16 byte.

Ta có thể sắp xếp lại các biến thành viên của lớp Test như sau theo chiều giảm dần kích thước:

```
class Test
{
    int c;

    int d;

    bool a;

    bool b;

};
```

Khi đó, hai biến "a" và "b" chiếm 2 byte, trình biên dịch chỉ cần bổ sung thêm 2 byte sau biến "b" để đảm bảo tính sắp xếp 4-byte. Kết quả tính kích thước sau khi sắp xếp lại class Test sẽ là 12 byte.

### **Tận dụng một số ưu điểm khác của C++**

- Khi thiết kế các lớp (class) hướng đối tượng, ta có thể sử dụng các phương thức "inline" để thực hiện các xử lý đơn giản và cần tốc độ nhanh. Theo thống kê, các phương thức inline thực hiện nhanh hơn khoảng 5-10 lần so với phương thức được cài đặt thông thường.

- Sử dụng ngôn ngữ cấp thấp assembly: một trong những ưu điểm của ngôn ngữ C/C++ là khả năng cho phép lập trình viên chèn các mã lệnh hợp ngữ vào mã nguồn C/C++ thông qua từ khóa `__asm { ... }`. Lợi thế này giúp tăng tốc đáng kể khi biên dịch và khi chạy chương trình.

*Ví dụ:*

```
int a, b, c, d, e;
```

```
e = a*b + a*c;
```

Trình biên dịch phát sinh mã hợp ngữ như sau:

```
mov eax, dword ptr [a]
imul eax, dword ptr [b]
mov ecx, dword ptr [a]
imul ecx, dword ptr [c]
add eax, ecx
mov dword ptr [e], eax
```

Tuy nhiên, ta có thể viết rút gọn giảm được 1 phép imul (nhân), 1 phép mov (di chuyển, sao chép):

```
__asm
{
    mov eax, b;
    add eax, c;
    imul eax, a;
    mov e, eax;
};
```

- Ngôn ngữ C++ cho phép sử dụng từ khóa "register" khi khai báo biến để lưu trữ dữ liệu của biến trong thanh ghi, giúp tăng tốc độ tính toán vì truy xuất dữ liệu trong thanh ghi luôn nhanh hơn truy xuất trong bộ nhớ.

*Ví dụ:*

```
for (register int i; i < max_number; ++i )
{
    // xử lý trong vòng lặp
}
```

- Ngôn ngữ C/C++ hỗ trợ các collection rất mạnh về tốc độ truy xuất như các bảng map, hash\_map,... nên tận dụng việc sử dụng các kiểu dữ liệu này thay cho các danh sách liên kết bộ nhớ động (linked list).

## Kết luận

Một chương trình được đánh giá tốt khi tất cả các bộ phận tham gia vào hoạt động của chương trình đạt hiệu suất cao nhất theo yêu cầu của người sử dụng. Một dòng lệnh đơn giản tưởng chừng sẽ hoạt động trong tích tắc có thể làm hệ thống trở nên chậm chạp khi được gọi hàng ngàn, hàng triệu lần trong khoảng thời gian ngắn. Do vậy, trong suốt qui trình hình thành sản phẩm phần mềm, giai đoạn cài đặt mã lệnh chiếm vai trò hết sức quan trọng và cần kỹ năng tối ưu hóa cao nhất. Để đạt được điều đó, không cách nào khác hơn là lập

trình viên cần tự rèn luyện thật nhiều để thông thạo ngôn ngữ mình chọn lựa, trình biên dịch mình sử dụng. Khi đó lập trình không còn là việc tạo những đoạn mã khô khan, mà là một nghệ thuật.

Nguyễn Văn Sơn  
Global CyberSoft Vietnam  
[sonnv@cybersoft-vn.com](mailto:sonnv@cybersoft-vn.com)

# Lập trình thay đổi Component Palette của Delphi IDE

Nếu bạn thường làm việc với Delphi, nếu Delphi của bạn đã được cài đặt thêm rất nhiều các thành phần điều khiển (component) và nếu bạn luôn phải sử dụng rất nhiều component trong các dự án của mình thì có bao giờ bạn thấy mệt mỏi khi phải tìm đến biểu tượng component Palette mà mình mong muốn trên thanh công cụ Component hay không?

Component Palette của Delphi IDE đơn giản là một điều khiển dạng TAB với tiêu đề chỉ gồm một hàng duy nhất, vì vậy sẽ khiến bạn mất nhiều thời gian tìm kiếm khi có quá nhiều component. Bài viết này nhằm giúp giải tỏa "nỗi bức xúc" trên bằng cách thiết lập thuộc tính Multi-lines cho điều khiển TAB Component Palette bằng những thủ thuật đơn giản mà có khi bạn không hề ngờ tới. Ở đây tôi sử dụng Delphi 7 tuy nhiên với các phiên bản thấp hơn cũng không có nhiều thay đổi.

## Giới thiệu về Delphi IDE

Delphi IDE (Integrated Development Environment) là môi trường phát triển tích hợp của Delphi. Tùy thuộc vào từng phiên bản cụ thể của Delphi mà các thành phần của Delphi IDE cũng có những thay đổi nhất định. Chẳng hạn trong Delphi 7, IDE gồm có 5 thành phần chính đó là:

- 1. Cửa sổ chính của Delphi:** Tên mã của cửa sổ này là *TAppBuilder*. Cửa sổ này bao gồm trình đơn, các thanh công cụ và một bảng gồm các công cụ phát triển (Component Palette).
- 2. Cửa sổ thiết kế FORM:** Đây chính là cửa sổ thực tế dành cho chương trình ứng dụng của bạn. Khởi đầu cửa sổ là một FORM trống mỗi khi bạn khởi động Delphi.
- 3. Cửa sổ Object Inspector:** Tên mã của cửa sổ là *TPropertyInspector*. Đây là cửa sổ cho phép bạn thay đổi các thuộc tính cho thành phần trên FORM như tiêu đề, tên... một cách trực quan.
- 4. Cửa sổ soạn thảo mã lệnh Code Editor:** Tên mã của cửa sổ là *TEditWindow*. Đây là nơi thực sự thể hiện nội dung của chương trình, là nơi bạn gõ lệnh, thiết kế nội dung cho thủ tục, cho hàm và cài đặt các phương thức cho lớp.

**5. Cửa sổ Object TreeView:** Tên mã của cửa sổ là *TObjectTreeView*. Cửa sổ sẽ thể hiện cho bạn một cách trực quan thứ tự cha con của các thành phần có mặt trên FORM...

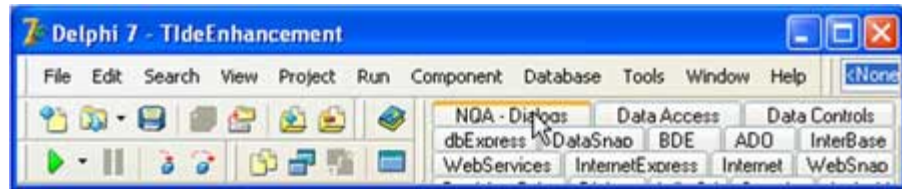
Bản thân Delphi IDE là một môi trường lắp ghép. Delphi mở ra cho bạn rất nhiều cách tiếp cận để thay đổi và chỉnh sửa sao cho phù hợp và thuận lợi với từng cá nhân. Chẳng hạn, thanh *Component Palette* của Delphi IDE thực tế là một đối tượng *TTabControl* không hơn không kém. Bạn có thể thấy được điều này thông qua một phần đoạn mã dùng để cài đặt cho cửa sổ *TAppBuilder*.

```
object TabControl: TComponentPaleAppBuildertteTabControl
  Left = 0
  Top = 0
  Width = 64
  Height = 47
  Align = alClient
  Constraints.MinWidth = 20
  HotTrack = True
  PopupMenu = PaletteMenu
  TabOrder = 0
  TabStop = False
  OnChange = TabControlChange
  OnDragDrop = TabControlDragDrop
  OnDragOver = TabControlDragOver
  OnEndDrag = TabControlEndDrag
  OnMouseDown = TabControlMouseDown
  OnMouseMove = TabControlMouseMove
  OnStartDrag = TabControlStartDrag
  BorderStyle = bsNone
  OnHelpRequest = ComponentPaletteHelpRequest
  object PageScroller1: TPageScroller
    Left = 32
    Top = 6
    Width = 31
    Height = 39
    Align = alClient
    AutoScroll = True
    TabOrder = 0
    OnScroll = PageScroller1Scroll
  end
  object Panel2: TPanel
    Left = 4
    Top = 6
    Width = 28
    Height = 39
    Align = alLeft
    BevelOuter = bvNone
    TabOrder = 1
    object SelectorButton: TSpeedButton
      Left = 0
      Top = 0
      Width = 28
      Height = 28
      GroupIndex = 1
      Down = True
      Flat = True
    end
  end
end
end
end
```

Như vậy, có hai cách để thiết lập thuộc tính Multi-lines cho điều khiển TAB Component Palette. Ý tưởng của cách thứ nhất là trực tiếp thay đổi mã nhị phân của file **delphi32.exe** trong thư mục BIN của Delphi. Để làm được điều này các bạn hãy thêm vào phần cài đặt thuộc tính của TabControl trong đoạn mã ở trên dòng lệnh sau:

```
MultiLine = True
```

Tôi đã thử cách này và kết quả mang lại khá tốt. Tuy nhiên cách này có một nhược điểm nhỏ khi Component Palette của bạn đang ở trạng thái Dock trên cửa sổ chính của Delphi thì việc thay đổi kích thước xem chừng không thể (xem hình 1).



Hình 1: Lỗi với cách sửa trực tiếp file delphi32.exe

Ý tưởng của cách thứ 2 là ta sẽ viết một component nhỏ. Mỗi khi Delphi nạp component này nó sẽ có nhiệm vụ đi tìm cửa sổ chính của Delphi, tiếp đến tìm đúng điều khiển TAB Component Palette và thay đổi trực tiếp thuộc tính MultiLine của TAB. Trông thì cứ như là chuyện không tưởng nhưng như đã đề cập, Delphi IDE là một môi trường lắp ghép chuyên nghiệp. Bản thân Delphi IDE mở ra rất nhiều hướng để bạn tùy biến. Chúng ta sẽ từng bước tìm hiểu mã lệnh để thực hiện những công việc trên.

### Tìm cửa sổ chính của Delphi

Có rất nhiều cách để tìm đến cửa sổ chính của Delphi. Lưu ý, component mà bạn chuẩn bị viết tương tác trực tiếp với Delphi IDE nên bản thân nó lấy cửa sổ Application như là cửa sổ Application của Delphi. Vì vậy, theo ý kiến riêng, bạn có thể dùng đoạn mã sau để tìm cửa sổ chính:

```
function GetIdeMainForm: TCustomForm;
begin
    Result := TForm(Application.FindComponent(AppBuilder));
end;
```

### Tìm điều khiển TAB Component Palette

Để tìm được điều khiển TAB này, bạn hãy dùng đoạn mã sau:

```
function GetTabControl : TTabControl;
var
    MainForm : TCustomForm;
begin
    Result := nil;
    MainForm := GetIdeMainForm;
    if MainForm <> nil then
        Result := TTabControl(MainForm.FindComponent(TabControl))
    end;
```

### Tìm menu popup của điều khiển TAB Component Palette

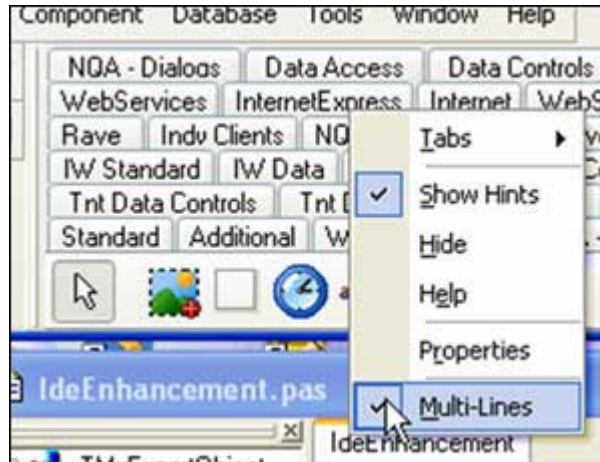
Để làm được điều này, bạn hãy dùng:

```

function GetComponentPalettePopupMenu : TPopupMenu;
var
    MainForm : TCustomForm;
begin
    Result := nil;
    MainForm := GetIdeMainForm;
    if MainForm <> nil then
        Result := TPopupMenu(MainForm.FindComponent(PaletteMenu));
end;

```

Sở dĩ chúng ta muốn tìm menu popup này vì ta sẽ thêm một mục chọn Multi-Lines dùng để chuyển đổi giữa hai trạng thái của TAB Component Palette (xem hình 2).



Hình 2: Mục chọn mới

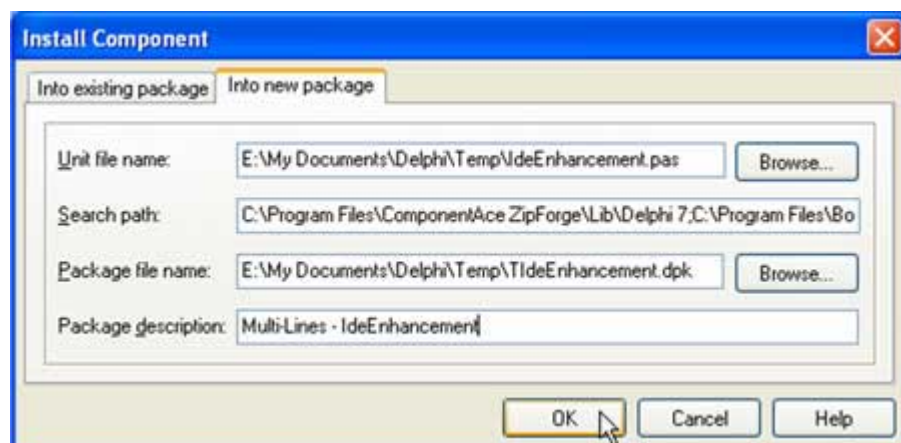
Toàn bộ nội dung mã lệnh của component có thể xem ở phần "[Mã nguồn](#)".

## Cài đặt và sử dụng

Để sử dụng component vừa tạo, bạn cần phải cài đặt vào Delphi IDE.

**Bước 1.** Lưu toàn bộ nội dung mã lệnh ở trên vào một file, chẳng hạn tôi chọn file tên là *IdeEnhancement.pas*.

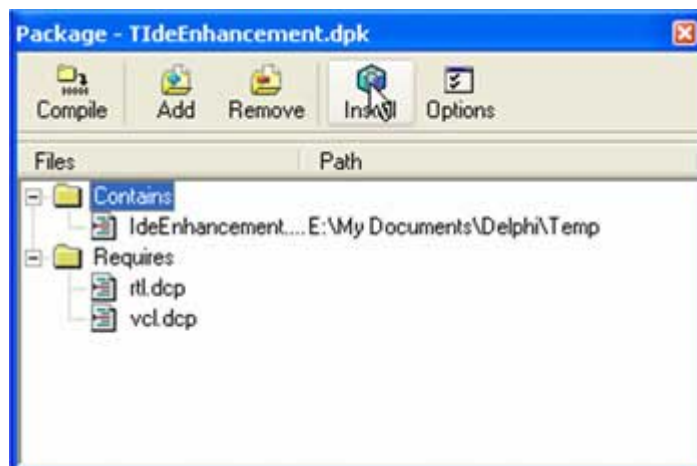
**Bước 2.** Chọn chức năng *Install Component* trên menu Component của Delphi IDE. Một cửa sổ mới xuất hiện. Bạn hãy khai báo các thông tin như ở hình 3. Sau đó nhấn *OK*.



Hình 3: Thiết lập thông tin cho component

**Bước 3.** Delphi sẽ hỏi bạn có biên dịch ngay component này hay không. Bạn hãy mạnh dạn chọn "không". Sau đó ghi lại những gì vừa thực hiện.

**Bước 4.** Trong cửa sổ Package của IDE bạn hãy chọn chức năng *Install* (xem hình 4).



Hình 4: Cài đặt Component

Như vậy là đã xong. Bạn hãy đóng package lại sau đó thử nhấn chuột phải trên TAB Component Palette xem sao. Chắc bạn sẽ ngạc nhiên vì thấy sự xuất hiện của một mục chọn mới với tên là *Multi-Lines*. Hãy nhấn mục chọn này và quan sát sự khác biệt. (Xem hình 5)



Hình 5. Minh họa kết quả

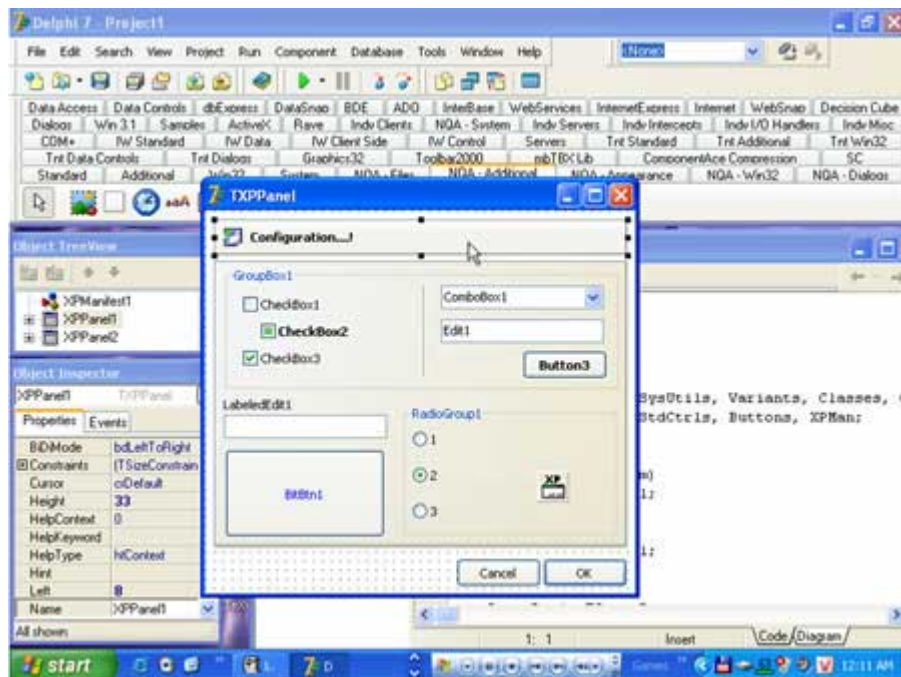
Nếu tình ý một chút chắc các bạn có thể dễ dàng nhận ra Delphi IDE của tôi được hỗ trợ theo Style XP (khi chạy trên nền Windows XP). Để làm được điều này, rất đơn giản các bạn hãy tạo một file tên ***delphi32.exe.manifest*** với nội dung như sau:

Ngo Quoc Anh

```
language="*" />
```

Sau đó lưu cùng thư mục với file ***delphi32.exe*** là được (xem hình 6).





Hình 6: Minh họa Style XP cho Delphi IDE

Bài viết này thực sự mới chỉ dừng lại ở giới thiệu một số mẹo nhỏ để tùy biến Delphi IDE. Hy vọng tôi sẽ có dịp khác trình bày các thủ thuật hay hơn trong lập trình cho Delphi IDE.

```
ancement;
```

```
Messages, SysUtils, Classes, Graphics, Controls, ExtCtrls,
Dialogs, ComCtrls, Menus, Registry;
```

```
Object = class(TComponent)
```

```
CustomForm;
```

```
rol: TTabControl;
```

```
ne : Boolean;
```

```
ntPaletteMenu : TPopupMenu;
```

```
ục chọn cho menu popup mà ta thêm vào
```

```
neItem, SeperatorItem : TMenuItem;
```

```
re UpdateOtherWindows(OldHeight: Integer);
```

```
re ResizeMultiLineComponentPalette(Sender : TObject);
```

```
n GetIdeMainForm: TCustomForm;
```

```
n GetTabControl: TTabControl;
```

```
n GetComponentPalettePopupMenu: TPopupMenu;
```

```
re OnMenuPopup(Sender: TObject);
```

```
re OnMultiLineItemClick(Sender : TObject);
```

```
re SetMultiLineComponentPalette(_multiLine : Boolean);
```

```
re CreateMenuItem( _multiLine : Boolean);
```

```
re DestroyMenuItem;
```

```
re SaveSettings;
```

```

ctor Create(AOwner: TComponent); override;
tor Destroy; override;

object : TMyExpertObject;

tion

ong thức sẽ được Delphi gọi mỗi khi component được nạp. Chúng ta cần
thức này để đọc thuộc tính MultiLines trong Registry}

    TMyExpertObject.Create;

/
istry.Create do

ey := HKEY_CURRENT_USER;
enKey(\Software\Ngô Quốc Anh, False) then
KeyExists(MultiLines) then
ultiLine := ReadBool(MultiLines);

ong thức sẽ được Delphi gọi mỗi khi component giải phóng}

TMyExpertObject.Destroy;

/

ong tin về MultiLines trong Registry mỗi khi có sự thay đổi}
MyExpertObject.SaveSettings;

istry.Create do

ey := HKEY_CURRENT_USER;
enKey(\Software\Ngô Quốc Anh, True) then
teBool(MultiLines, MultiLine)

lại kích thước của điều khiển TAB mỗi khi có sự thay đổi}
MyExpertObject.ResizeMultiLineComponentPalette(Sender: TObject);

Integer;

er as TTabControl do

nt := Height - ( DisplayRect.Bottom - DisplayRect.Top ) + 29;
raints.MinHeight := AHeight;
der as TTabControl).Parent as TWinControl).Constraints.MaxHeight :=

lại vị trí của 2 cửa sổ TObjectTreeView và TEditWindow mỗi khi thay
tróc của FORM chính}
MyExpertObject.UpdateOtherWindows(OldHeight: Integer);

s : array[0..1] of string = (TObjectTreeView, TEditWindow);

```

```

CustomForm;
nTop, HeightDelta : Integer;

GetIdeMainForm;
= nil then Exit;
elta := AForm.Height - OldHeight;
Delta = 0 then Exit;
:= AForm.Top;
Low(WinClasses) to High(WinClasses) do

qua tất cả các cửa sổ

:= 0 to Screen.CustomFormCount - 1 do
in

ều tìm được thì tiến hành thay đổi kích thước

f Screen.CustomForms[J].ClassNameIs(WinClasses[I]) then
begin
    AForm := Screen.CustomForms[J];
    AForm.Top := AForm.Top + HeightDelta;
    AForm.Height := AForm.Height - HeightDelta;
end;
/

```

chính của Delphi}

```

yExpertObject.GetIdeMainForm: TCustomForm;

TForm(Application.FindComponent(AppBuilder));

hiển TAB Component Palette}
yExpertObject.GetTabControl : TTabControl;

: TCustomForm;

nil;
:= GetIdeMainForm;
rm <> nil then
:= TTabControl(MainForm.FindComponent(TabControl))

opup cho điều khiển TAB Component Palette}
yExpertObject.GetComponentPalettePopupMenu : TPopupMenu;

: TCustomForm;

nil;
:= GetIdeMainForm;
rm <> nil then
:= TPopupMenu(MainForm.FindComponent(PaletteMenu));

```

o phương thức popup của menu popup của điều khiển TAB. Chúng tôi  
 ất mã lệnh nào cho sự kiện này. Điều này phụ thuộc vào ý chủ quan của

```

MyExpertObject.OnMenuPopup(Sender: TObject);

    {ở sự kiện OnClick của mục chọn mới trong menu popup của TAB}
    MyExpertObject.OnMultiLineItemClick(Sender: TObject);

    if Sender is TMenuItem then
    begin
        MultiLine := not (Sender as TMenuItem).Checked;
        {đổi trạng thái Checked của mục chọn}
        (Sender as TMenuItem).Checked := MultiLine;
        {đặt lại và ghi lại trạng thái vào Registry}
        MyExpertObject.OnMultiLineComponentPalette(MultiLine);
        MyExpertObject.Settings;

        {tạo menu popup của điều khiển TAB Component Palette}
        MyExpertObject.CreateMenuItem(_multiLine : Boolean);

        PaletteMenu := TPopupMenu.Create(nil);
        PaletteMenu.OnPopup := OnMenuPopup;
        PaletteMenu := GetComponentPalettePopupMenu;

        {nếu sự tồn tại của mục chọn trước, nếu chưa tồn tại thì tạo mới}
        if ComponentPaletteMenu.Items.Find(&Multi-Lines) = nil then
        begin
            SeparatorItem := TMenuItem.Create(nil);
            SeparatorItem.Caption := '-';

            {chèn thanh phân cách}
            ComponentPaletteMenu.Items.Add(SeparatorItem);

            MultiLineItem := TMenuItem.Create(nil);
            MultiLineItem.Checked := _multiLine;
            MultiLineItem.OnClick := OnMultiLineItemClick;
            MultiLineItem.Caption := &Multi-Lines;

            {chèn mục chọn với tên Multi-Lines}
            ComponentPaletteMenu.Items.Add(MultiLineItem);

            {đóng menu popup mỗi khi Component được giải phóng}
            MyExpertObject.DestroyMenuItem;

            MultiLineItem := nil;
            MultiLine := false;
        end;
    end;

```

```

ComponentPaletteMenu.Items.Find(&Multi-Lines);
if nil then

    Pos := ComponentPaletteMenu.Items.IndexOf(MI);
    ComponentPaletteMenu.Items.Delete(Pos - 1);
    ComponentPaletteMenu.Items.Delete(Pos - 1);

    thuộc tính Multi-Lines}
MyExpertObject.SetMultiLineComponentPalette(_multiLine : Boolean);

    _multiLine : Integer;

    if not TIDEMainForm then
        nil then

        OldHeight := App.Height;
        TabControl := GetTabControl;
        if TabControl <> nil then
            begin
                TabControl.MultiLine := _multiLine;
                if _multiLine then
                    begin
                        TabControl.OnResize := ResizeMultiLineComponentPalette;
                        TabControl.OnResize(TabControl);
                        CreateMenuItem(_multiLine);
                    end
                else
                    TabControl.OnResize := nil;
                    UpdateOtherWindows(OldHeight);
                end
            end
        Invalidate;

        { khi Component được nạp}
        on

        Expert := TMyExpertObject.Create(nil);
        Expert.CreateMenuItem(MyExpertObject.MultiLine);
        Expert.SetMultiLineComponentPalette(MyExpertObject.MultiLine);

        { khi Component bị hủy}
        on

        Expert.SetMultiLineComponentPalette(False);
        Expert.DestroyMenuItem;
        Expert.Free;
    end
end

```

**Ngô Quốc Anh**  
 ĐHKHTN, ĐHQG Hà Nội  
 Email: bookworm\_vn@yahoo.com

# Thuộc tính của .NET

**Thuộc tính là một trong những khái niệm quan trọng nhất của .NET, nó ảnh hưởng đến nhiều phương diện khác nhau của một ứng dụng .NET như khả năng giao tiếp với các thành phần COM, khả năng tạo ra trình dịch vụ, tính năng bảo mật, tính năng lưu dữ liệu của đối tượng vào tập tin...**

## **Thuộc tính là gì?**

Sức mạnh của .NET (so với các đời trước) có được phần lớn là do ý tưởng về thông tin mô tả (metadata) đem lại. Chính những thông tin này đã giúp cho các assembly tự mô tả đầy đủ chính nó, nhờ đó việc giao tiếp và sử dụng lại các chương trình viết bằng những ngôn ngữ khác nhau cũng trở nên dễ dàng, hiệu quả hơn. Việc lập trình tất nhiên cũng đơn giản hơn! Làm sao cung cấp những thông tin này? Câu trả lời là: dùng thuộc tính.

Thuộc tính là những đối tượng chuyên dùng để cung cấp thông tin mô tả cho các phần tử trong một assembly .NET. Phần tử ở đây bao gồm assembly, lớp, các thành viên của lớp (gồm hàm tạo, hàm thuộc tính, trường, hàm chức năng, tham biến, giá trị trả về), sự kiện.

## **Cách sử dụng thuộc tính trong C#**

Có một số qui tắc bắt buộc phải tuân theo khi dùng thuộc tính để viết mã chương trình:

- ***Thuộc tính phải đặt trong dấu ngoặc vuông.***

Ví dụ: Khi bạn tạo ra một ứng dụng loại Console trong VS.NET IDE, bạn sẽ thấy hàm Main được áp dụng thuộc tính STAThread như sau:

```
[STAThread]

static void Main(string[] args) {

    ...

}
```

- ***Tên các lớp thuộc tính thường có đuôi là "Attribute" nhưng bạn có thể không ghi đuôi này.***

Ví dụ: Hãy thử đổi [STAThread] thành [STAThreadAttribute] và biên dịch chương trình. Bạn sẽ thấy không có lỗi gì xảy ra.

- ***Thuộc tính có thể có nhiều biến thể ứng với nhiều bộ tham biến khác nhau.*** Khi cần truyền tham số cho thuộc tính, ghi chúng trong cặp ngoặc đơn. Riêng đối với biến thể không tham biến, có thể ghi hoặc không ghi cặp ngoặc rỗng "()". Ngoài ra, các tham số phải là các biểu thức hằng, biểu thức typeof hay biểu thức tạo mảng (như new Type[] {typeof(TargetException)}).

Ví dụ 1: Có thể thay [STAThread] bằng [STAThread()].

Ví dụ 2: Khi cần đánh dấu một lớp, hàm là "đã cũ, cần dùng phiên bản thay thế", ta có thể dùng thuộc tính [\*ObsoleteAttribute\*](#). 1 trong 3 biến thể của thuộc tính này là:

```
[Obsolete(string message, bool error)]
```

*Trong đó:* message dùng để cung cấp thông tin chỉ dẫn về lớp, hàm thay thế. error dùng để hướng dẫn cho

trình biên dịch biết cần làm gì khi biên dịch lớp, hàm sử dụng phần tử được áp dụng *Obsolete*. Nếu error bằng true, trình biên dịch báo lỗi và không biên dịch. Ngược lại, trình biên dịch chỉ cảnh báo và vẫn biên dịch bình thường.

Như vậy, ta có thể sử dụng như sau:

```
[Obsolete("Nên dùng lớp NewClass", false)]

public class OldClass{

    ...

}

// lớp này không được áp dụng thuộc tính Obsolete

public class ClientClass{

private OldClass a = new OldClass();

    ...

}
```

Khi biên dịch lớp ClientClass, VS.NET IDE sẽ thông báo ở cửa sổ Task List như *hình 1*:



Hình 1

Nếu bạn sửa false thành true thì bạn sẽ thấy bảng báo lỗi như *hình 2*:



### Ví dụ 3: không thể dùng

```
private string s = "Nên dùng lớp NewClass";

[Obsolete(s, false)]
```

Nhưng nếu thêm const vào phần khai báo của s thì hợp lệ.

• **Thuộc tính có mục tiêu áp dụng** (do người viết ra thuộc tính qui định) xác định nên vị trí đặt cũng bị hạn chế. Nói chung, thuộc tính phải đặt trước mục tiêu áp dụng và không thể đứng bên trong thân hàm. Nếu thuộc tính có nhiều mục tiêu áp dụng được thì có thể chỉ định mục tiêu cụ thể bằng một trong các từ khoá: assembly, module, type, event, field, property, method, param, return.

Ví dụ:

```
[assembly: AssemblyTitle("Demo")] // Đúng chỗ

namespace Demo;

[assembly: AssemblyTitle("Demo")] // Sai chỗ

[type: Obsolete] // Đúng chỗ

// [method: Obsolete] // Sai chỗ

public class OldClass{

    [type: Obsolete] // Sai chỗ

    ...

}

}
```

• **Thuộc tính có thể đặt trong các cặp ngoặc vuông liên tiếp nhau** hay đặt trong cùng một cặp ngoặc vuông nhưng cách nhau bởi dấu phẩy.

Ví dụ:

```
[type: Obsolete("Nên dùng lớp NewClass", false), Serializable]

tương đương với

[type: Obsolete("Nên dùng lớp NewClass", false)]

[Serializable]
```

• **Có những thuộc tính có thể được áp dụng nhiều lần cho cùng một mục tiêu.** Điều này cũng do người viết ra thuộc tính qui định.

Ví dụ 1:

```
// Trình biên dịch sẽ báo lỗi "Duplicate Obsolete attribute"

[type:Obsolete]

[type:Obsolete]

public class OldClass{

    ...

}
```



## Ví dụ 2:

```
// Trình biên dịch không báo lỗi

// Thuộc tính ExpectedException ở đây là thuộc tính custom mà ta sẽ tự tạo trong phần 5-

[type: ExpectedException( typeof(XXXException) )]

[type: ExpectedException( typeof(XXXException) )]

public class OldClass{
...
}
```

• **Một số thuộc tính có tính kế thừa.** Khi bạn áp dụng những thuộc tính này cho một lớp nào đó, hãy nhớ là các lớp con của lớp đó cũng mặc nhiên được áp dụng các thuộc tính đó. Bạn sẽ thấy rõ điều này trong phần "Tạo một thuộc tính custom".

• **Cuối cùng, khi sử dụng thuộc tính nào, nhớ tạo ra tham chiếu tới không gian kiểu chứa nó.** Chẳng hạn như, để dùng các thuộc tính như AssemblyTitle, AssemblyVersion, cần thêm:

```
using System.Reflection;
```

## Đặc điểm của thuộc tính

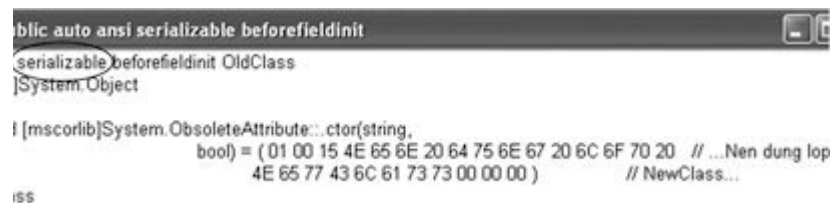
**1. Khi thêm thuộc tính vào mã chương trình,** ta đã tạo ra một đối tượng mà các thông tin của nó sẽ được lưu vào assembly chứa mục tiêu áp dụng của thuộc tính. Tùy theo thuộc tính thuộc loại custom hay p-custom (p- là pseudo) mà những thông tin này sẽ được lưu thành chỉ thị .custom hay khác (.ver, .hash, serializable,...) trong tập mã IL.

Ví dụ: lớp OldClass sau sẽ có mã IL (xem bằng ILDasm.exe) như hình 3:

```
[Obsolete("Nên dùng lớp NewClass", false)]

[Serializable]

public class OldClass{
...
}
```



• **Tuy được lưu trong assembly nhưng thuộc tính hoàn toàn không ảnh hưởng gì đến các mã lệnh khác.** Thuộc tính chỉ có ý nghĩa khi có một chương trình nào đó cần đến và truy xuất nó thông qua tính năng Reflection của .NET. Dĩ nhiên, ý nghĩa của thuộc tính sẽ do chương trình đó qui định. Điều đó cũng có nghĩa là cùng

một thuộc tính nhưng "dưới mắt" các chương trình đọc khác nhau sẽ có thể có công dụng khác nhau. Đây là đặc điểm đáng chú ý nhất của thuộc tính.

Ví dụ: thuộc tính `Obsolete` được trình biên dịch dùng để phát hiện những phần tử sẽ không được sử dụng nữa, `[TestFixture]` được `JUnit` dùng để chọn những lớp có chứa hàm kiểm tra cần được kích hoạt tự động,...

- *Dữ liệu chỉ định trong thuộc tính gắn chặt với mục tiêu áp dụng của thuộc tính* chứ không lỏng lẻo và do đó không linh hoạt như dữ liệu trong tập tin cấu hình. Nhờ vậy, dữ liệu mô tả lưu bằng thuộc tính an toàn hơn, khó sửa hơn.

- *Thuộc tính còn có những đặc điểm khác như:* có mục tiêu áp dụng xác định, có khả năng áp dụng nhiều lần cho cùng một mục tiêu, có thể được thừa kế.

## Một số ví dụ minh họa ứng dụng của thuộc tính

### *a - Thuộc tính `CLSCompliant`:*

Mục tiêu của .NET là tạo ra một nền tảng giao tiếp thống nhất giữa nhiều ngôn ngữ lập trình khác nhau. Để đạt được điều đó, .NET định ra 2 chuẩn là CTS và CLS, trong đó CTS bao gồm các kiểu cơ bản mà một ngôn ngữ .NET có thể chọn hỗ trợ còn CLS là một tập các qui tắc bắt buộc mọi ngôn ngữ .NET phải áp dụng cho các phần tử dùng để giao tiếp với nhau. Như vậy, một ngôn ngữ có thể hỗ trợ những kiểu mà ngôn ngữ khác không hỗ trợ. Kết quả là khi các ngôn ngữ muốn phối hợp với nhau thì những kiểu "không chung" này sẽ "phá đám", gây ra hiểu nhầm. Để tránh tình huống này, .NET tạo ra thuộc tính `CLSCompliant` dùng để nhờ trình biên dịch theo dõi và cảnh báo xem có phần tử nào vi phạm luật CLS hay không. Thuộc tính này có mục tiêu áp dụng là mọi phần tử.

#### Ví dụ:

```
// Kiểm tra xem mọi phần tử của assembly này có tương thích CLS không
```

```
[assembly: CLSCompliant(true)]
```

```
namespace Demo{
```

```
// Riêng: bỏ qua các phần tử của lớp này
```

```
[type: CLSCompliant(false)]
```

```
public class A{
```

```
private uint a;
```

```
public uint b;
```

```
}
```

```
public class B{
```

```
private uint a; // không bị coi là vi phạm vì có tầm vực private
```

```
public uint b; // vi phạm
```

```
}
```

```
}
```

### *b - Các thông tin mô tả về assembly:*

Khi sử dụng VS.NET IDE để tạo một dự án, bạn sẽ thấy là luôn có một tập tin tên AssemblyInfo.xx (tùy theo ngôn ngữ, xx có thể là cs với C#, vb với VB.NET,...). Sau đây là nội dung tập tin AssemblyInfo.cs đã lược bỏ phần chú thích:

```
using System.Reflection;

using System.Runtime.CompilerServices;

[assembly: AssemblyTitle("")]

[assembly: AssemblyDescription("")]

[assembly: AssemblyConfiguration("")]

[assembly: AssemblyCompany("")]

[assembly: AssemblyProduct("")]

[assembly: AssemblyCopyright("")]

[assembly: AssemblyTrademark("")]

[assembly: AssemblyCulture("")]

[assembly: AssemblyVersion("1.0.*")]

[assembly: AssemblyDelaySign(false)]

[assembly: AssemblyKeyFile("")]

[assembly: AssemblyKeyName("")]
```

(Lưu ý: Có thể bạn ngộ nhận tập tin trên là bắt buộc phải có. Nhưng không, nó chẳng qua là một công cụ mà VS.NET cung cấp, giúp bạn tập trung các thông tin chung về assembly lại một chỗ. Bạn hoàn toàn có thể xóa bỏ tập tin trên và tạo lại các mục tương tự nhưng để rải rác ở các tập tin trong dự án.)

Như bạn thấy, tập tin trên chỉ chứa toàn các thuộc tính với mục tiêu áp dụng là assembly. Những thuộc tính ấy nằm trong 2 không gian kiểu `System.Reflection` và `System.Runtime.CompilerServices`. 8 thuộc tính đầu dùng để cung cấp các thông tin chung về assembly (có thể xem những thông tin này bằng *ILDasm.exe* hay *Windows Explorer*). *AssemblyVersion* dùng để ghi nhận số phiên bản cho assembly, số này sẽ được CLR cần đến. Cụ thể là nếu assembly A tham chiếu đến assembly B thì trong assembly A sẽ ghi nhận phiên bản của B mà A tham chiếu. Nhờ đó, khi CLR cần tải B để hỗ trợ cho A thì CLR có thể biết được và tải đúng phiên bản thích hợp của B.

*AssemblyKeyFile* dùng để chỉ định tập tin chứa cặp khóa chung/riêng mà trình biên dịch sẽ dựa vào để tạo ra assembly duy nhất. Nếu không dùng *AssemblyKeyFile* thì có thể dùng *AssemblyKeyName* thay thế, chỉ khác là cần chỉ định tên của khóa đã được cài đặt vào Crypto Service Provider trên máy. Cũng có thể dùng cùng lúc cả 2 thuộc tính để chỉ định khóa; khi ấy, *AssemblyKeyName* sẽ được ưu tiên dùng trước.

Cuối cùng, *AssemblyDelaySign* dùng để yêu cầu trình biên dịch tạo ra một assembly giả duy nhất (vì chỉ cần dựa vào khóa chung) giúp cho việc thử nghiệm dễ dàng hơn. Đến khi cần triển khai ứng dụng thực sự mới phải dùng khóa riêng để tạo ra assembly duy nhất. Nhờ có *AssemblyDelaySign*, khóa riêng có thể được giữ bí mật bởi một người nào đó mà không làm ảnh hưởng đến quá trình phát triển phần mềm chung của cả nhóm.

## Tạo một thuộc tính custom

Trong các phần trước, chúng ta đã sử dụng các thuộc tính có sẵn của .NET. Trong phần này, chúng ta sẽ tìm hiểu cách tự tạo lấy các thuộc tính cho riêng mình "xài" thông qua quá trình xây dựng thuộc tính `ExpectedException`.

Cũng như những thuộc tính custom có sẵn, thuộc tính tự tạo của chúng ta phải là một lớp con của lớp `System.Attribute`:

```
// Theo qui ước, tên thuộc tính nên có đuôi là Attribute

public class ExpectedExceptionAttribute : System.Attribute{

    ...

}
```

Thuộc tính tự tạo có thể có các hàm tạo và hàm thuộc tính như một lớp thông thường:

```
...

private Type expected = null;

private string msg = "";

public ExpectedExceptionAttribute(Type expectedType):this(expectedType, "") {}

public ExpectedExceptionAttribute(Type expectedType, string message){

    if (expectedType as Exception == null)

        throw ...

    expected = expectedType;

    msg = message;

}

public Type ExpectedType{

    get{

        return expected;

    }

}

public string Message{

    get{

        return msg;

    }

    set{

        msg = value;

    }

}
```

```

}

}

...

```

Khi sử dụng, các tham biến của hàm tạo trở thành các tham số vị trí (tức là bắt buộc có và được truyền theo đúng thứ tự khai báo), còn các hàm thuộc tính trở thành tham số có tên (tức không bắt buộc có và có thể được truyền theo thứ tự tùy ý, miễn là phải sau các tham số vị trí). Sau đây là một số cách dùng hợp lệ:

```

[ExpectedException(typeof(Exception))]

[ExpectedException(typeof(Exception), "Expected type: System.Exception")]

[ExpectedException(typeof(Exception), Message="Expected type: System.Exception")]

[ExpectedException(typeof(Exception), "Expected type: System.Exception"),
Message="Expected type: System.Exception")]

```

Thuộc tính của ta chỉ cần áp dụng cho hàm tạo, hàm chức năng, hàm thuộc tính. Do đó, ta cần chỉ định mục tiêu áp dụng cho nó thông qua thuộc tính `AttributeUsage` như sau:

```

[AttributeUsage(AttributeTargets.Constructor | AttributeTargets.Method |
AttributeTargets.Property)]

public class ExpectedExceptionAttribute: System.Attribute {...}

```

Mặt khác, một hàm có thể phát ra nhiều lỗi khác nhau, tức là thuộc tính `ExpectedException` có thể áp dụng nhiều lần cho cùng một mục tiêu. Ta chỉ định thêm:

```

[AttributeUsage(..., AllowMultiple = true)]

```

Cuối cùng, ta muốn rằng nếu các hàm virtual của lớp A được áp dụng thuộc tính `ExpectedException` thì các hàm override tương ứng của lớp con của A cũng kế thừa thuộc tính này. Do đó ta thêm:

```

[AttributeUsage(..., ..., Inherited = true)]

```

Xin lưu ý là chỉ khi cả `AllowMultiple` và `Inherited` đều bằng `true` thì lớp con mới được kế thừa toàn bộ thuộc tính với cùng giá trị đã áp dụng cho lớp cha.

Đến đây coi như ta đã hoàn tất phần định nghĩa thuộc tính. Ta đặt thuộc tính vừa tạo vào assembly tên `DemoAttrLib.dll`. Tiếp đến, ta xây dựng một chương trình sử dụng `ExpectedExceptionAttribute`. Ta đặt chương trình này trong assembly `DemoAttrClient.exe`.

```

/* Chương trình này gồm 2 lớp là DemoParentClient và DemoChildClient */

using System;

using System.Reflection;

using DemoAttrLib;

namespace DemoAttrClient
{

public class DemoParentClient

```

```

{
[method:ExpectedException(typeof(TargetException)) ]
public DemoParentClient() {...}

[method:ExpectedException(typeof(ArgumentException)) ]
[method:ExpectedException(typeof(TargetException)) ]
public void TestMethod1() {...}

[method:ExpectedException(typeof(TargetException)) ]
public virtual void TestMethod2() {...}
}

class DemoChildClient:DemoParentClient
{
[method:ExpectedException(typeof(ArgumentException)) ]
public override void TestMethod2() {...}

[method:ExpectedException(typeof(ArgumentException)) ]
public new void TestMethod1() {...}
}
}

```

Như đã nói, một thuộc tính chỉ có giá trị khi một chương trình nào đó dùng đến nó. Chương trình này sẽ dùng các lớp trong không gian kiểu System.Reflection để kiểm tra các thuộc tính đi kèm từng phần tử trước khi ra quyết định xử lý thích hợp đối với phần tử đó. Dưới đây là một ví dụ đơn giản về chương trình như thế (trong assembly DemoAttrReader.exe):

```

/* Đây là chương trình loại Console dùng để liệt kê các hàm được áp dụng thuộc tính
ExpectedException trong assembly chỉ định. */

using System;

using System.Reflection;

using DemoAttrLib;

namespace DemoAttrReader
{
class DemoReader
{
// Hàm này trả về một chuỗi chứa thông tin báo cáo về mọi hàm được áp dụng thuộc tính
ExpectedException trong assembly chỉ định.

public static string Read(string assemblyName)

```

```

{...}

// Hàm này trả về một chuỗi chứa thông tin báo cáo về mọi hàm được áp dụng thuộc tính
ExpectedException trong kiểu chỉ định.

private static string AttrRead(Type t)

{...}

[STAThread]

static void Main(string[] args)

{

if (args.Length != 1)

{

Console.WriteLine("Hay chỉ định một assembly nào đó.");

return;

}

Console.WriteLine("BAO CAO:");

Console.WriteLine(DemoReader.Read(args[0]));

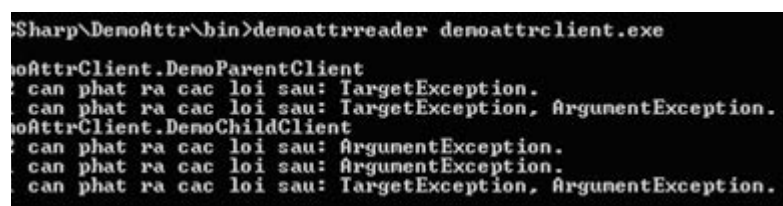
}

}

}

```

Kết quả chạy chương trình như ở hình 4- (chú ý là có tới 2 hàm *TestMethod1* đối với lớp *DemoChildClient*):



```

Sharp\DemoAttr\bin>demoattrreader demoattrclient.exe
DemoAttrClient.DemoParentClient
can phát ra các lỗi sau: TargetException.
can phát ra các lỗi sau: TargetException, ArgumentException.
DemoAttrClient.DemoChildClient
can phát ra các lỗi sau: ArgumentException.
can phát ra các lỗi sau: ArgumentException.
can phát ra các lỗi sau: TargetException, ArgumentException.

```

Nếu bạn muốn có một ví dụ phức tạp hơn, mời bạn tham khảo mã nguồn của NUnit ([www.nunit.org](http://www.nunit.org)), chương trình kiểm tra tự động khá thông dụng với các lập trình viên .NET. Cách hoạt động của NUnit đơn giản là: dò trong assembly chỉ định những lớp nào có thuộc tính *TestFixture* và kích hoạt những hàm được đánh dấu bằng thuộc tính *Test*, *SetUp*, *TearDown*,... trong các lớp ấy.

Vậy là chúng ta đã cơ bản tìm hiểu xong về thuộc tính của .NET. Hy vọng những điều vừa trình bày sẽ giúp ích cho các bạn trong công việc lập trình của mình.

**Nguyễn Phương**

**Email:** [hungphung@hcm.fpt.vn](mailto:hungphung@hcm.fpt.vn)

-----

### Tài liệu tham khảo

- "Applied .NET Attributes", tác giả: Tom Barnaby và Jason Bock, NXB: Appress.
- "C# Attributes", "Extending Metadata Using Attributes" và các bài liên quan trong thư viện MSDN.

# OBEX và kỹ thuật lập trình cho cổng hồng ngoại, Bluetooth

Chúng ta đang sống trong một thế giới mà các thiết bị không dây dần len lỏi vào cuộc sống của mỗi gia đình. Từ chiếc điện thoại di động (ĐTDD) xinh xắn, thiết bị hỗ trợ cá nhân đa năng (Pocket PC, Palm...), đến các phương tiện giải trí trong gia đình như hệ thống loa, đầu DVD, tivi, tất cả được kết nối với nhau mà không cần một sợi dây nào. Nếu bạn đã từng đặt câu hỏi: "*Có thể lập trình để truyền tải dữ liệu hình ảnh, âm thanh vào ĐTDD qua cổng hồng ngoại hay Bluetooth?*" thì bài viết này sẽ giúp bạn hình dung cách thức và hướng giải quyết vấn đề trên.

## OBEX LÀ GÌ?

OBEX (OBject EXchange) là giao thức trao đổi dữ liệu giữa các thiết bị dùng cổng hồng ngoại được hiệp hội IrDA (Infrared Data Association) đưa ra lần đầu tiên năm 1997. Ban đầu, giao thức này chỉ giới hạn cho các thiết bị sử dụng môi trường ánh sáng hồng ngoại, nhưng rất nhanh sau đó nó được tổ chức Bluetooth SIG (Bluetooth Special Interest Group) đưa vào hầu hết các thiết bị Bluetooth của mình.

### 1. Vị trí OBEX trong mô hình OSI

Cũng giống như các giao thức khác, giao thức OBEX được xây dựng trên nền mô hình OSI (Open Systems Interconnection) bao gồm hai thành phần chính:

- **OBEX session protocol (giao thức phiên OBEX)**: mô tả cấu trúc gói tin trong phiên làm việc giữa hai thiết bị.
- **OBEX application framework**: tập các dịch vụ OBEX cung cấp cho các ứng dụng đầu cuối như truyền file, in ảnh...

OBEX Application			ứng dụng
OBEX Framework			
OBEX Session			hiện
			hiển
Tiny TP	TM		o vận
IrLMP	P		ạng
IrLAP	er		t dữ liệu



IrPHY	nd		ât lý
IrDA	oth		
<b>Hình 1: Giao thức OBEX trong mô hình OSI</b>			

## 2. Cấu trúc gói tin trong giao thức phiên OBEX

Giao thức OBEX được sử dụng chủ yếu trong các ứng dụng kiểu "đẩy" (Push) hoặc "kéo" (Pull), cho phép máy khách (client) "đẩy" dữ liệu lên máy chủ (server) hoặc "kéo" dữ liệu từ server xuống. Để thực hiện điều này, các gói tin trao đổi giữa client và server phải tuân thủ chặt chẽ cấu trúc đề ra. Dưới đây là một vài cấu trúc được sử dụng trong quá trình truyền file giữa client và server (chi tiết có thể tham khảo tài liệu IrOBEX1.3 trên website <http://www.hitekgroup.net>).

### 2.1 Gói tin yêu cầu

Mọi gói tin yêu cầu đều có cấu trúc như sau:

Opcode	Packet length	Packet ID
--------	---------------	-----------

*Opcode*: Mã lệnh ứng với từng yêu cầu (Bảng 1). Bit cao nhất gọi là Final bit.

*Packet length*: Độ dài của gói tin

*Header*: Thông tin đầu có cấu trúc như sau:

Header ID	Header Length	Header Data
-----------	---------------	-------------

Yêu cầu		
Header ID	Header Length	Header Data
Header ID	Header Length	Header Data
Header ID	Header Length	Header Data
Header ID	Header Length	Header Data
Header ID	Header Length	Header Data

Thông tin đầu		
Header ID	Header Length	Header Data
Header ID	Header Length	Header Data
Header ID	Header Length	Header Data

		ủa file
	Y	uối cùng của file

## 2.2 Gói tin trả lời

Giống như gói tin yêu cầu, gói tin trả lời có cấu trúc như sau:

Byte 0	Byte 1, 2	Byte 3 to n
Response opcode	Response length	Response data

Một số mã trả lời (response opcode) thường gặp:

Mã trả lời	
	Thành công hoặc trả lời
	Yêu cầu yêu cầu hoặc trả lời
	Không quyền
	Đã bị huỷ bỏ
	Không file

## 3. Cùng giải quyết

Quá trình trao đổi file giữa client và server được chia làm 3 giai đoạn:

- *Thiết lập phiên*: CONNECT
- *Nhận/gửi file*: GET/PUT
- *Ngừng phiên*: DISCONNECT

### 3.1 Thiết lập phiên (CONNECT)

Gói tin CONNECT có cấu trúc như sau:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5, 6	Byte 7 to n
	Length	Version number		OBEX packet header	Parameters

*OBEX version number*: Phiên bản giao thức OBEX gồm major number lưu tại 4 bit cao, minor number lưu tại 4 bit thấp. Phiên bản hiện tại là 1.0, do đó giá trị này là 0x10.

*Flags*: Giá trị này luôn là 0x00 trong phiên bản hiện tại.

*Maximum OBEX packet length:* Giá trị lớn nhất của gói tin trong giao thức OBEX mà thiết bị có thể nhận hoặc gửi. Giá trị này ở client và server có thể khác nhau. Do đó khi thiết lập phiên, client cần gửi giá trị này lên server để kiểm tra xem kích thước gói tin lớn nhất mà server có thể nhận hoặc gửi là bao nhiêu?

*Optional headers:* Thông tin đầu được tùy chọn ứng với mục đích của mỗi phiên giao dịch. Trong ví dụ dưới đây, giá trị này có thể bỏ qua.

từ client		Ý nghĩa
yêu cầu		
		= 7 bytes
		OBEX 1.0
		hiện bản hiện tại
		lớn nhất của gói tin là 8K

Server		
		= 7 bytes
		OBEX 1.0
		hiện bản hiện tại
		lớn nhất của gói tin mà server có gửi là 512 bytes

### 3.2 Gửi file (PUT)

Không giống như gói tin CONNECT, gói tin PUT có thêm một số thông tin đầu sau:

*NAME:* Thông tin về tên file

*LENGTH:* Thông tin về kích thước file

*BODY:* Đoạn dữ liệu file

*END OF BODY:* Đoạn dữ liệu cuối cùng của file

Dưới đây là ví dụ gửi 1 file hello.gif có kích thước 721 bytes từ client (PC) lên server (ĐTĐĐ). Do kích thước của file lớn hơn kích thước gói tin lớn nhất mà server có thể nhận (với Sony Ericsson T610 là 512 bytes) nên client sẽ chia file thành hai gói tin để gửi. Gói tin 1 có mã yêu cầu PUT là 0x02 (không thiết lập Final bit). Gói tin 2 có mã yêu cầu PUT là 0x82 (thiết lập Final bit).

ent		Ý nghĩa
		không thiết lập để chỉ cho client còn gửi yêu cầu tiếp theo
		= 482 bytes

		1g tin đầu NAME (tên file)
		ng tin đầu NAME = 20+3 =
		de) có ký tự kết thúc es)
		1g tin đầu LENGTH (kích
		ở = 721 bytes
		1g tin đầu BODY (dữ liệu
		ng tin đầu BODY = 448+3
		file có kích thước 448 bytes
<b>Server</b>		
		tiếp tục nhận yêu cầu từ
		= 3 bytes
<b>Client</b>		
		được thiết lập để chỉ cho là gói tin cuối cùng
		= 279 bytes
		1g tin đầu END OF BODY
		ng tin đầu END OF BODY
		file có kích thước 721-448 =
<b>Server</b>		
		= 3 bytes

### 3.3 Nhận file (GET)

Khác với gói tin PUT, gói tin GET chỉ có thông tin đầu NAME. Ví dụ sau sẽ mô tả quá trình nhận file hello.gif có kích thước 721 bytes từ server. Trước tiên client (PC) sẽ gửi yêu cầu GET đến server (ĐTDĐ) với thông tin đầu NAME là tên file. Do kích thước của file yêu cầu lớn hơn 512 bytes (với Sony Ericsson T610) nên gói tin trả lời đầu tiên có mã trả lời là 0x90 (CONTINUE) cùng với một phần dữ liệu của file. Khi nhận được mã trả lời là CONTINUE, client biết rằng đây chưa phải là đoạn dữ liệu cuối cùng của file nên tiếp tục gửi yêu cầu (không cần thông tin đầu NAME) cho đến khi nhận được mã trả lời là 0xA0 (SUCCESS).

ent		Ý nghĩa
		được thiết lập

		= 26 bytes
		1g tin đầu NAME (tên file)
		ng tin đầu NAME = 20+3 =
		de) có ký tự kết thúc NULL
<b>ver</b>		
		òn dữ liệu trên server
		= 454 bytes
		1g tin đầu BODY (dữ liệu
		ng tin đầu BODY = 448+3 =
		ile có kích thước 448 bytes
<b>ent</b>		
		êu cầu nhận file
		= 3 bytes
<b>ver</b>		
		ạn dữ liệu cuối cùng
		= 279 bytes
		1g tin đầu END OF BODY
		ng tin đầu END OF BODY
		ile có kích thước 273 bytes

### 3.4 Ngừng phiên (DISCONNECT)

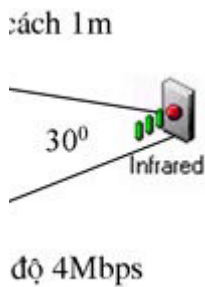
Để kết thúc phiên giao dịch, client cần gửi gói tin DISCONNECT tới server.

<b>ent</b>		
		Γ
		= 3 bytes
<b>ver</b>		
		= 3 bytes

OBEX trong điện thoại di động

Ngày nay, ĐTĐĐ không chỉ là phương tiện liên lạc đơn thuần mà còn là một thiết bị giải trí với nhiều chức năng như nghe nhạc, chụp ảnh, chơi game. Điều này đồng nghĩa với việc người dùng luôn có nhu cầu cập nhật những bản nhạc hay, những trò chơi yêu thích hoặc lưu lại những khoảnh khắc đáng nhớ trên chiếc điện thoại của mình. Dưới đây tôi xin giới thiệu hai kiểu kết nối phổ biến nhất trên hầu hết các ĐTĐĐ đời mới hiện nay là cổng hồng ngoại và Bluetooth; đồng thời hướng dẫn cách lập trình trao đổi dữ liệu giữa PC và ĐTĐĐ sử dụng giao thức OBEX qua hai loại kết nối này.

### 1. Kết nối qua hồng ngoại bằng C#



Giao thức IrDA được hiệp hội IrDA giới thiệu lần đầu tiên năm 1994 với mục đích tăng cường khả năng kết nối không dây giữa các thiết bị qua ánh sáng hồng ngoại. Với phạm vi hoạt động lên tới 1 m, góc mở từ 15 đến 30 độ, tốc độ có thể đạt 4Mbps, cổng hồng ngoại nhanh chóng được đưa vào trong hầu hết các thiết bị không dây như ĐTĐĐ, PDA...

Trước đây, việc lập trình với cổng hồng ngoại là một rào cản đối với những ai chưa quen với giao diện lập trình API (Application Programming Interface) của Windows, còn ngày nay, với phiên bản .NET 2.0, Microsoft đã đưa vào bộ Framework này lớp thư viện IrDA, cho phép người lập trình viết mã dễ dàng và nhanh chóng hơn. Giống như giao thức TCP/IP, client có thể thiết lập kết nối IrDA tới server bằng việc chỉ ra địa chỉ của server (tương tự như địa chỉ IP) và tên dịch vụ trên server (tương tự như TCP Port).

Mỗi chiếc ĐTĐĐ đều được gán một địa chỉ duy nhất tương ứng với cổng hồng ngoại trên đó. Đoạn mã sau cho phép xác định địa chỉ này:

```
using System.Net.Sockets;

void Form1_Load(object sender, EventArgs e)
{
    /* Khởi tạo client */
    IrDAClient irClient = new IrDAClient();

    /* Tìm kiếm tối đa 2 thiết bị */
    IrDADeviceInfo[] irDevices = irClient.DiscoverDevices(2);

    /* In thông báo khi không tìm thấy thiết bị nào */
    if (irDevices.Length == 0) {
        Console.WriteLine("Không tìm thấy thiết bị hồng ngoại");
    }
    else {
        /* In tên và địa chỉ của từng thiết bị tìm thấy */
        for (int i = 0; i < irDevices.Length; i++) {
            Console.WriteLine("Device Name:{0}", irDevices[i].DeviceName);
            Console.WriteLine("Device ID:{0}", irDevices[i].DeviceID);
        }
    }
}
```



Các dịch vụ IrDA được xây dựng sẵn (built-in) trong ĐTDĐ có thể khác nhau mỗi nhà sản xuất. Tuy nhiên, nhìn chung hầu hết các ĐTDĐ hiện nay đều cung cấp hai dịch vụ chính là: IrDA:IrCOMM và IrDA:OBEX. Trong phạm vi bài viết này tôi chỉ đề cập đến dịch vụ IrDA:OBEX, dịch vụ cho phép PC và ĐTDĐ trao đổi dữ liệu qua giao thức OBEX. Việc kết nối tới dịch vụ này được thực hiện thông qua đoạn mã dưới đây:

```
using System.Net;

using System.Net.Sockets;

void Form1_Load(object sender, EventArgs e)

{

    ...

    /* Thiết lập EndPoint */

    IrDAEndPoint irEndPoint = new IrDAEndPoint(irDevices[0].DeviceID, "IrDA:OBEX");

    /* Khởi tạo socket */

    Socket irSocket = new Socket(AddressFamily.Irda,
    SocketType.Stream, ProtocolType.Unspecified);

    /* Kết nối tới ĐTDĐ qua dịch vụ OBEX*/

    irSocket.Connect(irEndPoint);

}
```



Như vậy, từ giờ chúng ta có thể trao đổi dữ liệu giữa PC và ĐTDĐ qua IrSocket bằng việc push/pull những gói tin OBEX thích hợp như đã đề cập ở phần trên.

## 2. Kết nối qua Bluetooth bằng VC++

Năm 1994, hãng cung cấp thiết bị viễn thông hàng đầu thế giới Ericsson đã nghiên cứu thành công công nghệ không dây cho phép ĐTDĐ có thể kết nối với các phụ kiện như tai nghe, microphone qua sóng radio. Sau đó 4 năm, tổ chức Bluetooth SIG được thành lập (bao gồm Ericsson, Intel, IBM, Nokia và Toshiba) đã chính thức đưa ra đặc tả kỹ thuật phiên bản 1.0A cho công nghệ Bluetooth vào năm 1999. Bluetooth hay còn có cái tên IEEE 802.15.1 hoạt động ở tần số 2,4 GHz, phạm vi phủ sóng lên tới 100 m (Class 1), tốc độ có thể đạt 3Mbps đối với phiên bản 2.0+EDR (Enhanced Data Rate).

Kể từ phiên bản Windows XP SP1, Microsoft đã đưa vào hệ điều hành của mình mô hình lập trình Microsoft Bluetooth Stack cho phép kết nối với thiết bị Bluetooth thông qua Bluetooth socket. Một vấn đề nảy sinh là không phải tất cả các chipset Bluetooth đều hỗ trợ Microsoft Bluetooth Stack, mà phần lớn phải có driver đi kèm cũng như bộ SDK (Software Development Kit) riêng để phát triển. Chúng ta có thể tìm thấy chipset Bluetooth hỗ trợ Microsoft Bluetooth Stack trên các máy tính xách tay SONY VAIO, trong khi dòng IBM lại hỗ trợ Widcomm Bluetooth Stack (bộ SDK có giá tới 1400 USD).

Để lập trình với Bluetooth socket chúng ta phải cài bộ SDK for Windows XP SP2

(<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/XPSP2FULLInstall.htm>). Bộ SDK này sẽ cung cấp một số file header cũng như các thư viện cần thiết trong quá trình lập trình. Do phiên bản .NET 2.0 chưa hỗ trợ Bluetooth nên việc viết mã phải hoàn toàn thực hiện trên nền thư viện API có sẵn của Windows.

Cách tiếp cận thiết bị Bluetooth hoàn toàn giống như thiết bị hồng ngoại, chỉ có điều, thay vì sử dụng hàm có sẵn của .NET, bạn phải tự thân vận động. Dưới đây là đoạn mã viết bằng VC++ 2005 cho phép tìm kiếm thiết bị Bluetooth, đồng thời in ra tên và địa chỉ của mỗi thiết bị tìm thấy:

```
#include <winsock2.h>

#include <Ws2bth.h>

#include <BluetoothAPIs.h>

#pragma comment(lib, "ws2_32.lib")

#pragma comment(lib, "irprops.lib")

/* compile with: /clr */

using namespace System;

void main()

{

    WORD wVersionRequested = 0x202;

    WSADATA m_data;

    /* Khởi tạo Windows Socket */

    if (WSAStartup(wVersionRequested, &m_data) == 0) {

        /* Thiết lập thông số tìm kiếm */

        WSAQUERYSET querySet;

        memset(&querySet, 0, sizeof(querySet));

        querySet.dwSize = sizeof(querySet);

        /* Xác lập phạm vi tìm kiếm là các thiết bị Bluetooth */

        querySet.dwNameSpace = NS_BTH;

        HANDLE hLookup;

        /* Thiết lập các thông tin trả về */

        DWORD flags = LUP_RETURN_NAME | LUP_CONTAINERS | LUP_RETURN_ADDR | LUP_FLUSHCACHE | LUP_RETURN_BLOB;

        /* Tìm kiếm tối đa 10 thiết bị */

        int maxDevices = 10;

        /* Bắt đầu quá trình tìm kiếm */

        int result = WSALookupServiceBegin(&querySet, flags, &hLookup);
```



```

while (count < maxDevices && result == 0) {

BYTE buffer[1000];

DWORD bufferLength = sizeof(buffer);

WSAQUERYSET *pResults= (WSAQUERYSET*)&buffer;

result=WSALookupServiceNext(hLookup, flags, &bufferLength,
pResults);

/* In tên và địa chỉ của từng thiết bị tìm thấy */

if (result == 0) {

CSADDR_INFO *pCSAddr = (CSADDR_INFO*)pResults->lpcsaBuffer;

SOCKADDR_BTH *bts=(SOCKADDR_BTH*)pCSAddr->RemoteAddr.lpSockaddr;

Console::WriteLine(pResults->lpszServiceInstanceName);

Console::WriteLine("Device ID:{0}", bts->btAddr);

count++;

}

}

/* Kết thúc quá trình tìm kiếm */

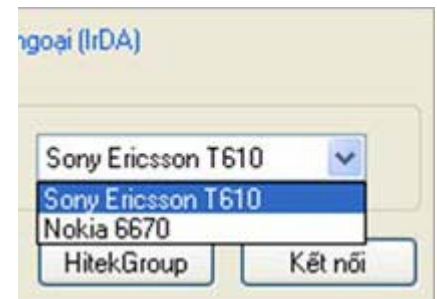
result = WSALookupServiceEnd(hLookup);

WSACleanup();

}

}

```



on kết nối Bluetooth

Trên ĐTDĐ hiện nay, các dịch vụ Bluetooth ngày càng phong phú đáp ứng đầy đủ nhu cầu kết nối không dây của người dùng như: truyền file, in ảnh, tai nghe... Những dịch vụ này đều có một định danh duy nhất (UUID) và được định nghĩa sẵn trong file header bthdef.h. Trong đó dịch vụ OBEX Object Push có vai trò tương tự như dịch vụ IrDA:OBEX của thiết bị hồng ngoại. Dưới đây là đoạn mã cho phép kết nối tới dịch vụ này:

```

#include <bthdef.h>

void main()

{

...

/* Khởi tạo Windows Socket */

if (WSAStartup(wVersionRequested, &m_data) == 0) {

/* Khởi tạo Bluetooth socket */

```

```

SOCKET s = socket(AF_BTH, SOCK_STREAM, BTHPROTO_RFCOMM);

SOCKADDR_BTH sin;

sin.addressFamily = AF_BTH;

/* Địa chỉ thiết bị đã tìm thấy ở trên */

sin.btAddr = bts->btAddr;

/* định danh dịch vụ OBEX Object Push */

sin.serviceClassId = OBEXObjectPushServiceClass_UUID;

sin.port = 0;

/* Kết nối tới dịch vụ OBEX Object Push */

int result = connect(s, (SOCKADDR*) &sin, sizeof(sin));

WSACleanup();

}

}

```

Như vậy bằng việc sử dụng các hàm send() và recv() cùng với giao thức OBEX, chúng ta có thể dễ dàng gửi và nhận file giữa PC và ĐTDĐ.



le trong thư mục điện thoại

## LỜI KẾT

Trên đây, tôi vừa giới thiệu với các bạn hai cách kết nối không dây phổ biến nhất hiện nay phục vụ nhu cầu trao đổi dữ liệu giữa PC và ĐTDĐ. Để thuận tiện cho việc phát triển ứng dụng trên nền .NET, tôi đã viết lớp thư viện Bluetooth cho phép dễ dàng kết nối tới thiết bị Bluetooth tương tự như lớp thư viện IrDA của Microsoft (có thể tải về tại website [www.hitekgroup.net](http://www.hitekgroup.net)). Chương trình demo do được viết riêng cho ĐTDĐ

Sony Ericsson T610 nên có thể làm việc không hiệu quả trên các dòng máy khác. Hy vọng qua bài viết này, các bạn có thể tự xây dựng cho mình một ứng dụng quản lý file phù hợp với chiếc điện thoại của mình.

Tài liệu và chương trình demo có thể tải tại website [www.hitekgroup.net](http://www.hitekgroup.net) hoặc <http://hitekgroup.cabspage.com/>.

Nguyễn Đức Thắng  
thangnd@hitekgroup.net

# Đồng bộ dữ liệu khi truy xuất tập tin với Java

Có những chương trình ứng dụng đòi hỏi việc ghi thông tin vào tập tin (file) để quản lý (log) hay chia sẻ dữ liệu giữa các tiến trình (process) hay tiểu trình (thread). Một vấn đề thường đặt ra đó là đồng bộ dữ liệu trong khi đọc hay ghi file. Giả sử hai tiến trình A và B cùng lúc ghi dữ liệu vào file X (hoặc một tiến trình ghi và một tiến trình đọc) thì khi đó dữ liệu được xử lý ra sao?

Chúng ta xem một ứng dụng thực tế sau: Ta dùng file counter.dat để đếm số người truy cập vào một trang web. Giả sử con số trong file này đã là 1000. Giả sử có 2 người cùng truy cập site vào một thời điểm, khi đó có 2 giao dịch xảy ra:

Giao dịch 1:

```
f = open("counter.dat");
```

```
hits = f.readNumber();
```

```
f.close();
```

Như vậy giao dịch 1 đọc "1000" vào biến hits. Tương tự, giao dịch 2 cũng đọc "1000" vào biến hits. Sau đó:

```
++hits;
```

```
Print("Hits on this page: " + hits);
```

Mỗi giao dịch tăng giá trị hits của mình và mỗi giao dịch có được con số 1001. Sau đó, hai giao dịch thực hiện việc cập nhật số lần truy cập trong file counter.dat lên 1001.

```
f = open("counter.dat");
```

```
f.write(hits);
```

```
f.close();
```

Vấn đề ở đây là trang web đã được truy cập 2 lần, trong khi giá trị đếm chỉ tăng lên 1. Đó là chưa kể mỗi người sử dụng đều thấy giá trị đếm là 1001 thay vì một người thấy 1001 và một người thấy 1002.

Ở đây ta dùng cơ chế Semaphore để giải quyết vấn đề. Một tiểu trình trước khi truy xuất file (đọc/ ghi) phải đưa ra yêu cầu (acquire) để được cấp phát tài nguyên. Sau khi xong việc phải trả lại (release) tài nguyên cho trình quản lý.

Class Semaphore sẽ hiện thực interface trong Code 1. Ta hình dung Semaphore như một cơ chế cấp phát tài nguyên mà ở đó số lượng tài nguyên có hạn. Khi một tiểu trình xin cấp phát tài nguyên, nó kiểm tra xem có còn tài nguyên hay không. Nếu còn thì tiến hành cấp phát, ngược lại báo tiểu trình đó chờ đến khi tiểu trình khác giải phóng tài nguyên thì tiến hành cấp phát. Ở đây có hai chế độ chờ: chờ cho đến khi được cấp phát (acquire) hay chờ trong khoảng thời gian bao lâu (attempt) – trường hợp này nhằm tránh tắt nghẽn. Xem Code 2.

Khi một tiểu trình muốn đọc file phải đưa ra yêu cầu đọc, còn khi muốn ghi file phải đưa ra yêu cầu ghi. Xem Code 3.

Nếu tiểu trình A đang đọc, tiểu trình B yêu cầu đọc thì cho phép. Ngược lại, nếu tiểu trình A đang đọc, tiểu trình B yêu cầu ghi thì tiểu trình B phải đợi cho đến khi không còn tiểu trình đọc nào đang được phục vụ.

Tương tự nếu tiểu trình A đang ghi thì không cho phép tiểu trình khác đọc hay ghi cho đến khi tiểu trình A ghi xong.

Class SemReadWrireLock trong Code 4 sẽ hiện thực interface trong Code 3.

Các lớp khi dùng chung file phải thiết lập interface như Code 5.

Code 6 là ví dụ ứng dụng với Class A và B đọc/ ghi chung file.

Hy vọng các đoạn code giới thiệu trong bài sẽ giúp ích cho các bạn trong việc xây dựng ứng dụng có dùng chung – chia sẻ tài nguyên hay đọc/ghi file.

```
horeapp;
```

s represents a Synchronization Object.

```
Sync {
```

```
rows InterruptedException;
```

```
.(long msec) throws InterruptedException;
```

```
horeapp;
```

Read/Write Lock class represents a Read/Write Lock Object.

```
ReadWriteLock {
```

```
);
```

```
horeapp;
```

Semaphore class represents a Semaphore Synchronization Resource Object.

```
Semaphore implements Sync {
```

```
permits;
```

```
number of available permits
```

```
Semaphore(long initialPermits) {
```

```
permits; }
```

```
void release() {
```

```
wait() throws InterruptedException {
```

```
if (isInterrupted()) throw new InterruptedException();
```

```
if (is) {
```

```
if (permits >= 0) wait();
```

```
throw new InterruptedException();
```

```
try { attempt(long secs) throws InterruptedException {
```

```
if (isInterrupted()) throw new InterruptedException();
```

```
if (is) {
```

```
wait() but messier
```

```
if (permits <= 0)
```

```
wait if not needed
```

```
long millis = System.currentTimeMillis();
```

```
long msecs;
```

```

-out
em.currentTimeMillis();
cs - (now - startTime);
))

```

```

dException ie) {

```

```

horeapp;

```

WriteLock class represents a Object Semaphore Read/Write Lock.

```

nReadWriteLock implements ReadWriteLock {
:cess to active slot
new Semaphore(1);
aring by readers
te implements Sync {
;
zed void acquire()
edException {
on lock until first passes
= 0) active_.acquire(); }
zed void release() {
0) active_.release(); }
zed boolean attempt(long msec)
edException {
= 0) {
tempt(msec);

```

```

new ReaderGate();
elock() { return active_; }
llock() { return rGate_; } }

```

```

horeapp;

```

er class represents a FileManager Object.

```

FileManager {
.ock oSemaphore = new SemReadWriteLock(); }

```

```

plements FileManager{

```

```

adLock().acquire();

```

```

adLock().release();

```

```
implements FileManager{
```

```
    fileLock().acquire();
```

```
    fileLock().release();
```

# Ô mật khẩu “tự bảo vệ”

Rất nhiều bạn đọc gửi hỏi về cách làm ô mật khẩu bảo vệ trong môi trường .NET, cách chặn message `EN_UPDATE` của `Textbox` để lưu mật khẩu sang một thuộc tính khác và biến thuộc tính `Text` thành những dấu \* cùng nhiều câu hỏi khác liên quan tới vấn đề bảo vệ ô mật khẩu.

Trước hết, xin thông báo với các bạn một tin mừng là trong môi trường .NET việc “subclass” các form, control đã trở nên cực kỳ đơn giản. Thay vì sử dụng các hàm API theo một quy trình lằng nhằng hoặc phải cầu viện tới những công cụ như *MsgHook OCX* (mà không phải lúc nào cũng có thể kiếm được đồ miễn phí), lập trình viên chỉ cần “override” thủ tục `WndProc`. Thủ tục này nhận đầu vào là một đối tượng kiểu *System.Windows.Forms.Message* với các thành phần mô tả message: *hWnd* (handle của cửa sổ nhận message), *Msg* (số hiệu message), *WParam* và *LParam* (các dữ liệu bổ sung đi kèm với message). Việc tạo một kiểu `TextBox` đặc biệt trở nên đơn giản hơn bao giờ hết:

```
Public Class SecureTextBox

Inherits TextBox

Private Const WM_GETTEXT As Integer = &HD

Protected Overrides Sub WndProc(ByRef m As System.Windows.Forms.Message)

    Select Case m.Msg

        Case WM_GETTEXT

            ' Do nothing here to disable the default

        Case Else

            'pass unhandled messages back to the default message handler

            MyBase.WndProc(m)

    End Select

End Sub

End Class
```

Sau đó, chúng ta cần thay thế các tham chiếu tới `TextBox` chuẩn thành tham chiếu tới lớp *SecureTextBox* vừa tạo. Ví dụ:

```
' Friend WithEvents TextBox1 As TextBox

' chuyển thành

Friend WithEvents TextBox1 As SecureTextBox

' Me.TextBox1 = New System.Windows.Forms.TextBox()
```



` chuyển thành

```
Me.TextBox1 = New SecureTextBox()
```

Nếu muốn dùng một control kiểu mới để chặn tận gốc mọi mưu đồ đọc trộm thuộc tính Text, bạn có thể vào địa chỉ <http://www.codeproject.com/cs/miscctrl/passwordbox.asp> xem cách làm và tải chương trình nguồn về tham khảo.

Với một số bạn sử dụng VB6, sự nhầm lẫn đáng tiếc của tôi khi đề cập tới message EN\_UPDATE ở phần cuối bài viết trước đã gây cho họ không ít bối rối. Thực ra, việc chặn message đó thích hợp với VC++ hơn là với VB6. Với VB6, chúng ta có thể xử lý mọi chuyện trong thủ tục ứng với sự kiện Change của Textbox. Để sửa sai, tôi xin gửi tặng các bạn chương trình nguồn của SECURPWD.OCX do tôi phát triển. Chương trình của tôi chưa thật hoàn thiện (tôi chưa nghĩ được cách nào giải quyết dứt điểm vấn đề) vì nó dựa trên giả định mật khẩu do người dùng nhập vào không chứa ký tự nào giống ký tự được chọn làm PasswordChar. Để tránh khả năng đó và tạo hình tròn màu đen trong ô mật khẩu giống Windows XP, tôi sử dụng ký tự số 149 làm PasswordChar, đặt phông chữ Times New Roman cỡ 11 cho Textbox trong chương trình và không để lộ các thuộc tính Font, PasswordChar của SecurPwd. Sau khi hoàn thành chương trình tôi mới biết rằng Microsoft cũng sử dụng ký tự số 149 (nhưng là của phông Tahoma, phông chữ mặc định trong Windows XP) để làm ký tự thay thế trong ô mật khẩu của thư viện Comctl32.dll phiên bản 6. Vì ký tự đó trong các phông khác có thể có hình dạng khác nên nếu người lập trình đặt lại phông chữ thì ký tự thay thế có thể hiển thị không đúng ý định của Microsoft. Do ký tự 149 của phông Times New Roman hiển thị dấu tròn hơi nhỏ nên tôi buộc phải ép cỡ chữ 11 để dấu tròn khỏi biến thành hình vuông.

Đoạn mã cho sự kiện Change không có gì phức tạp nhưng tôi buộc phải tách làm hai trường hợp vì hàm Mid của VB6 không chấp nhận giá trị âm cho tham số độ dài chuỗi cần lấy.

```
Private Sub Text1_Change()  
  
    Dim text_len, RealText_len, cnt, diff_start, diff_end, SelStart_pos  
  
    text_len = Len(Text1.Text)  
  
    RealText_len = Len(m_RealText)  
  
    SelStart_pos = Text1.SelStart  
  
    Sel_len = Text1.SelLength  
  
    diff_start = 0  
  
    For cnt = 1 To text_len  
  
        If Mid$(Text1.Text, cnt, 1) <> Text1.PasswordChar Then diff_start = cnt: Exit For  
  
    Next  
  
    If diff_start = 0 Then  
  
        If text_len <> RealText_len Then  
  
            m_RealText = Mid$(m_RealText, 1, SelStart_pos) & Mid$(m_RealText,  
Abs(RealText_len - text_len) + SelStart_pos + 1)  
  
            Text1.Text = String(text_len, Text1.PasswordChar)  
  
            Text1.SelStart = SelStart_pos
```

```

End If

Exit Sub

End If

For cnt = diff_start + 1 To text_len

    If Mid$(Text1.Text, cnt, 1) = Text1.PasswordChar Then diff_end = cnt: Exit For

Next

If diff_end = 0 Then diff_end = text_len + 1

m_RealText = Mid$(m_RealText, 1, diff_start - 1) & Mid$(Text1.Text, diff_start,
diff_end - diff_start) & Mid$(m_RealText, diff_end - (text_len - RealText_len))

Text1.Text = String(text_len, Text1.PasswordChar)

Text1.SelStart = SelStart_pos

End Sub

```

Tôi đặt thuộc tính RealText thành chỉ đọc trong lúc chạy và không tồn tại khi thiết kế vì hai lý do: rất ít khi lập trình viên có nhu cầu đặt giá trị mật khẩu thay cho người dùng cuối, việc cho phép lập trình viên đặt lại RealText cho ô mật khẩu sẽ dẫn đến việc phải đặt lại thuộc tính Text (và làm đảo lộn hết mọi logic trong sự kiện Change của TextBox).

```

Public Property Let RealText(ByVal New_RealText As String)

    If Ambient.UserMode = False Then Err.Raise 387

    If Ambient.UserMode Then Err.Raise 382

    m_RealText = New_RealText

    PropertyChanged "RealText"

End Property

```

Vấn đề cuối cùng của SecurPwd là việc đặt thuộc tính PasswordChar cho Text1. Việc này dường như không thật sự cần thiết vì có thể dùng một biến chứa ký tự 149 để dùng trong thủ tục Text1\_Change. Tuy nhiên, nếu không đặt thuộc tính PasswordChar cho Text1 thì người dùng cuối có thể chép nội dung của ô mật khẩu. Bản thân việc chép nội dung của ô mật khẩu không làm lộ mật khẩu vì chúng ta đã biến tất cả các ký tự trong thuộc tính Text thành các dấu tròn. Nhưng chuyện gì sẽ xảy ra nếu người dùng cuối lại dán những ký tự đó vào ô mật khẩu? Các bạn hãy nhớ lại hạn chế căn bản của SecurPwd là không xử lý được tình huống người dùng nhập ký tự trùng với ký tự thay thế!

**Đỗ Thanh Xuân**

Trung tâm CNTT – Ngân hàng Công Thương Việt Nam

# Hỗ trợ XP Themes trên Visual C++ 2005: Tưởng khó mà dễ!

Bài viết này muốn đề cập đến các dự án dạng Win32 Application (dùng thuần Win32 API). Không như các ứng dụng MFC tự động hỗ trợ XP Theme ngay khi tạo dự án, các ứng dụng Win32 này vẫn mang dáng dấp các thành phần giao diện từ thời "xa xưa". Chắc chắn là các bạn muốn "thổi hồn thời đại" vào ứng dụng của mình, bắt nó hỗ trợ giao diện XP. Tôi cũng vậy và tôi đã có một buổi toát mồ hôi đánh vật với Visual C++ 2005!

Đối với các phiên bản trước, cách đơn giản nhất là copy file *MyApp.exe.manifest* vào thư mục chứa file *MyApp.exe*. File này chắc các bạn đã biết, nó thông báo cho Windows XP sử dụng file *Comctl32.dll* phiên bản 6.0 để ứng dụng có thể dùng XP theme, nếu không Windows chỉ gọi phiên bản 5.82, khi đó ứng dụng chỉ có giao diện kiểu cũ.

```
<?xml encoding="UTF-8" standalone="yes"?>
<asm:assemblyIdentity xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <asm:assemblyIdentity version="1.0.0.0" processorArchitecture="X86"
    name="MyName.ProductName.YourApplication" type="win32">
  </asm:assemblyIdentity>
  <asm:assemblyDescription>Your application description here.</asm:assemblyDescription>
</asm:assemblyIdentity>
<asm:assemblyIdentity type="win32"
  xmlns="urn:schemas-microsoft-com:asm.v1" name="Microsoft.Windows.Common-Controls" version="6.0.0.0"
  processorArchitecture="X86" publicKeyToken="6595b64144ccf1df" language="*">
</asm:assemblyIdentity>
</asm:assembly>
```

Tôi cũng làm tương tự nhưng những gì tôi nhận được là một thông báo lỗi khó hiểu.

Tôi đành sử dụng cách "chính qui" hơn là đưa thẳng file *.manifest* vào trong tài nguyên của dự án – cách đã được dùng hiệu quả trên Visual C++ 7.1 trở về trước. Cách này chắc nhiều bạn biết, nhưng tôi vẫn muốn nêu lại:

- Trong thẻ *Resource view*, dùng *Add Resource* thêm file *\*.manifest* vào
- Mở file *\*.rc* bằng cách chọn *View code*, sửa đổi dạng tài nguyên cho file *\*.manifest* là *RT\_MANIFEST* và tên tài nguyên là *IDR\_MANIFEST*
- Mở file *Resource.h*, thêm 1 dòng *#define IDR\_MANIFEST 1*
- Dịch lại ứng dụng

Cách này cho kết quả rất tốt trên các phiên bản trước, nhưng lần này tôi lại nhận được thông báo lỗi khi dịch:

tài nguyên MANIFEST đã bị lặp lại 2 lần.

Đến khi dùng *Resource Hacker* (hay là dùng chính VS IDE) để mở file \*.exe (nguyên gốc, chưa có chỉnh sửa gì) do VC++ tạo ra, tôi mới hiểu được nguyên nhân: trong file \*.exe này đã có sẵn tài nguyên manifest, đó là lí do trình dịch báo lỗi trùng.

Nội dung tài nguyên này như sau:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <assemblyIdentity
    name="Microsoft.VC80.DebugCRT"
    version="6.0.60608.0" processorArchitecture="x86"
    publicKeyToken="1fc8b3b9a1e18e3b">
  </assemblyIdentity>
</assembly>
```

Đến đây, ta hiểu được nguyên nhân của thông báo lỗi khó hiểu ở trên. Nó chỉ cho ứng dụng liên kết đến thư viện CRT của VC++ (file **MSVCR80D.DLL** - vì đây là phiên bản debug), nhưng khi đó file **MyApp.exe.manifest** đã "nhảy ra" chiếm quyền, nên ứng dụng không thể liên kết được đến thư viện này. Bản thân nội dung tài nguyên này lại không chứa phần chỉ dẫn để sử dụng file **Comctl32.dll** ver 6.0 – điều kiện để ứng dụng hỗ trợ XP theme. Một cách đơn giản là bạn bổ sung một đoạn XML có nội dung sau vào phần tài nguyên ở trên (vào trước tag `</assembly>`), rồi ghi lại.

```
<assemblyIdentity type="win32"
  name="Microsoft.Windows.Common-Controls" version="6.0.0.0"
  processorArchitecture="X86" publicKeyToken="6595b64144ccf1df" language="*">
</assemblyIdentity>
</assembly>
```

Nhưng như thế thì thủ công quá! Tôi tình cờ phát hiện ra một cách đơn giản đến không ngờ: Bạn chỉ cần thêm file **\*.manifest** (tên file không quan trọng, có nội dung như file **MyApp.exe.manifest** ở trên) ở thẻ *Solution Explorer* (không phải là *Resource View*), nhấn chuột phải vào dự án, chọn *Add/ Existing Item*. Cho dịch lại, thế là xong. Khi dịch, bạn sẽ thấy có thông báo:

```
resources...  
manifest to resources...
```

Sau khi biên dịch xong, ứng dụng của bạn sẽ hỗ trợ XP theme.

Nếu bây giờ bạn dùng Resource Hacker để mở lại file \*.exe thì bạn sẽ thấy tài nguyên 24/1/1033 phía trên đã bao hàm cả phần chỉ định liên kết với thư viện CRT và **Comctl32.dll 6.0**.

Nếu thích, bạn có thể sửa lại file MyApp.exe.manifest để nó chứa cả phần chỉ định đến thư viện **MSVCR80.DLL** và đặt cùng với **MyApp.exe** chứ không nhất thiết phải đưa nó vào ứng dụng.

Hi vọng bài viết nhỏ này sẽ giúp ích cho các bạn. Rất mong được các bạn trao đổi ý kiến.

Mai Văn Quân

Email: [Vimvq1987@gmail.com](mailto:Vimvq1987@gmail.com)



*Theo PC World VN*

## Code Snippet trong VS 2005

31/5/2006 10h:56

Một trong những tính năng nổi bật nhất trong bộ VS2005 là việc sử dụng Code Snippet. Đây là tiện ích giúp phát triển ứng dụng nhanh hơn nhờ việc tự động chèn các đoạn mã chương trình thường được sử dụng vào trong môi trường soạn thảo, nhờ đó lập trình viên sẽ không phải gõ lại những đoạn mã có sẵn đó.

Một lợi ích khác của Code Snippet là giảm thiểu lỗi. Code Snippet thường là những đoạn mã đơn giản, nhưng đôi khi việc gõ lại những đoạn mã này cũng dễ bị sai sót. Một ví dụ điển hình là cấu trúc Select case trong VB.NET, chúng ta rất dễ thiếu đoạn mã "Case Else" khi lập trình, nhưng nhờ Code Snippet chúng ta sẽ không bao giờ bị thiếu đoạn mã này.

VS 2005 không giới hạn số lượng Code Snippet được tạo. Bạn có thể tạo mới hoặc sử dụng, thay đổi (như mở rộng chức năng) Code Snippet sẵn có.

### Tạo mới Code Snippet

Code Snippets được tạo ra dưới định dạng file XML. Để xem cấu trúc file XML của Code Snippet, bạn có thể mở một file sẵn có hoặc tìm hiểu ở trang web <http://msdn2.microsoft.com/en-us/library/ms171418.aspx>. Bạn có thể tạo Code Snippet bằng cách tạo một file .snippet có cấu trúc theo định dạng XML như hướng dẫn ở website trên. Tuy nhiên cách làm này rất mất công và dễ gây chán với người lập trình.

Cách tạo Code Snippet đơn giản nhất là dùng Visual Basic Snippet Editor. Đây là tiện ích tạo Code Snippet do Microsoft phát triển. Bạn có thể tải về Visual Basic Snippet Editor theo địa chỉ <http://go.microsoft.com/fwlink/?LinkId=46593>.

Giả sử muốn xây dựng Code Snippet để đọc một giá trị trong Session khi biết tên bảng dữ liệu, trường dữ liệu trong hàng dữ liệu cần đọc, bạn tạo Code Snippet như hình sau:

Trong đó, phần giữa hai dấu \$ là tham số để thay đổi giá trị, tên cho Code Snippet. Để thêm một tham số cho Code Snippet, chọn Tab Replacements, nhấn đúp chuột lên tên tham số và nhấn vào biểu tượng (+).

### Sử dụng Code Snippet

Để sử dụng được Code Snippet bạn phải chép file .snippet vào thư mục *C:\Documents and Settings\Administrator\My Documents\Visual Studio 2005\Code Snippets\Visual Basic\My Code Snippets* của VS2005. Ngoài ra, bạn cũng có thể chép chúng vào các thư mục Code Snippet sẵn có của VS2005.

Nếu bạn dùng Visual Basic Snippet Editor thì công cụ này sẽ tự động chép file Code Snippet vào thư mục Code Snippet.

Để chèn Code Snippet, tại cửa sổ soạn thảo VS 2005, nhấn tổ hợp phím Ctrl + K + X, một menu Code Snippet được hiện ra. Chọn loại Code Snippet, ấn Enter, sau đó chọn tên Code Snippet và đoạn mã Code Snippet sẽ được chèn vào vị trí con trỏ.

### Kết luận

Nhờ Code Snippet thời gian lập trình giảm đi đáng kể. Việc mấu chốt nhất là chúng ta phải xem xét những đoạn mã nào hay được sử dụng để tạo Code Snippet phục vụ cho sử dụng sau này. Tất nhiên, cũng có thể tạo Code Snippet chỉ sử dụng một lần trong ứng dụng, nhưng việc làm này tốn thời gian vô ích.

Hiện có hai tổ chức chuyên nghiên cứu và tạo Code Snippet được giới thiệu ở website: <http://DotNetJunkies.com> và <http://GotCodeSnippets.com>.

**Dương Bá Hồng Minh**

Email: [minhdbh@yahoo.com](mailto:minhdbh@yahoo.com)

Tham khảo: <http://msdn.microsoft.com/msdnmag/issues/06/04/AdvancedBasics/>  
<http://msdn2.microsoft.com/en-us/library/ms171418>

*Theo PC World VN*

# Thiết kế MVC và Java Web Mail

20/6/2005 9h:47

Phương pháp thiết kế MVC bắt nguồn từ việc phát triển giao diện người dùng trong ngôn ngữ lập trình Smalltalk, đây là một trong những phương pháp thiết kế thành công nhất trong các phương pháp thiết kế hướng đối tượng. Hiện nay, MVC được dùng rộng rãi trong nhiều hệ thống phần mềm hướng đối tượng, bất kể được viết bằng ngôn ngữ hướng đối tượng nào.

Bài viết này giới thiệu tổng quan về phương pháp thiết kế MVC, và minh họa cách sử dụng MVC trong thiết

kế hướng đối tượng bằng việc xây dựng chương trình Java Web Mail. Bạn đọc phải quen thuộc với ngôn ngữ lập trình Java, các khái niệm về JSP, Servlet, Java Mail API.

### **Thiết kế MVC và Java**

MVC là viết tắt của Model-View-Controller. Phương pháp thiết kế MVC (MVC Design Pattern)[1] là phương pháp chia nhỏ một ứng dụng nhiều lớp hoặc chia nhỏ phần giao diện người dùng (user interface) của một ứng dụng thành ba thành phần chính là Model, View và Controller (hình 1).

- Model (tạm dịch là phần “Mô hình” [2]): Là một đối tượng hoặc tập hợp các đối tượng biểu diễn cho phần dữ liệu của chương trình, ví dụ các dữ liệu được lưu trong cơ sở dữ liệu (CSDL) hay từ các hệ thống ứng dụng khác (như mail...).
- View (tạm dịch là phần “Hiển thị”): Là phần giao diện với người dùng, bao gồm việc hiện dữ liệu ra màn hình, cung cấp các menu, nút bấm, hộp đối thoại, chọn lựa..., để người dùng có thể thêm, xóa, sửa, tìm kiếm và làm các thao tác khác đối với dữ liệu trong hệ thống.
- Controller (tạm dịch là phần “Điều khiển”): Là phần điều khiển toàn bộ logic về hoạt động của giao diện, tương tác với thao tác của người dùng (từ chuột, bàn phím và các thiết bị ngoại vi khác) và cập nhật, thao tác trên dữ liệu theo đầu vào nhận được và điều khiển việc chọn phần “Hiển thị” thích hợp để truyền dữ liệu tới người dùng.

Với phương pháp thiết kế này, các chức năng hiển thị, chức năng logic điều khiển và chức năng truy cập dữ liệu của chương trình được chia làm các phần tách biệt.

Java là một ngôn ngữ lập trình hướng đối tượng thuần túy nên việc áp dụng MVC vào các phần mềm viết bằng Java rất dễ dàng và hiển nhiên. Có hai hình mẫu chính của phương pháp thiết kế MVC trong Java là MVC model 1 (hình 2) và MVC model 2 (hình 3).

Trong MVC model 1, các trang JSP đóng vai trò “Hiển thị” (View) và “Điều khiển” (Controller). Có thể có nhiều trang JSP khác nhau đóng các vai trò khác nhau.

Thao tác của người dùng trên trình duyệt web được gửi tới một trang JSP.

Trang JSP này sẽ khởi tạo một hoặc nhiều Java Bean (nếu cần thiết), truyền các lệnh cần thi hành tới Java Bean (không phải Enterprise Java Bean).

Sau khi Java Bean thực hiện xong việc truy xuất hoặc cập nhật dữ liệu, trang JSP ban đầu có thể hiển thị dữ liệu lấy từ Bean (JSP ban đầu đóng luôn vai trò View), hoặc chọn một trang JSP khác để hiện dữ liệu từ Bean (JSP ban đầu đóng luôn vai trò Controller). Trong một thiết kế tốt, để bảo đảm việc tách rời phần trình bày và logic của chương trình, trang JSP nhận yêu cầu chỉ đóng vai trò “Điều khiển” (Controller).

MVC model 1 có một nhược điểm là phần logic điều khiển được viết trong trang JSP, như vậy phần chương trình Java phức tạp dùng để điều khiển sẽ bị lẫn vào trong mã HTML dùng để trình bày. Độ phức tạp của chương trình càng cao, thì trang JSP càng khó phát triển và bảo trì. Hơn nữa, trong các dự án phức tạp, phần hiển thị do người thiết kế web giỏi về HTML và đồ họa thực hiện, còn phần điều khiển được người chuyên về lập trình thực hiện. Dùng JSP làm phần điều khiển sẽ khó phân ranh giới trách nhiệm giữa nhóm thiết kế đồ họa và nhóm lập trình. Để khắc phục nhược điểm này, MVC model 2 ra đời.

Trong MVC model 2, một hoặc nhiều servlet (thường là một) đóng vai trò điều khiển, các Java Bean đóng vai trò mô hình và các trang JSP đóng vai trò hiển thị.

Trong model 2, các logic phức tạp của chương trình được viết hoàn toàn trong các servlet (chương trình Java). Phần hiển thị chỉ gồm các trang JSP với một vài mã đơn giản để lấy dữ liệu có sẵn, không có logic phức tạp, vì thế hoàn toàn có thể giao cho người thiết kế web.

Các yêu cầu của người dùng được gửi từ trình duyệt web tới servlet. Servlet sẽ khởi tạo Java Bean (nếu cần thiết), ra lệnh thu thập, cập nhật thông tin. Khi Java Bean hoàn thành công việc, servlet sẽ chọn trang JSP thích hợp để hiện thông tin trong Java Bean cho người dùng.

Đây là một cách sử dụng MVC rất hiệu quả trong Java. Tất nhiên, sử dụng MVC model 2 một cách hoàn toàn cứng nhắc, phần “Điều khiển” chỉ dùng servlet, phần “Hiển thị” chỉ dùng JSP sẽ dẫn đến một vài trường hợp kém hiệu quả, nhất là khi các yêu cầu từ trình duyệt web chỉ đòi hỏi việc hiển thị thông tin. Trong trường hợp này, gửi thẳng yêu cầu hiển thị từ trình duyệt web tới trang JSP sẽ hiệu quả hơn (hình 4).

Trong cách áp dụng MVC này, các yêu cầu có liên quan đến logic chương trình hoặc truy cập dữ liệu sẽ được gửi tới servlet controller, còn các yêu cầu chỉ liên quan tới hiển thị sẽ được gửi tới JSP controller.

### **Chương trình Java Web Mail**

Chương trình Java Web Mail được thiết kế theo MVC model 2. Yêu cầu của chương trình chỉ là hiện màn hình đăng nhập username và password, không có logic, nên nó có thể được gọi thông qua servlet controller hoặc JSP controller. Sau đó, tùy theo các yêu cầu mà HTTP request sẽ được chuyển đến servlet controller hoặc JSP controller cho phù hợp.

Chương trình gồm các chức năng sau:

- Sử dụng HTTP để đọc và gửi mail từ bất kỳ một mail server nào dùng POP3 protocol trên Internet hoặc trong intranet.
- Có thể truy cập mail qua bất kỳ proxy server nào có chức năng HTTP proxy. Chức năng này rất có ích khi người dùng kết nối vào Internet từ một mạng intranet phía sau một tường lửa (firewall), và firewall này ngăn chặn các máy tính trong intranet truy cập POP server bên ngoài, trong khi đó, người dùng muốn gửi và nhận mail từ một POP server trên Internet.
- Sử dụng khả năng xử lý các loại dữ liệu của trình duyệt web để hiện attachment.

Chương trình gồm các thành phần sau:

Một servlet controller MailUtilServlet, dùng để nhận các yêu cầu: log in, log out, gửi mail, hiện tập tin đính kèm (attachment). Những yêu cầu này sẽ được xử lý về mặt logic rồi gửi tới Java Bean để thực sự làm công việc truy cập mail server. Sau khi Java Bean thực hiện việc truy cập dữ liệu xong, MailUtilServlet sẽ chọn trang JSP thích hợp để hiển thị dữ liệu.

Một Java Bean MailUserBean dùng để truy cập mail server, lấy danh sách và nội dung mail trong mail box, xóa mail trong server.

Trang JSP index.jsp đóng vai trò JSP controller, dùng để nhận các yêu cầu: hiện danh sách các mail trong hộp thư, hiện nội dung của một mail được chọn trong danh sách, hiện trang soạn thảo mail để người dùng soạn thảo và gửi mail. Những thông tin cần để hiển thị đã có sẵn trong MailUserBean, vì MailUserBean đã lấy những thông tin này khi nhận được yêu cầu log in từ MailUtilServlet. Vì thế, những loại yêu cầu này thuộc về loại yêu cầu hiển thị, không có logic phức tạp, nên không cần phải gửi qua MailUtilServlet.

Tập hợp các trang JSP:

- \* menu.jsp dùng để hiện menu lệnh bao gồm Log in, Inbox, Compose và Exit.
- \* first.jsp là trang để nhập username, password, mailserver cho việc login.
- \* messageheaders.jsp là trang hiện danh sách mail có trong mail box để người dùng chọn xem và xóa mail.
- \* messagecontent.jsp là trang để hiện nội dung của mail đã chọn từ danh sách.
- \* compose.jsp là trang để soạn thảo mail cần gửi.
- \* status.jsp là trang dùng để báo về lỗi khi log in, log out không thành công, và thông báo về kết quả gửi mail thành công hay không.
- \* errordetails.jsp là trang dùng để cung cấp thông tin chi tiết mỗi khi có lỗi log in, log out, gửi mail không thành công. Thông tin trong trang này bao gồm cả Stack Trace của exception khi sinh ra lỗi, chủ yếu dành cho lập trình viên dùng để xem chi tiết về vấn đề đã xảy ra.
- \* logout.jsp là trang hiện ra khi người dùng log in ra khỏi hệ thống mail.

Một vài trang JSP và text file khác dùng để trang trí.



\* Một CSS (Cascade Style Sheet) tên là styleSheet.txt, dùng để định dạng về font và màu sắc cho tất cả các file JSP.

Trong hệ thống này, không có database sever. MailUserBean lấy và cập nhật dữ liệu từ POP mail server, gửi mail từ SMTP server, sử dụng Java Mail API.

Đây là một ứng dụng web sử dụng JSP/Servlet nên phải được cài đặt trên một web server có hỗ trợ servlet engine (ví dụ BEA WebLogic, IBM Web Sphere, Sun One, JBoss, Tomcat, Alaire JRun...) hoặc một web server kết nối với servlet engine (ví dụ như IIS + Tomcat, Apache + Tomcat, IIS + JRun, Apache + Jrun...).

Bạn đọc có thể **tải về toàn bộ mã nguồn chương trình** hoặc:

<http://www.quantrimang.com/download/ebooks/javawebmailsource.zip>

**Châu Hồng Linh**

[chauhonglinh@hanoian.com](mailto:chauhonglinh@hanoian.com)

Theo PCWorld Việt Nam

# Phân tích phát hiện tấn công DDos và giải pháp

9/5/2005 13h:58

## **CERT<sup>®</sup> Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks**

Original issue date: September 19, 1996

Last revised: November 29, 2000

Updated vendor information for the Linux kernel.

A complete revision history is at the end of this file. **This advisory supersedes the IP spoofing portion of [CA-95.01](#).**

Two "underground magazines" have recently published code to conduct denial-of-service attacks by creating TCP "half-open" connections. This code is actively being used to attack sites connected to the Internet. There is, as yet, no complete solution for this problem, but there are steps that can be taken to lessen its impact. Although discovering the origin of the attack is difficult, it is possible to do; we have received reports of attack origins being identified.

Any system connected to the Internet and providing TCP-based network services (such as a Web server, FTP server, or mail server) is potentially subject to this attack. Note that in addition to attacks launched at specific hosts, these attacks could also be launched against your routers or other network server systems if these hosts enable (or turn on) other TCP services (e.g., echo). The consequences of the attack may vary depending on the system; however, the attack itself is fundamental to the TCP protocol used by all systems.

If you are an Internet service provider, please pay particular attention to Section III and Appendix A, which describes step we urge you to take to lessen the effects of these attacks. If you are the customer of an Internet service provider, please encourage your provider to take these steps.

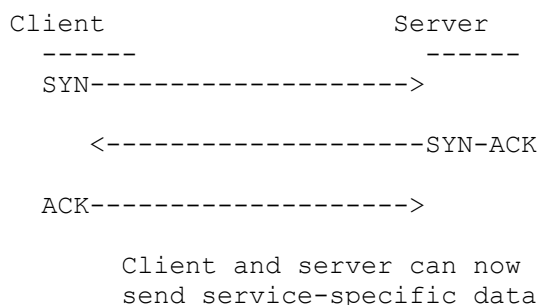
This advisory provides a brief outline of the problem and a partial solution. We will update this advisory as we receive new information. If the change in information warrants, we may post an updated advisory on [comp.security.announce](mailto:comp.security.announce) and redistribute an update to our cert-advisory mailing list. As always, the latest information is available at the URLs listed at the end of this advisory.

---

## I. I. DESCRIPTION

When a system (called the client) attempts to establish a TCP connection to a system providing a service (the server), the client and server exchange a set sequence of messages. This connection technique applies to all TCP connections--telnet, Web, email, etc.

The client system begins by sending a SYN message to the server. The server then acknowledges the SYN message by sending SYN-ACK message to the client. The client then finishes establishing the connection by responding with an ACK message. The connection between the client and the server is then open, and the service-specific data can be exchanged between the client and the server. Here is a view of this message flow:



The potential for abuse arises at the point where the server system has sent an acknowledgment (SYN-ACK) back to client but has not yet received the ACK message. This is what we mean by half-open connection. The server has built in its system memory a data structure describing all pending connections. This data structure is of finite size, and it can be made to overflow by intentionally creating too many partially-open connections.

Creating half-open connections is easily accomplished with IP spoofing. The attacking system sends SYN messages to the victim server system; these appear to be legitimate but in fact reference a client system that is unable to respond to the SYN-ACK messages. This means that the final ACK message will never be sent to the victim server system.

The half-open connections data structure on the victim server system will eventually fill; then the system will be unable to accept any new incoming connections until the table is emptied out. Normally there is a timeout associated with a pending connection, so the half-open connections will eventually expire and the victim server system will recover. However, the attacking system can simply continue sending IP-spoofed packets requesting new connections faster than the victim system can expire the pending connections.

In most cases, the victim of such an attack will have difficulty in accepting any new incoming network connection. In these cases, the attack does not affect existing incoming connections nor the ability to originate outgoing network connections.

However, in some cases, the system may exhaust memory, crash, or be rendered otherwise inoperative.

The location of the attacking system is obscured because the source addresses in the SYN packets are often implausible. When the packet arrives at the victim server system, there is no way to determine its true source. Since the network forwards packets based on destination address, the only way to validate the source of a packet is to use input source filtering (see Appendix A).

## **II. II. IMPACT**

Systems providing TCP-based services to the Internet community may be unable to provide those services while under attack and for some time after the attack ceases. The service itself is not harmed by the attack; usually only the ability to provide the service is impaired. In some cases, the system may exhaust memory, crash, or be rendered otherwise inoperative.

## **III. III. SOLUTION**

There is, as yet, no generally accepted solution to this problem with the current IP protocol technology. However, proper router configuration can reduce the likelihood that your site will be the source of one of these attacks.

Appendix A contains details about how to filter packets to reduce the number of IP-spoofed packets entering and exiting your network. It also contains a list of vendors that have reported support for this type of filtering.

NOTE to Internet Service Providers:

We STRONGLY urge you to install these filters in your routers to protect your customers against this type of an attack. Although these filters do not directly protect your customers from attack, the filters do prevent attacks from originating at the sites of any of your customers. We are aware of the ramifications of these filters on some current Mobile IP schemes and are seeking a position statement from the appropriate organizations.

NOTE to customers of Internet service providers:

We STRONGLY recommend that you contact your service provider to verify that the necessary filters are in place to protect your network.

Many networking experts are working together to devise improvements to existing IP implementations to "harden" kernels to this type of attack. When these improvements become available, we suggest that you install them on all your systems as soon as possible. This advisory will be updated to reflect changes made by the vendor

## **IV. IV. DETECTING AN ATTACK**

Users of the attacked server system may notice nothing unusual since the IP-spoofed connection requests may not load the system noticeably. The system is still able to establish outgoing connections. The problem will most likely be noticed by client systems attempting to access one of the services on the victim system.

To verify that this attack is occurring, check the state of the server system's network traffic. For example, on SunOS this may be done by the command:

```
netstat -a -f inet
```

Note that use of the above command depends on the OS version, for example for a FreeBSD system use

```
netstat -s |grep "listenqueue overflows"
```

Too many connections in the state "SYN\_RECEIVED" could indicate that the system is being attacked.

---

## **V. APPENDIX A - REDUCING IP SPOOFED PACKETS**

### **1. Filtering Information**

With the current IP protocol technology, it is impossible to eliminate IP-spoofed packets. However, you can take steps to reduce the number of IP-spoofed packets entering and exiting your network.

Currently, the best method is to install a filtering router that restricts the input to your external interface (known as an input filter) by not allowing a packet through if it has a source address from your internal network. In addition, you should filter outgoing packets that have a source address different from your internal network to prevent a source IP spoofing attack from originating from your site.

The combination of these two filters would prevent outside attackers from sending you packets pretending to be from your internal network. It would also prevent packets originating within your network from pretending to be from outside your network. These filters will *\*not\** stop all TCP SYN attacks, since outside attackers can spoof packets from *\*any\** outside network, and internal attackers can still send attacks spoofing internal addresses.

We STRONGLY urge Internet service providers to install these filters in your routers.

In addition, we STRONGLY recommend customers of Internet service providers to contact your service provider to verify that the necessary filters are in place to protect your network.

### **2. Vendor Information**

The following vendor(s) have reported support for the type of filtering we recommend and provided pointers to additional information that describes how to configure your router. If we hear from other vendors, we will add their information to the "Updates" section at the end of this advisory.

If you need more information about your router or about firewalls, please contact your vendor directly.

#### **Cisco**

Refer to the section entitled "ISP Security Advisory" on <http://www.cisco.com> for an up-to-date explanation of how to address TCP SYN flooding on a Cisco router.

NOTE to vendors:

If you are a router vendor who has information on router capabilities and configuration examples and you are not represented in this list, please contact the CERT Coordination Center at the addresses given in the Contact Information section below. We will update the advisory after we hear from you.

### 3. Alternative for routers that do not support filtering on the inbound side

If your vendor's router does not support filtering on the inbound side of the interface or if there will be a delay in incorporating the feature into your system, you may filter the spoofed IP packets by using a second router between your external interface and your outside connection. Configure this router to block, on the outgoing interface connected to your original router, all packets that have a source address in your internal network. For this purpose, you can use a filtering router or a UNIX system with two interfaces that supports packet filtering.

Note: Disabling source routing at the router does not protect you from this attack, but it is still good security practice to follow.

On the input to your external interface, that is coming from the Internet to your network, you should block packets with the following addresses:

- Broadcast Networks: The addresses to block here are network 0 (the all zeros broadcast address) and network 255.255.255.255 (the all ones broadcast network).
- Your local network(s): These are your network addresses
- Reserved private network numbers: The following networks are defined as reserved private networks, and no traffic should ever be received from or transmitted to these networks through a router:

10.0.0.0	-	10.255.255.255	10/8	(reserved)
127.0.0.0	-	127.255.255.255	127/8	(loopback)
172.16.0.0	-	172.31.255.255	172.16/12	(reserved)
192.168.0.0	-	192.168.255.255	192.168/16	(reserved)

---

The CERT Coordination Center staff thanks the team members of NASIRC for contributing much of the text for this advisory and thanks the many experts who are devoting time to addressing the problem and who provided input to this advisory.

---

## VI. UPDATES

### 3COM

Please refer to the "Network Security Advisory" for a thorough discussion of how to address TCP SYN flooding attacks on a 3Com router:

<http://www.3com.com/>

### Berkeley Software Design, Inc.

BSDI has patches available.

### PATCH

K210-021 (<ftp://ftp.bsdi.com/bsdi/patches/patches-2.1/K210-021>)

md5 checksum: c386e72f41d0e409d91b493631e364dd K210-021

This patch adds two networking features that can help defeat and detect some types of denial of service attacks.

This patch requires U210-025 which provides new copies of *sysctl(8)* and *netstat(1)* for configuration and monitoring of these new features.

## **PATCH**

K210-022 (<ftp://ftp.bsdi.com/bsdi/patches/patches-2.1/K210-22>)

md5 checksum: 9ec62b5e9cc424b9b42089504256d926 K210-022

This patch adds a TCP SYN cache which reduces and/or eliminates the effects of SYN-type denial of service attacks such as those discussed in CERT advisory CA 96.21.

## **PATCH**

U210-025 (<ftp://ftp.bsdi.com/bsdi/patches/patches-2.1/U210-025>)

md5 checksum: d2ee01238ab6040e9b7a1bd2c3bf1016 U210-025

This patch should be installed in conjunction with IP source address check and IP fragmentation queue limit patch (K210-021) and SYN flooding patch (K210-022).

Additional details about these patches are available from

<http://www.bsdi.com>  
<ftp://ftp.bsdi.com>

## **Hewlett-Packard Company**

HPSBUX9704-060

Description: SYN Flooding Security Vulnerability in HP-UX

HEWLETT-PACKARD SECURITY BULLETIN: #00060

Security Bulletins are available from the HP Electronic Support Center via electronic mail.

User your browser to get to the HP Electronic Support Center page at:

<http://us-support.external.hp.com>  
(for US, Canada, Asia-Pacific, & Latin-America)

<http://europe-support.external.hp.com>  
(for Europe)

## **IBM Corporation**

Any system that is connected to a TCP/IP-based network (Internet or intranet) and offers TCP-based services is vulnerable to the SYN flood attack. The attack does not distinguish between operating systems, software version levels, or hardware platforms; all systems are vulnerable. IBM has released AIX operating system fixes for the SYN flood vulnerability.

NOTE: If you are using the IBM Internet Connection Secured Network Gateway (SNG) firewall software, you must also apply the fixes listed in the next section.

The following Automated Program Analysis Reports (APARs) for IBM AIX are now available to address the SYN flood attack:

### **AIX 3.2.5**

No APAR available; upgrade to AIX 4.x recommended

### **AIX 4.1.x**

APAR - IX62476

### **AIX 4.2.x**

APAR - IX62428

## **Fixes for IBM SNG Firewall**

The following Automated Program Analysis Reports (APARs) for the IBM Internet Connection Secured Network Gateway firewall product are now available to address the SYN flood and "Ping o' Death" attacks:

NOTE: The fixes in this section should ONLY be applied to systems running the IBM Internet Connection Secured Network Gateway (SNG) firewall software. They should be applied IN ADDITION TO the IBM AIX fixes listed in the previous section.

### **IBM SNG V2.1**

APAR - IR33376 PTF UR46673

### **IBM SNG V2.2**

APAR - IR33484 PTF UR46641

## **Obtaining Fixes**

IBM AIX APARs may be ordered using Electronic Fix Distribution (via the FixDist program), or from the IBM Support Center. For more information on FixDist, and to obtain fixes via the Internet, please reference

<http://service.software.ibm.com/aixsupport/>

or send electronic mail to ["aixserv@austin.ibm.com"](mailto:aixserv@austin.ibm.com) with the word "FixDist" in the "Subject:" line.

## **Linux**

A patch for version 2.0.29 of the linux kernel source is available from:

<http://www.kernel.org/pub/linux/kernel/v2.0/patch-2.0.30.gz>

The patch allows tcp/ip processing to continue as normal, until the queue gets close to full. Then, instead of just sending the synack back, it sends a syn cookie back, and waits for a response to IT before sending the synack. When it sends the cookie, it clears the syn from the queue, so while under attack, the queue will never fill up. Cookies expire shortly after they are sent. Basically this prevents people from filling up the queue completely. No one flooding from a spoof will be able to reply to the cookie, so nothing can be overloaded. And if they aren't flooding from a spoof, they would be getting a cookie they would have to respond to, and would have a hard time responding to all the cookies and continuing the flood.

## **Livingston Enterprises, Inc.**

Refer to the following Applications Note for more information on configuring a Livingston IRX or PortMaster to help block outgoing SYN attacks from an ISP's users:

<ftp://ftp.livingston.com/pub/le/doc/notes/filters.syn-attack>

## **Silicon Graphics, Inc.**

Updated Silicon Graphics information concerning SYN attacks can be found in SGI Security Advisory, "IRIX IP Spoofing/TCP Sequence Attack Update," 19961202-01-PX, issued on August 6, 1998.

Patches are available via anonymous FTP and your service/support provider.

The SGI anonymous FTP site is [sgigate.sgi.com](ftp://sgigate.sgi.com) (204.94.209.1) or its mirror, [ftp.sgi.com](ftp://ftp.sgi.com). Security information and patches can be found in the `~ftp/security` and `~ftp/patches` directories, respectfully.

For subscribing to the wiretap mailing list and other SGI security related information, please refer to the Silicon Graphics Security Headquarters website located at:

<http://www.sgi.com/Support/security>

## **Sun Microsystems, Inc.**

Sun published a bulletin on October 9, 1996--Sun security bulletin number 00136. Sun Security Bulletins are available via the [security-alert@sun.com](mailto:security-alert@sun.com) alias and on SunSolve.

Note: Advisories from vendors listed in this section can also be found at <ftp://ftp.cert.org/pub/vendors/>

---

This document is available from: <http://www.cert.org/advisories/CA-1996-21.html>

---



## VII.

## CERT/CC CONTACT INFORMATION

**Email:** [cert@cert.org](mailto:cert@cert.org)

**Phone:** +1 412-268-7090 (24-hour hotline)

**Fax:** +1 412-268-6989

**Postal address:**

CERT Coordination Center  
Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh PA 15213-3890  
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

### Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from

[http://www.cert.org/CERT\\_PGP.key](http://www.cert.org/CERT_PGP.key)

If you prefer to use DES, please call the CERT hotline for more information.

### Getting security information

CERT publications and other security information are available from our web site

<http://www.cert.org/>

\* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

---

### **NO WARRANTY**

Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

---

[Conditions for use, disclaimers, and sponsorship information](#)

Copyright 1996, 1997, 1998, 1999, 2000 Carnegie Mellon University.

---

## Revision History

Nov. 29, 2000	Updated vendor information for the Linux kernel.
Aug. 24, 1998	Updated vendor information for Silicon Graphics, Inc.
Sep, 24, 1997	Updated copyright statement
July 18, 1997	Updates - added information
May 08, 1997	Updates - updated vendor information for Hewlett-Packard.
Jan. 02, 1997	Updates - added or modified vendor information for SGI, Livingston, HP, 3COM.
Dec. 19, 1996	Updates - corrected Sun Microsystems security-alert email address.
Dec. 10, 1996	Appendix A, #3 - corrected next to last reserved private network number entry.
Dec. 09, 1996	Updates - added IBM patch information.
Nov. 12, 1996	Introduction, paragraph 2 - added some clarification.
Oct. 10, 1996	Updates - added a pointer to Sun Microsystems advisory. added a pointer to the CERT /pub/vendors directory.
Oct. 08, 1996	Appendix A, #3 - revised the last item, reserved private network numbers Updates - added BSDI patch information.
Oct. 07, 1996	Updates - added a pointer to Silicon Graphics advisory.
Sep. 24, 1996	Modified the supersession statement

# Distributed Denial-of-Service Attacks and You

9/5/2005 14h:2

It sounds like a bad 1950s monster movie: "Attack of the Killer Zombies." Unfortunately, though, it could easily be a headline from the recent spate of network attacks targeted at corporate, government, education, and private computer systems. In this article, you will learn how these attacks work and how to keep from becoming an unwitting *zombie* in the army of a malicious attacker.

## On This Page

[the Problem](#)  
[I to Death by Ducks](#)  
[Vork](#)  
[: Yourself](#)  
[1 More](#)

## VIII. UNDERSTANDING THE PROBLEM

The more technical term for a zombie attack is a *distributed denial-of-service* (DDoS) attack. This sounds like an intimidating term, but it's actually pretty simple. Let's start with the "denial of service" part. Imagine that your town's emergency response center only had one phone line, and that a prankster kept calling it and hanging up. During the time the phone line was in use, no one else could report a legitimate emergency. That's denial of service.

Of course, this would not be a recommended way to design an emergency service, so it's safe to assume that even small towns have several inbound lines. Let's say you live in a mid-size town that has 100 inbound 911

lines. What would happen if a coordinated gang of 120 bad actors went to pay phones and simultaneously called 911. The call center would be flooded, and legitimate inbound calls wouldn't be able to get through. Unlike the earlier attack, though, it's harder to prevent coordinated action from many people at once, especially if you don't know who or where they are. That's where the "distributed" in DDoS comes from.

What does this have to do with *zombies*? Simple. Imagine that you're a malicious attacker who can trick someone into running a program of your choice. That program can lie dormant, like a monster-movie zombie, until you send it a signal; at that point, it can begin generating network traffic sent to a particular target. If you can recruit enough zombies, you can flood even very large services like Yahoo! or CNN, for example.

[Top of page](#)

## **IX. GETTING PECKED TO DEATH BY DUCKS**

This may seem counterintuitive at first how can a few individual systems tie up a behemoth like CNN? The answer is twofold: scale and bandwidth. Let's say you can get 500 machines, each of which is on a garden-variety DSL line, to attack a single host. A typical DSL line allows 256Kb/s of upstream bandwidth, so let's conservatively say that the 500 machines can generate  $500 * 256\text{Kb/s} = 128000\text{ Kb/s} = 128\text{ Mb/s}$ . As it turns out, this is roughly the size of 42 T1 lines, or about 1.4 T3 lines a considerable amount of bandwidth. If the target only has a single T1 (or even only a dozen), it's toast. Of course, varying the number of hosts and their average bandwidth changes the traffic volume that can be brought to bear, but the important point is that a moderate number of independent systems, acting together, can easily flood even very large networks. There are other tricks that can be used to increase the effectiveness of these attacks, but you don't have to understand them to protect yourself. (If you do want more details, SANS maintains an excellent [summary of attack methods](#).)

[Top of page](#)

## **X. HOW ATTACKS WORK**

I mentioned earlier that DDoS attacks depend on getting the DDoS client to run on a wide range of machines. The usual trick is to package it as a "[Trojan horse](#)", an innocuous-looking but secretly malicious program that unsuspecting people will run. The key word is "unsuspecting". Many computer users don't think twice about running executable programs or attachments that they get from unknown sources, especially if they think that the program they've downloaded will give them something (like free porn or lottery winnings) for nothing. For example, I know of one DDoS attacker who would pose in AOL chat rooms as a teenage girl, offering self-running slide shows of "herself" to people who chatted with her. Of course, the "slide show" was actually a disguised Trojan; over time, this attacker was able to bag a large number of DDoS clients to do his bidding. Of course, someone who's trying to build an arsenal of DDoS clients doesn't want to waste time sending Trojans to well-educated security professionals; better to send them to people who will run the malicious code without a second thought, especially if they aren't likely to notice that their machine is infected.

Once a Trojan is activated, one of the first things it typically does is register its presence somewhere, usually by sending TCP/IP packets to a well-known destination. The popular [SubSeven](#) Trojan registers itself by sending messages to an attacker-selected IRC channel. These registration messages usually indicate the IP address of the zombied machine, and it may include some useful information like the apparent bandwidth between the zombie and a preselected target.

Once a machine's infected, it typically stays that way. Depending on the Trojan, it may actively attempt to disguise itself (as with the recent Code Red variant that added a second, bogus copy of *explorer.exe* to infected machines), or it may depend on user inattention to stay hidden. Whenever the attacker desires to, he can send a trigger command to one, or all, infected zombies; that command will tell the Trojan to attack a designated target by sending it lots of packets. The Trojan may also attempt to spread itself, and many Trojans offer an attacker direct remote control of a compromised machine.

[Top of page](#)

## XI. HOW TO PROTECT YOURSELF

It's hard to defend against DDoS attacks. There are actually two separate issues: keeping your network from being attacked by others and hardening your machines so they can't be compromised and used in attacks.

The first step is to protect your own network against being attacked. This is hard to do, since any network is vulnerable to being overloaded by seemingly-legitimate traffic. Turning on ingress filtering (as described in [RFC 2267](#)) will help screen out junk packets. In addition, there are a number of settings for the Windows NT/2000 TCP/IP stack that you can adjust to harden it against common attacks. Here's what to do:

.It's [security bulletin site](#). Download all the pertinent patches that you don't already have is now, then come back and finish reading. (While you're there, sign up to [get bulletins](#)

.: firewall to block (or, better, ignore) traffic on any port you don't actually need. If you don't rts should be open for particular services, see Microsoft Knowledge Base (KB) article

.P/IP hardening settings described in "[Security Considerations for Network Attacks](#)". Apply ver which is exposed directly to the Internet.

Unfortunately, protecting your machines against attacks can be difficult because attackers keep changing their *modus operandi*. It's simpler to prevent your computers from becoming zombies and contributing to the DDoS problem. A few simple steps **that you can take now** will do the trick:

.achines against compromise. If your machines don't get Trojans on them in the first place, as DDoS participants.

.n't run attachments you get from unknown or untrusted sources. Be careful with programs that claim to come from-- adult sites, online casinos, and the like. If you're running Windows 2000 P, never use the Administrator account, or any account with similar privileges, for routine i Trojan is bad; it's worse when that Trojan runs with administrative privileges.

.One great way to help keep Trojans off your machines is to deploy the [Outlook E-mail e](#) (built in to Outlook 2002, or available as a [download](#) for [Outlook 2000/98](#)). This update types of attachments may be sent and opened using Outlook.

.ing system's security measures. For Windows 2000 and Windows XP, make sure that [System](#) (SFC) is enabled. If you're using Windows XP, be sure that you have the [Internet Connection](#) enabled. By default, it's turned off; turning it on greatly strengthens your Internet-connected defenses. ICF provides a degree of egress filtering as well.

.ivity. If you have computers that are directly connected to the Internet, configure your routers that their connectivity is restricted. For example, on a typical client machine you might only i a few selected ports (HTTP, SMTP, FTP, IMAP, and POP) to pass from those machines 3wall.

*software*. The major antivirus vendors are all very good at quickly producing updates when a release is in the wild. Diligent use of these tools will help keep your machines clean, and you use them to scan new files before you execute them.

*Firewall on individual machines*. There are a number of good-quality "personal firewalls" Windows XP includes ICF, and other vendors, such as [ZoneAlarm](#) and [Symantec](#), make firewalls parts of Windows. These programs filter inbound and outbound traffic, alerting you when unusual happens. For example, ZoneAlarm will warn you every time an application on your computer tries to open an outbound TCP/IP connection. By paying attention to its warnings, you can quickly distinguish harmless traffic (HTTP requests from Internet Explorer, for example) from suspicious traffic that is allegedly originating from *winword.exe*.

*Filtering*. P.T. Barnum is famously reputed to have fleeced people who attended his circus by directing them "This Way to Egress", but if you understand that [egress filtering](#) is a way for which packets are allowed to leave your network you'll understand why it's useful. In general, it keeps your routers from forwarding any packets whose actual source address doesn't match the destination. DoS attacks commonly involve spoofing the origin address of the attack packets; with egress filtering, your network will never forward those packets out to the rest of the world.

This sounds so banal as to be worthless, but it's actually worth repeating, if only because not everyone does it as diligently as they should. To help keep safe:

*Network traffic*. This is particularly useful for home users and small networks, where you have a good idea of what "normal" outbound traffic looks like. If your cable modem normally has a few blinks per day suddenly the "transmit" light comes on and stays on, that's a sign that something unusual is happening.

*News*. I'm not talking about CNN, but regular visits to <http://www.microsoft.com/security>, the Emergency Response Team ([CERT](#)) website, and places like [ntbugtraq.com](#) will help keep you up to date on the latest threats and countermeasures.

*Warning fatigue*. When you use a tool like ZoneAlarm, it's easy to get conditioned to the warning messages, to the point where you blindly click "OK" when it asks you whether program X should access the Internet. Don't get complacent!

[Top of page](#)

## XII.

### WHERE TO LEARN MORE

The Emergency Response Team ([CERT](#)) keeps tabs on emerging attacks and technologies. In their advisory to [home users](#) contains a lot of good information you might consider sending it to all network users so they can secure their own machines at home.

There is a [tool](#) for checking your network computers to make sure they have all relevant security updates and use it often.

[iNet](#) keeps an archive of [security columns](#); they make good background reading, and many recommendations for Windows 98 and Windows 2000 computers.

[iNet](#) also maintains a [master summary page](#) covering DDoS-related attacks.

Paul Robichaux is the principal of [Robichaux & Associates, Inc.](#), which provides programming, technical communications, and security services to customers ranging in size from local auto dealerships to Microsoft. He's glad to have his latest book (Managing Microsoft Exchange Server, published by O'Reilly & Associates) out on the shelves so he can spend more time with his family.

For any feedback or comments in regards to the content of this article, please send them to [Microsoft TechNet](#).

We at Microsoft Corporation hope that the information in this work is valuable to you. Your use of the information contained in this work, however, is at your sole risk. All information in this work is provided "as -is", without any warranty, whether express or implied, of its accuracy, completeness, fitness for a particular purpose, title or non-infringement, and none of the third-party products or information mentioned in the work are authored, recommended, supported or guaranteed by Microsoft Corporation. Microsoft Corporation shall not be liable for any damages you may sustain by using this information, whether direct, indirect, special, incidental or consequential, even if it has been advised of the possibility of such damages.

*The Microsoft.com*

# Xây dựng bộ đếm cho Website bằng ngôn ngữ ASP

25/3/2005 11h:16

**Bạn muốn biết tại một thời điểm có bao nhiêu người đang truy cập website của mình? Việc này không thể thực hiện được với mã lệnh HTML thuần túy hay JavaScript mà phải dùng đến ngôn ngữ kịch bản chạy trên server (server-side scripting language) như ASP, PHP hay JSP.**

Bài viết này giới thiệu cách hiển thị thông tin thống kê về số người đang truy cập (online), số lượt truy cập, số lần duyệt xem các trang trong website bằng ngôn ngữ ASP thông dụng của Microsoft. Những thông tin về số lượt truy cập được gọi là bộ đếm cho website - một trong những ứng dụng cơ bản mà các website chuyên nghiệp thường dùng đến. Bộ đếm giúp cho chính bạn và khách truy cập biết được website hoạt động hiệu quả như thế nào, “đông khách” hay “vắng khách”.

## 1. Hiển thị số người đang truy cập website:

Mỗi ứng dụng web ASP đều có thể có một file cấu hình chung là GLOBAL.ASA. File GLOBAL.ASA được server đọc và thực thi khi ứng dụng web bắt đầu chạy mỗi khi có một yêu cầu tải trang web từ client (phía người dùng). Trong file GLOBAL.ASA thường có các thủ tục sau:

**Application\_OnStart:** Được gọi khi ứng dụng web bắt đầu chạy. Như vậy thủ tục này chỉ được gọi duy nhất một lần khi website của bạn có yêu cầu truy cập đầu tiên cho tới khi ứng dụng web kết thúc hoặc file GLOBAL.ASA có thay đổi. Chúng ta sẽ khởi tạo biến đếm số người đang truy cập website tại đây:

```
Application("So_Nguoi_Online") = 0
```

Biến Application có ảnh hưởng tới toàn bộ ứng dụng web, tức là tất cả các trang của website đều dùng chung biến này.

**Session\_OnStart:** Được gọi khi có một yêu cầu kết nối của client tới server được thiết lập, tức là khi người dùng gõ vào địa chỉ của website trong trình duyệt để truy cập. Khi đó, chúng ta sẽ tăng biến đếm số người online:

```
Application("So_Nguoi_Online") = Application("So_Nguoi_Online") + 1
```

**Session\_OnEnd:** Được gọi khi client ngắt kết nối với server, tức là khi người dùng chuyển qua trang web khác hoặc đóng trình duyệt lại. Khi đó chúng ta sẽ giảm biến đếm số người online:

*Application("So\_Nguoi\_Online") = Application("So\_Nguoi\_Online") - 1*

Nội dung của file GLOBAL.ASA như sau:

*<SCRIPT LANGUAGE="VbScript" RUNAT="Server">*

*Sub Application\_OnStart()*

*Khởi tạo giá trị ban đầu cho biến = 0*

*Application("So\_Nguoi\_Online") = 0*

*End Sub*

*Sub Session\_OnStart()*

Mỗi khi có một yêu cầu truy cập khác thì tăng số người truy cập lên 1

Khóa biến Application để tránh trường hợp có nhiều yêu cầu truy cập cùng một lúc

*Application.Lock*

*Application("So\_Nguoi\_Online") = Application("So\_Nguoi\_Online") + 1*

Mở khóa biến Application để yêu cầu truy cập sau sẽ thay đổi được giá trị của biến

*Application.Unlock*

*End Sub*

*Sub Session\_OnEnd()*

Mỗi khi có một kết nối bị ngắt thì giảm số người truy cập đi 1

*Application.Lock*

*Application("So\_Nguoi\_Online") = Application("So\_Nguoi\_Online") - 1*

*Application.Unlock*

*End Sub*

*</SCRIPT>*

Bạn phải đặt file GLOBAL.ASA trong thư mục gốc của website của mình, sau đó để hiển thị số người đang truy cập website bạn thực hiện như sau:

*<h1>*

*Số người đang xem website: <% = Application("So\_Nguoi\_Online") %> </h1>*

2. Hiển thị số lượt truy cập website:

Số lượt truy cập website được tính bằng số lần yêu cầu truy cập website trên server của bạn. Số lượt truy cập website luôn tăng, do đó chúng ta chỉ cần khai báo biến trong thủ tục Application\_OnStart:

*Application("So\_Luot\_Truy\_Cap") = 0*

Và tăng biến này lên 1 trong thủ tục Session\_OnStart:

*Application("So\_Luot\_Truy\_Cap") = Application("So\_Luot\_Truy\_Cap") + 1*

Và hiển thị trong website:

*Số lượt truy cập: <% = Application("So\_Luot\_Truy\_Cap")*

### **3. Hiển thị số lần duyệt xem các trang của website:**

Trong một phiên truy cập thì người dùng có thể duyệt xem nhiều trang trong website của bạn, chúng ta phải lưu thông tin này vào cơ sở dữ liệu. Mỗi trang ASP trong website của bạn sẽ include file chứa bộ đếm (chúng ta tạm đặt tên là bodem.asp) vào đầu trang để đếm số lần duyệt xem website.

Chúng ta sẽ dùng cơ sở dữ liệu Access để lưu trữ thông tin về số lần duyệt xem website. Bạn tạo một bảng DUYETXEM với các trường như sau:

Tên trường Kiểu

*ID AutoNumber*

*SO\_LAN\_XEM Long Integer*

Nội dung của file bodem.asp như sau:

*<%*

*'Biến để truy xuất CSDL*

*Dim rs*

*'Biến ghi số lần duyệt xem các trang trong website*

*Dim SoLanXem*

*'Chứa chuỗi lệnh SQL cần thực hiện*

*Dim strSQL*

*'Chuỗi kết nối với CSDL*

*Dim strConnection*

*'Biến thiết lập kết nối*



*Dim myConnection*

*'Đầu tiên xác định chuỗi kết nối với CSDL, thay đổi đường dẫn tới CSDL*

*'cho phù hợp với đường dẫn và tên CSDL của website của bạn*

*strConnection = "provider=microsoft.jet.oledb.4.0; data source = " &  
Server.MapPath("../db/DATA.MDB")*

*'Câu lệnh SQL dùng để truy xuất CSDL*

*strSQL = "SELECT \* FROM DUYETXEM WHERE SO\_LAN\_XEM > 0"*

*'Khởi tạo kết nối*

*Set myConnection = Server.CreateObject("ADODB.Connection")*

*'Mở kết nối và thực hiện câu lệnh SQL*

*myConnection.Open strConnection*

*Set rs = myConnection.Execute(strSQL)*

*'Nếu số lần xem > 0 thì tăng biến đếm lên 1*

*If Not rs.EOF Then*

*SoLanXem = rs("SO\_LAN\_XEM")*

*SoLanXem = (SoLanXem + 1)*

*'Cập nhật vào CSDL*

*strSQL = "UPDATE DUYETXEM SET SO\_LAN\_XEM = " & SoLanXem  
Set rs = myConnection.Execute(strSQL)*

*Else 'Nếu chưa có lần xem nào*

*SoLanXem = 1*

*'Thêm vào CSDL*

*strSQL = "INSERT INTO DUYETXEM (SO\_LAN\_XEM) VALUES (" & SoLanXem & ");"*

*Set rs = myConnection.Execute(strSQL)*

*End If*

*'Đóng kết nối*

*Set myConnection = Nothing*

*Set rs = Nothing*

*%>*

Trong đầu mỗi trang ASP bạn include file bodem.asp vào và hiển thị như sau:

```
<!--#INCLUDE FILE = bodem.asp -->
```

```
<%
```

```
Response.Write("Trang web đã được duyệt " & SoLanXem & " lần")
```

```
%>
```

Chúc các bạn thành công!

**Hà Minh Nam**

ha\_minh\_nam@hotmail.com

*Theo PCWorld Vietnam*

## **Xây dựng một ứng dụng từ điển đơn giản** - 25/10/2004 10h:32

**Công cụ cần thiết nhất khi học ngoại ngữ chính là từ điển. Chắc hẳn các bạn cũng như tôi luôn cảm thấy vất vả khi phải tra từ trên cuốn từ điển dày cộp. Giải pháp đáng giá là sử dụng ứng dụng từ điển trên máy vi tính. Mặc dù hiện nay các ứng dụng từ điển đã có nhiều nhưng vốn là dân tin học, tôi đã quyết định tự xây dựng cho mình một ứng dụng từ điển riêng.**

### **Cơ sở dữ liệu**

Phần quan trọng nhất đối với một ứng dụng từ điển chính là cơ sở dữ liệu (CSDL). Việc xây dựng CSDL cho từ điển phải đảm bảo được khả năng truy cập nhanh bởi dữ liệu của từ điển thường khá lớn, lên tới hàng chục nghìn từ. Thật may, DICT.ORG ([www.dict.org](http://www.dict.org)) đã xây dựng một format (định dạng) từ điển rất dễ sử dụng, format này đã được dùng để xây dựng những bộ từ điển khá lớn. Dict format được mô tả như sau: toàn bộ CSDL được chứa trong 2 file, một file chứa nghĩa của từ và một file index. File index bao gồm tên từ, vị trí nghĩa của từ bắt đầu trong file chứa nghĩa và độ dài của nghĩa. Vị trí bắt đầu và độ dài của nghĩa được mã hoá theo cách như sau: Sử dụng 64 chữ cái:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-

chữ cái A tương đương số 0, chữ cái B tương đương số 1 v.v... Giữa từ, vị trí bắt đầu và độ dài nghĩa phân cách nhau bởi ký tự tab (ASCII 9). Mỗi dòng trong file index chứa dữ liệu của một từ. Các dòng phân cách nhau bởi ký tự xuống dòng (ASCII 10).

Ví dụ trong file index của từ điển Đức-Việt có một dòng như sau:

Abdeckung kbpP D3

Như vậy nghĩa của từ Abdeckung trong file chứa nghĩa sẽ bắt đầu tại offset kbpP (theo mã 64 ký tự) và có độ dài là D3.

Việc chuyển từ mã cơ số 64 về cơ số 10 được thực hiện như sau:

Đối với vị trí bắt đầu: kbpP. Ta có k (ở cơ số 64) = 36 (ở cơ số 10), b = 27, p = 41, P = 15.  
Như vậy chuyển sang cơ số 10, mã kbpP có giá trị là:  $36 \cdot 64^3 + 27 \cdot 64^2 + 41 \cdot 64 + 15 = 9550415$

Đối với độ dài nghĩa: D3. Ta có D = 3, 3 = 55. Như vậy chuyển sang cơ số 10, mã D3 ở cơ số 64 có giá trị là: 247.

File index được sắp xếp để giảm bớt thời gian tìm kiếm. Việc mã hoá theo cơ số 64 như trên giúp cho kích thước file index giảm xuống rất nhiều so với không mã hóa.

Còn cấu trúc của file chứa nghĩa gồm các phần như sau:

@headword

\* tu loại (noun, verb...)

- định nghĩa 1

= câu ví dụ cho định nghĩa 1 + nghĩa của câu do

- định nghĩa 2

= câu ví dụ cho định nghĩa 2 + nghĩa của câu do

\* tu loại

- định nghĩa 3

Nghĩa của mỗi từ gồm một phần như trên, các nghĩa của mỗi từ nối tiếp nhau liên tục.

Như vậy, các bạn hoàn toàn có thể xây dựng cho mình các bộ từ điển riêng. Tuy nhiên, công việc nhập dữ liệu lại không hề đơn giản chút nào. Nhưng, lại một lần nữa, chúng ta thật may mắn vì có một số bạn đã bỏ công nhập sẵn một số bộ từ điển thông dụng. Các bạn có thể tham khảo thêm tại: [www.ttdomain.net/ttdownload/](http://www.ttdomain.net/ttdownload/), [www.informatik.uni-leipzig.de/~duc/Dict/](http://www.informatik.uni-leipzig.de/~duc/Dict/), [huybien.vze.com](http://huybien.vze.com)... Ngoài ra còn rất nhiều bộ từ điển chuyên dụng khác nữa, các bạn có thể tham khảo thêm ở các địa chỉ trên hoặc tại [www.dict.org](http://www.dict.org).

## Xây dựng chương trình

Ở đây tôi xin trình bày cách sử dụng ngôn ngữ Visual C++ 6.0 và thư viện MFC. Các bạn hoàn toàn có thể sử dụng các ngôn ngữ khác. Trong khuôn khổ bài viết, tôi chỉ xin đưa ra những phần chủ yếu nhất. Các phần như thiết kế giao diện, bố trí giao diện v.v... các bạn có thể tham khảo chương trình mẫu và tự sáng tạo thêm.

### 1. Các thành phần giao diện cơ bản:

- **Edit Box:** dùng để nhập từ. Cài biến cho điều khiển: 'Variable name: m\_word, Category: Value, Type: CString;'

- **WebBrowser:** dùng để hiện nghĩa của từ. Việc sử dụng WebBrowser chỉ nhằm mục đích

hiển thị nghĩa trực quan và sinh động hơn bằng cách xử lý chuỗi (sẽ được đề cập sau này). Bạn có thể thêm điều khiển ActiveX Web Browser vào ứng dụng của mình bằng cách chọn Project->Add to Project->Components And Controls, chọn trong thư mục Registered ActiveX Controls điều khiển Microsoft Web Browser. Cài biến cho điều khiển: 'Variable name : m\_wordmean;'

- **Listbox, dùng để hiện danh sách từ.** Cài biến cho điều khiển: 'Variable name: m\_wordlist; Category: Control;'

- **Listbox, dùng để lưu trữ dữ liệu về từ.** Cài biến cho điều khiển: 'Variable name: m\_worddata, Category:Control;'

## 2. Mã chương trình:

- **Nạp dữ liệu vào các list box:** bạn đặt phần mã này ở sự kiện WM\_OnInitDialog() để dữ liệu được nạp ngay từ lúc khởi động chương trình. Ở đây, bạn thay tên file index bằng tên file tương ứng với từ điển bạn sử dụng.

```
FILE *inFile;

inFile = fopen ('mydic.index','r');

if (inFile == NULL){

MessageBox ('Cannot open index file');

}else {

    char * line;

    char lineBuf[100];

    line = (char *) lineBuf;

    m_wordlist.ResetContent();

    m_worddata.ResetContent();

    CString word = "";

    CString sWord = "";

    CString sData = "";

    while (!feof(inFile)){

        fgets(line,99,inFile);

        if (strlen(line)>=2){

            word = line;
```

```

        int pos = word.Find('t',0);

        sWord = word.Left(pos);

        sData = word.Mid(pos+1,word.

            GetLength()-pos-1);

        if (sData.Find('\n',0) > 0){

            sData = sData.Left(sData.

                GetLength()-1); }

        if (sWord.GetLength()>=1){

            m_wordlist.AddString(sWord);

            m_worddata.AddString(sData);}

    } } }

fclose(inFile);

```

• **Hàm chuyển từ mã cơ số 64 sang cơ số 10**

```

int GetDemicalValue (CString str){

    CString base64 = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/';

    int decValue = 0;

    int len = str.GetLength();

    for (int i = 0; i<len; i++){

        int pos = base64.Find(str.

            GetAt(i),0);

        decValue += (int)pow(64,len-i-

            1)*pos;

    }

    return decValue;

}

```

- Hàm xử lý chuỗi ký tự nghĩa. Như đã đề cập ở trên, tôi sử dụng điều khiển Web Browser để hiển thị nghĩa cho thêm phần sinh động. Hàm này có tác dụng xử lý chuỗi ký tự nghĩa bằng cách thêm các tag HTML để nghĩa được thể hiện sinh động hơn ví dụ như: các nghĩa con thì in đậm, chữ đỏ, các ví dụ thì in nghiêng, chữ xanh v.v...

```
CString ChangeStyle(CString wordmean){
    CString meaning = wordmean;

    meaning = meaning.Right(meaning.GetLength()-1);

    int pos = meaning.Find('\n',1);

    meaning.Insert(pos,'<b>');

    meaning = '<b>' + meaning;

    meaning.Replace('\n','<br>');

    meaning.Replace('{','<font color=\#FF0000\><b>');

    meaning.Replace('}','</b></font>');

    meaning.Replace('[','<font color=\#FF0000\><b>');

    meaning.Replace(']',</b></font>');

    meaning.Replace('+',' ');

    return meaning;
}
```

- Hàm lấy nghĩa của từ: hàm này có tác dụng đọc từ file chứa nghĩa để lấy nghĩa của từ, sau đó xử lý chuỗi nghĩa rồi ghi ra file temp.htm. Nếu lấy nghĩa thành công, hàm trả về giá trị TRUE, nếu lấy nghĩa không thành công, hàm trả về giá trị FALSE. Bạn thay tên file mydict.dict bằng tên của file từ điển tương ứng.

```
BOOL CXDictDlg::GetMeaning (){
    CFile f;

    CString meaning="";

    if (f.Open('mydic.dict',CFile::modeRead) == FALSE) {

        meaning = 'Can not open database file!';

    } else {
```

```

CString sOffLen;

m_worddata.GetText(m_wordlist.GetCurSel(),sOffLen);

int pos = sOffLen.Find("\t",0);

CString sOff = sOffLen.Left(pos);

CString sLen = sOffLen.Right(sOffLen.GetLength()-pos-1);

int iOff = GetDemicalValue(sOff);

int iLength = GetDemicalValue(sLen);

int temp = f.Seek(iOff,CFile::begin);

char buff[64];

DWORD dwRead;

do {

    if (iLength>=64)

        dwRead = f.Read(buff,64);

    else

        dwRead = f.Read(buff,iLength);

    iLength -= dwRead;

    CString stemp = buff;

    stemp = stemp.Left(dwRead);

    meaning += stemp;

}while (iLength>0);

f.Close();

}

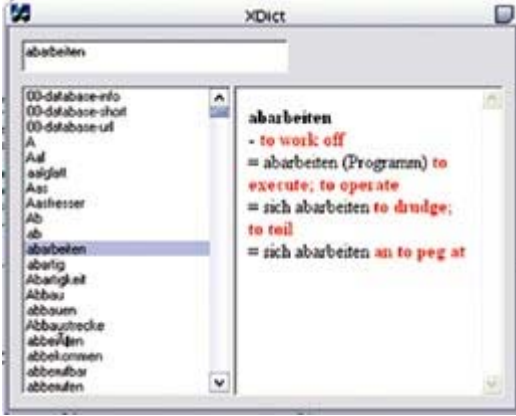
meaning = ChangeStyle(meaning);

CString strHtml("");

strHtml += '<html>\n<head>\n';

strHtml += '<meta http-equiv=\\"Content-Type\" content=\\"text/html; charset=utf-8\">\n';

```



```
strHtml += '</head>\n<body>\n';
```

```
strHtml += meaning + "\n</body>\n</html>";
```

```
CFile f2;
```

```
if (f2.Open('temp.htm',CFile::modeCreate | CFile::modeWrite) ==  
FALSE){
```

```
    MessageBox('Cannot write meaning file!','Error!');
```

```
    return 0;
```

```
}
```

```
f2.Write(strHtml,strHtml.GetLength());
```

```
f2.Close();
```

```
return 1;
```

```
}
```

- Hiện thị nghĩa của từ, bạn cài đoạn mã này vào sự kiện DoubleClick của điều khiển list box.

```
BOOL gm = GetMeaning();
```

```
if (gm){
```

```
    m_wordlist.GetText(m_wordlist.GetCurSel(),m_word);
```

```
    UpdateData(FALSE);
```

## Bảo vệ thông tin 'nhạy cảm' khỏi các chương trình tìm kiếm

21/10/2004 10h:22

Cụm từ “Robots” có thể bạn đã được nghe tới nhiều lần trong các bộ phim khoa học viễn tưởng. Tuy nhiên, nếu bạn sở hữu một trang web, bạn sẽ hiểu đây sẽ là một tệp tin rất quan trọng. Vậy vai trò của của Robots.txt là gì? Làm thế nào để sử dụng Robots.txt? Bài viết giúp bạn vén bớt một phần tấm màn “bí mật” xung quanh tệp tin này.

Khác các công cụ hỗ trợ tìm kiếm khác, tệp tin Robots.txt xác định những gì các cỗ máy tìm kiếm (search engine robot) không được phép “bén mảng”- bao gồm cả các cây thư mục hoặc các tệp tin cụ thể. Robots.txt còn xác định cả các chủng loại công cụ rò tìm thông tin trực tuyến nào được phép rà quét thông tin; ví dụ phân lọc và lưu trữ email, phần mềm “ngửi” nội dung (content sniffer), phần mềm tìm kiếm thông tin dạng thống kê v.v... (bạn có thể duyệt



trang web <http://www.robotstxt.org/wc/active.html> để xem danh sách các chủng loại “robot” và tính năng của chúng). Như vậy, nhìn tổng thể vai trò của Robots.txt là chối bỏ quyền truy cập của một số công cụ tìm kiếm thông tin trực tuyến, bảo vệ nhiều tệp tin nhà quản trị cần thiết phải để “online” nhưng lại không muốn ai cùng “chia sẻ”.

Hầu hết các website không có tệp tin Robots.txt. Đương nhiên, không vì thế mà các trang web không vận hành bình thường hoặc ít được mọi người biết tới. Vậy lý do gì các nhà quản trị nên nhanh chóng tạo lập tệp tin Robots.txt?

**Thứ nhất**, không phải cỗ máy tìm kiếm nào cũng được triển khai với ý đồ tốt hoặc ít nhất là vô hại. Trên thực tế, số lượng lớn các cỗ máy rà quét thông tin để lượm lặt địa chỉ email- công đoạn đầu tiên trong chiến dịch gửi thư rác hàng loạt (spam).

**Thứ hai**, website của bạn có lẽ chưa hoàn thiện và có những thông tin bạn chưa thể ngay lập tức tung ra. Cụ thể, một đường dẫn “không dẫn tới đâu cả” có thể để lại ấn tượng xấu đối với người truy cập.

**Thứ ba**, website của bạn có khu vực dành riêng cho thành viên đăng nhập (có thể tính phí) và bạn không muốn bất cứ công cụ tìm kiếm nào bén mảng tới- Robots.txt là cách duy nhất bạn có thể chặn đứng các ‘spider’.

**Thứ tư**, bạn muốn giữ một số tệp tin “bí mật” (có thể là các thông tin cá nhân). Những tệp tin này không có đường dẫn liên kết tới nhưng lại chứa các đường dẫn liên kết ra bên ngoài. Nếu không sử dụng Robots.txt, khó có thể loại trừ tình huống các công cụ rà quét thông tin “lần” theo đường liên kết ra ngoài của tệp tin nhạy cảm và tìm ra chúng.

### ***Tạo tệp tin Robots.txt: Đơn giản!***

Chỉ cần dùng một chương trình soạn thảo văn bản thông dụng (ví dụ Notepad, Nottab Light...) người dùng không chuyên cũng có thể tạo cho mình một tệp tin Robots.txt. Hãy tham khảo một câu lệnh (syntax) của tệp tin:

*User-agent:*  
*Disallow:*

Dòng lệnh thứ nhất xác định loại công cụ tìm kiếm không được phép rà quét thông tin trong một khu vực được định sẵn. Dòng lệnh thứ hai (có thể là 3, 4 v.v...) xác định “khu vực cấm” đối với các chương trình rà quét thông tin bị “điểm danh” trong dòng lệnh một (User-agent). Sau khi tạo (lưu ở định dạng .txt), hãy đẩy tệp tin lên thư mục gốc (Root- nơi lưu trữ các tệp tin kiểu index.html, index.php, index.asp ...)

Tăng cường hơn nữa tính năng của Robots.txt, bạn có thể tham khảo một số snippets:

```
User-agent: *  
Disallow: /
```

Tổ hợp mã này loại bỏ quyền tìm kiếm đối với mọi công cụ tìm kiếm trong một máy chủ (server) duy nhất.

```
User-agent: *  
Disallow: /cgi-bin/  
Disallow: /tmp/  
Disallow: /private/
```

Tổ hợp mã không cho phép các công cụ tìm kiếm thâm nhập các thư mục /cgi-bin, /tmp và /private.

```
User-agent: esculapio  
Disallow: /secret
```

Tổ hợp phím không cho phép công cụ tìm kiếm esculapio (một chương trình rà quét thông tin theo đường dẫn hoạt động trên nền Linux) thâm nhập thư mục /secret.

Như chúng ta đã thấy, Robots.txt khá thú vị và thực sự là một tệp tin quan trọng và hữu ích. Để tìm hiểu thêm thông tin chi tiết về tệp tin này, bạn có thể truy nhập website <http://www.robotstxt.org>.

## Bước tiến vượt bậc của PHP 5

13/9/2004 16h:49

PHP 5.0 bản chính thức được công bố vào ngày 13/7 vừa qua là một trong những sự kiện gây nhiều chú ý trong cộng đồng các nhà phát triển. PHP hiện là ngôn ngữ lập trình được sử dụng phổ biến nhất để xây dựng các trang web động với 16.251.453 tên miền, và 1.346.521 địa chỉ IP, tính đến ngày 18/7/2004.

PHP 5.0 có những cải tiến vượt bậc so với các bản PHP 4.x trước đây. Cho đến nay, PHP 5 được coi là bản cập nhật đáng kể nhất trong vòng 2 năm trở lại đây trong khi vẫn giữ được thế mạnh vốn có của PHP kể từ thời PHP/FI 2.0 là tốc độ và tính hiệu quả. (Tham khảo lịch sử phát triển PHP trên Wikipedia tại địa chỉ <http://vi.wikipedia.org/wiki/PHP> và các thông tin khác về PHP trên website của cộng đồng PHP Việt Nam tại địa chỉ <http://www.phpvietnam.net>).

Trở lại năm 1998, việc đưa Zend Engine 1 vào bộ lõi của PHP Engine đã đem đến sức mạnh vượt trội của PHP 4. Mặc dù PHP đã thu hút được số lượng người dùng đông đảo nhưng hạn chế chính của nó là không cung cấp thêm công cụ để xây dựng website.

Sáu năm sau đó, cộng đồng đã thực sự nghiêm túc xem xét lại những tồn tại cố hữu của PHP khiến các nhà phát triển gặp nhiều khó khăn khi xử lý một số vấn đề. Trong số đó, các nhà phát triển nhấn mạnh đến việc hỗ trợ lập trình hướng đối tượng trong PHP 4 một cách nửa vời, thư viện mở rộng MySQL không hỗ trợ giao thức máy khách MySQL 4.1 mới, hỗ trợ XML chấp vá và hỗn độn.

PHP 5 ra đời đã cải thiện 3 vấn đề quan trọng trên (được viết lại hoàn toàn) và kết quả thực sự ấn tượng. Tuy nhiên, đây vẫn chưa phải là tất cả những gì tốt đẹp nhất mà PHP 5 mang đến. Trong bài viết này, tôi sẽ đề cập đến 7 đặc điểm nổi bật của PHP 5. Những đặc điểm này cho phép phát triển mã lệnh ngắn hơn, đẹp hơn và cơ động hơn.

### Hỗ trợ mạnh lập trình hướng đối tượng

Mặc dù những tính năng hỗ trợ đối tượng đã được tuyên bố từ PHP 4, tuy nhiên cộng đồng các nhà phát triển đã nhanh chóng nhận ra sự non nớt của chúng. Với PHP 5, các nhà phát triển từ nay có trong tay các phương tiện lập trình hướng đối tượng hoàn thiện (PHP 5 vẫn cho phép làm việc bình thường với kiểu lập trình cấu trúc). Có thể liệt kê một số đặc điểm:

1. Mô hình đối tượng hoàn toàn mới dựa trên nguyên mẫu là mô hình của Java
2. Hàm tạo dựng thống nhất `_construct()`: tạo điều kiện để di chuyển các lớp từ cây phân cấp lớp này sang cây phân cấp lớp khác.
3. Hàm hủy `_destruct()`: tạo điều kiện để ghi lại các thông tin phục vụ cho việc tìm lỗi, đóng kết nối dữ liệu và thực hiện các công việc giải phóng tài nguyên khác.
4. Các thuộc tính và các phương thức `public`, `protected`, và `private`: cho phép định nghĩa tính hữu hình của các thuộc tính lớp. Điều này rất quan trọng để hỗ trợ tính kế thừa và tính bao gói trong lập trình hướng đối tượng.
5. Giao diện (Interface): tính năng này lấy từ mô hình của Java nhằm hỗ trợ đa kế thừa.

6. Các lớp trừu tượng.
7. Cấu trúc gợi nhắc kiểu lớp (class hints).
8. Các thuộc tính và phương thức static.
9. Các thuộc tính và phương thức Final.
10. Một loạt các phương thức hỗ trợ mạnh khác như auto loading, reflection, xử lý lỗi ngoại lệ (exceptions), nhân bản đối tượng (object cloning), toán tử instanceof...

Thêm vào đó, các đối tượng bây giờ đồng thời cả gán lẫn truyền đều theo tham chiếu (reference hoặc handle) thay vì theo giá trị, nhờ vậy lập trình viên không còn phải sử dụng rườm rà các kí hiệu & xuyên suốt chương trình. Các tính năng ngôn ngữ này cũng đã mở đường cho lập trình viên PHP sử dụng tối ưu sức mạnh của các mô hình thiết kế (pattern).

### **Thư viện mở rộng MySQL được viết lại hoàn toàn**

Hệ quản trị CSDL MySQL song hành với PHP như hình với bóng, mặc dù PHP có khả năng hỗ trợ nhiều hệ CSDL khác. Tuy nhiên, trong khi sự kết hợp này đem đến những thành công vang dội cho cả hai sản phẩm thì sự lạc hậu của mô đun mở rộng để PHP làm việc với MySQL lại là một trở ngại khi nó không cho phép tận dụng các tính năng cực kì hấp dẫn của MySQL 4.1 và các bản cao hơn.

Nhóm phát triển PHP 5 đã viết một thư viện mở rộng hoàn toàn mới, rất mạnh để hỗ trợ MySQL 4.1 và 5.x, mang tên MySQLi (MySQL Improved). Các tính năng bao gồm:

1. Câu lệnh biên dịch trước
2. Tham số xuất và nhập có giới hạn
3. Kết nối SSL
4. Hàm đa truy vấn.

MySQLi còn tận dụng được hỗ trợ lập trình hướng đối tượng mới của PHP 5 để cung cấp một giao diện hướng đối tượng với MySQL. Trên hết, những bản mới nhất của MySQL bây giờ cho phép dùng chọn lọc con (subselects), chuyển tác (transactions) và bản sao dữ liệu (replication).

### **Bộ công cụ XML có thể tương tác**

PHP 5 đã cải tiến hỗ trợ XML với các thư viện XML mở rộng như sau:

1. Làm việc được với nhau như một thể thống nhất.
2. Được chuẩn hóa trên một thư viện XML duy nhất: libxml2.
3. Tuân thủ hoàn toàn các đặc tả của W3.

4. Xử lý dữ liệu hiệu quả.
5. Cung cấp công cụ xử lý XML mạnh.

Thêm vào đó, để tiếp tục duy trì tính dễ dàng của PHP trong việc xây dựng ứng dụng web, người ta đã đưa vào một thư viện XML mới cho phép đọc và thay đổi dễ dàng các tài liệu XML. Mô đun mở rộng SimpleXML cho phép tương tác với các thông tin có trong một văn bản XML như thể các thông tin này là mảng hay đối tượng, có thể dùng vòng lặp để duyệt qua chúng, biên tập chúng tại chỗ chỉ bằng cách gán các giá trị mới vào các biến.

Nếu bạn đã biết trước định dạng của tài liệu, ví dụ như khi phân tách các file RSS, dữ liệu cấu hình thì dùng SimpleXML rất tiện.

Nếu thích dùng DOM, bạn sẽ rất hứng thú với mô đun mở rộng DOM của PHP 5, mô đun này đã được viết lại và cải tiến rất nhiều so với bản trong PHP 4.

### **Hệ CSDL nhúng SQLITE**

Sự gắn kết hoàn hảo với MySQL là một lợi thế rất lớn cho PHP 4.x, tuy nhiên nhiều khi ứng dụng web lại không cần nhiều đến khả năng của hệ quản trị CSDL mạnh. Do đó, việc phân phối kèm hệ CSDL nhúng SQLite cho phép nhà phát triển có thêm công cụ sẵn sàng để lưu trữ và truy vấn dữ liệu bằng phương thức SQL mà không cần cài đặt hệ CSDL riêng biệt nào cả. SQLite xử lý chính xác các truy cập khóa chặn (locking) và truy cập đồng thời (concurrent), hai vấn đề nổi cộm khi xử lý các file dữ liệu phẳng (flat file) tự tạo.

SQLite đủ mạnh để xử lý phần lớn các nhu cầu lưu trữ và thao tác dữ liệu thông thường. Nó hỗ trợ:

1. Chuyên tác (transactions):
2. Truy vấn con (Subqueries)
3. Kiểm soát toàn vẹn tham chiếu (triggers)
4. Nhiều tính năng cao cấp liên quan đến CSDL như hỗ trợ View...

SQLite tương thích với phần lớn các tính năng chuẩn SQL92, tuy nhiên bản 3.0 mới nhất chưa hỗ trợ chuyên tác lồng nhau, ràng buộc khóa ngoại, cũng như hỗ trợ kiểm soát toàn vẹn tham chiếu một cách đầy đủ. Bạn thậm chí có thể viết các hàm cho người sử dụng tự định nghĩa bằng PHP và gọi chúng bên trong SQLite. Đây là tính năng nổi bật nhất từ trước đến nay có được trong một mô đun mở rộng liên quan đến CSDL trong PHP.

### **Cơ chế xử lý lỗi sáng sủa hơn với Exceptions**

PHP 5 cung cấp một mô hình kiểm tra lỗi hoàn toàn khác so với PHP 4. Nó có tên là xử lý lỗi ngoại lệ. Với lỗi ngoại lệ, nhà phát triển được giải phóng khỏi việc phải kiểm tra giá trị trả lại của tất cả các hàm. Thay vào đó, bạn có thể tách biệt luận lý lập trình xử lý lỗi và đặt chúng nằm bên cạnh các khối mã.

Lỗi ngoại lệ thường được thấy ở các ngôn ngữ lập trình hướng đối tượng như Java và C++. Hiện tại, chỉ có một số ít mô đun mở rộng của PHP sử dụng lỗi ngoại lệ, nhưng dần dần điều

này sẽ được thực hiện đồng bộ. Tuy nhiên, bạn có thể sử dụng lỗi ngoại lệ cho bất cứ đoạn mã PHP nào của mình.

### **Bản thực thi SOAP cao cấp**

SOAP là thành phần chủ yếu của lĩnh vực dịch vụ web đang phát triển rất nhanh. Hỗ trợ SOAP trong PHP 4 chỉ ở cấp trung bình, không chỉ là việc có rất ít các gói SOAP mà còn ở chỗ các gói quan trọng nhất viết bằng PHP thay vì C nên chậm.

Với PHP 5, cuối cùng chúng ta đã có một mô đun mở rộng cho SOAP viết bằng C. Hiện tại, mô đun này thực thi được phần lớn, mặc dù không phải là tất cả các đặc điểm nêu trong SOAP 1.2. Trong tương lai, mô đun này sẽ được cải thiện nhiều hơn. Hiện tại, so với .NET hay Java thì hỗ trợ SOAP trong PHP vẫn bị xếp ở chiều dưới.

## **CÀI ĐẶT PHP 5 VỚI APACHE 2.0.X VÀ MYSQL TRÊN WINXP**

### Các phần mềm cần thiết:

- Microsoft Windows XP hoặc Windows 2000 (tôi chưa test trên Windows 98).
- Apache 2.0.x (tôi dùng Apache 2.0.50). Bạn có thể tải về phiên bản Apache 2.0.50 mới nhất cho Win32 ở địa chỉ <http://httpd.apache.org/>.
- PHP 5.0 (tôi sẽ cài PHP 5 như là một mô đun của Apache, theo cách mà người ta cài PHP cho các máy chủ). Bạn tải phiên bản PHP 5.0 bản chính thức tại <http://www.php.net/download> hoặc các bản phát triển tại <http://snaps.php.net>.
- MySQL 4.0 bản mới nhất có thể tải về từ <http://dev.mysql.com/get/Downloads/MySQL-4.0/mysql-4.0.20d-win.zip/from/pick#mirrors>.

### CÀI ĐẶT APACHE 2.0

Apache 2.0.x có trình cài đặt tự động và bạn sẽ không gặp nhiều khó khăn để cài đặt phần mềm máy chủ web này trên Windows. Ví dụ, để cài Apache 2.0.50, bạn chỉ việc tải về một file có tên apache\_2.0.50-win32-x86-no\_ssl.msi và chạy nó. Theo mặc định nó được cài đặt vào c:\Program Files\Apache Group\Apache2\ . Tuy nhiên, để tiện việc quản lý máy chủ và các mô đun đi kèm, tôi cài nó vào c:\webserver\Apache2\ . Bạn có thể tham khảo bài viết hướng dẫn trực quan bằng hình ảnh tại <http://www.opensourcevn.org/javavietnam>. Sau khi cài đặt xong, bạn hãy tạm tắt Apache. Tuy nhiên, bạn có thể không cần làm việc này cho đến khi bạn sửa đổi file cấu hình của Apache ở dưới.

### CÀI ĐẶT PHP 5.0

Hiện tại PHP 5 chưa có trình tự động cài đặt. Bạn tải về từ PHP.net file php-5.0.0-Win32.zip với dung lượng chừng 7.6 MB. Giải nén file này vào thư mục mà bạn chọn. Để tiện quản lý, tôi giải nén vào c:\webserver\php5.

Sau khi giải nén, bạn sẽ thấy PHP 5 có một cấu trúc thư mục khác hẳn với PHP 4. Lý do là PHP đã được viết lại gần như toàn bộ để có được những cải thiện về tính năng và tốc độ. Tuy nhiên, bạn không cần phải lo lắng, tất cả những gì bạn cần làm sau khi giải nén là tìm đến tệp tin php.ini-recommended và thực hiện các bước sau:

- Chép nó vào thư mục Apache đã nói đến ở trên và đổi tên nó thành php.ini (c:\webserver\Apache2\php.ini).

- Dùng trình soạn thảo văn bản thuần túy như NotePad hay EditPlus để mở file này.

- Tìm đến dòng `extension_dir = './'`

và thay nó bằng

`extension_dir = 'c:\webserver\php5\ext'`

dụ, c:\webserver\php5\ext) và kiểm tra xem đã có file php\_mysql.dll ở đó chưa. Đây là thư viện để PHP 5 xử lý các lệnh gọi hàm xử lý dữ liệu với MySQL. Nếu bạn tải về PHP 5.0 bản chính thức thì file này chắc chắn có ở đó. Nhưng nếu bạn dùng các bản PHP 5.0 đang được phát triển và xuất ra dưới dạng snapshot

(<http://snaps.php.net>) thì file này có thể bị bỏ ra ngoài. Nếu vậy, bạn hãy tìm nó trong thư mục giải nén và chép nó vào thư mục con ext.

- Chép file libmysql.dll trong thư mục PHP 5 vào thư mục

c:\Windows\system32 hoặc thư mục bin của Apache 2 (ví dụ,

c:\webserver\Apache2\bin\). Nếu không thực hiện bước này, bạn có thể sẽ gặp lỗi 'can't load

c:\webserver\php5\ext\php\_mysql.dll'.

- Kế tiếp, hãy vào thư mục con conf của Apache và tìm đến file httpd.conf. Nếu cài Apache 2.0.x theo mặc định thì đường dẫn của nó là c:\Program Files\Apache Group\Apache2\conf\httpd.conf.

Trong trường hợp của tôi, nó nằm ở c:\webserver\Apache2\conf\httpd.conf.

Mở file này với trình soạn thảo văn bản thuần túy và tìm đến dòng

```
#LoadModule ssl_module  
modules/mod_ssl.so
```

Bạn bổ sung thêm hai dòng sau vào dưới dòng này:

```
LoadModule php5_module
```

```
'c:/webserver/php5/php5apache2.dll'
```

```
AddType application/x-httpd-php .php
```

Dòng LoadModule sẽ cho phép

Apache nạp mô đun thư viện gọi PHP 5 thực hiện các tác vụ xử lý liên quan đến các file php. Bạn cần chú ý thay đổi đường dẫn file php5apache2.dll tùy theo trường hợp của bạn.

Bây giờ là lúc khởi động lại Apache 2. Nếu không có lỗi nào, ở khay hệ thống xuất hiện như hình 1.



- Tìm đến dòng

```
; extension=php_mysql.dll
```

và bỏ dấu ';' ở đầu dòng (áp dụng khi bạn cần dùng các hàm PHP xử lý dữ liệu chứa trong MySQL).

Ngay từ khi phiên bản PHP 5.0 Beta ra mắt lần đầu tiên, các nhà phát triển đã thông báo là nó được thiết lập mặc định không làm việc với MySQL vì trong bộ cài PHP 5.0 đã có chứa SQLite, một hệ quản trị CSDL nhúng, chạy cực nhanh, đủ tính năng để dùng cho doanh nghiệp nhỏ. Ngoài ra, lý do chủ yếu là vấn đề giấy phép GPL. Tuy nhiên, nếu cần đến một hệ quản trị CSDL cao cấp hơn như MySQL thì PHP 5 cho phép bạn cấu hình lại để sẵn sàng làm việc. Nếu không chú ý đến điều này, bạn có thể sẽ gặp phải lỗi như sau:

**Fatal error:** Call to undefined function

mysql\_connect() in

[c:\webserver\webrroot\text\textpattern\lib\txplib\\_db.](#)

php on line 15

- Cũng trong file php.ini này, bạn tìm đến dòng

```
;session.save_path
```

và điều chỉnh nó thành

```
session.save_path = 'c:\webserver\php5\tmp'
```

(tùy theo thư mục mà bạn cài đặt PHP5).

- Kế tiếp, tìm đến dòng [mail function], bên dưới mấy dòng bạn sẽ thấy dòng

```
;sendmail_from
```

Hãy bỏ dấu ';' ở đầu dòng và thay đổi địa chỉ email theo ý bạn.

Ví dụ: [localhost@phpvietnam.net](mailto:localhost@phpvietnam.net)

Cấu hình này sẽ được sử dụng đến khi bạn lập trình với hàm mail() trong PHP để gửi nhận mail với một máy chủ email cài sẵn như QK Server.

- Vào thư mục con ext trong thư mục PHP 5 (ví

## CÀI ĐẶT MYSQL 4.0

Như vậy bạn đã cấu hình xong PHP 5.0 và Apache 2.0 để chúng chạy được cùng nhau. Bạn cũng đã cấu hình để các hàm của PHP 5.0 truy vấn được MySQL 4.0. Tuy nhiên bạn cần cài đặt thêm MySQL 4.0 để có đủ công cụ lập trình PHP 5. Việc cài đặt MySQL 4.0.20 khá đơn giản. Bạn chỉ việc tải về một file .zip. Giải nén và chạy file setup.exe. Phần mềm MySQL sẽ tự động cài đặt vào thư mục mặc định c:\mysql. Sau đó bạn vào thư mục c:\mysql\bin để chạy file winmysqladmin.exe. Khi chạy chương trình này bạn sẽ thấy hiện ra cửa sổ như hình 2.



Cửa sổ này cho phép bạn cung cấp tên/mật khẩu của người có toàn quyền quản trị MySQL Server. Thường thì cấp tên/mật khẩu này là root/localhost hoặc bạn cũng có thể để trống mật khẩu theo mặc định. Sau khi nhập xong và ấn nút OK, cửa sổ này biến mất và trên khay hệ thống sẽ xuất hiện biểu tượng dịch vụ của MySQL Server như hình 3.



Như vậy, bạn đã có MySQL Server chạy trên máy tính của mình và sẵn sàng xử lý các câu truy vấn.

### KIỂM TRA KẾT QUẢ CÀI ĐẶT

Để kiểm tra lại công việc cài đặt, bạn thực hiện hai bước sau:

- Vào thư mục gốc phục vụ web của Apache 2 (Apache2\htdocs), và dùng trình soạn thảo văn bản thuần túy như Notepad để tạo ra một file có tên phpinfo.php với nội dung:

```
<?php  
echo phpinfo();  
>
```

- Mở trình duyệt và gõ vào địa chỉ <http://localhost/phpinfo.php>. Bạn sẽ thấy hiện ra toàn bộ cấu hình của PHP 5. Hãy bật cửa sổ của chương trình Apache Service Monitor và bạn sẽ thấy như hình 4.





Bây giờ bạn đã có thể lập trình với PHP 5. Chúc các bạn thành công.

## Bộ lặp (Iterator)

Các bộ lặp là tính năng hoàn toàn mới trong PHP 5.0. Chúng cho phép bạn sử dụng vòng lặp for-each để lặp vòng qua các kiểu dữ liệu khác nhau: danh sách thư mục, tập kết quả CSDL và cả tài liệu XML. SPL - Thư viện PHP chuẩn - là tập các bộ lặp cung cấp chức năng này cùng với các chức năng lọc, giới hạn, ghi bộ nhớ đệm và thay đổi kết quả bộ lặp.

Bộ lặp là cách làm thủ công rất hay để loại bỏ tất cả các đoạn mã gây nhiễu trong mã nguồn của bạn.

**Ví dụ**, bộ lặp DirectoryIterator biến việc lặp qua các thư mục từ cách viết:

```
$dir = opendir($path);

while (false !== ($file = readdir($dir))) {

    print '$file\n';

}

closedir($dir);
```

thành cách viết:

```
foreach (new DirectoryIterator($path) as $file) {

    print '$file\n';

}
```

Như vậy là không có con trỏ tham chiếu thư mục nào nằm lộn xộn vào đó, cũng như không có điều kiện so sánh thiếu trong sáng nào cần phải kiểm tra bên trong vòng lặp.

Ngoài các điểm nổi bật trên, PHP 5.0 còn cung cấp:

- **Cải tiến dòng (streams), bộ bao (wrapper) và bộ lọc (filter).**

Dòng là khái niệm được đưa ra trong PHP 4.3, nhưng trước PHP 5.0 nó là phần ít được dùng đến. Những cải tiến được đưa vào từ bản Beta 3 của PHP 5.0 cho phép đặt giao diện file vào việc đọc và viết dữ liệu bằng cách sử dụng các đối tượng phụ thuộc theo giao thức, mà thường được nhắc đến với cái tên bộ bao. Dòng cũng cho phép thay đổi việc truyền dòng dữ liệu qua chúng bằng cách đính kèm vào đó các bộ lọc.

- **Nội soi mã bằng cách sử dụng các lớp Reflection.**

Tập hợp các lớp này cho phép bạn kiểm tra các lớp, các phương thức, các tham số và những thứ khác để thăm dò các thuộc tính của đối tượng. Với chúng, bạn có thể dễ dàng tạo ra các bộ thăm dò lớp, các bộ dò lỗi PHP và các công cụ khác dựa vào việc thu thập các chi tiết về đối tượng và hàm.

- **Sinh mã HTML chuẩn nhờ Tidy.**

Mô đun mở rộng Tidy giúp bạn dễ dàng nắm chắc việc sinh mã HTML và XHTML hợp lệ. Bộ phân tách thông minh của nó thậm chí còn biến các trang viết mã tùy tiện nhất trở thành các trang tuân thủ chặt chẽ các đặc tả mới nhất của W3C.

- **Xử lý dòng lệnh ưu việt hơn**

Phiên bản dòng lệnh PHP5 giờ đây cho phép xử lý từng dòng riêng lẻ, giống như Perl và awk. Bạn có thể xác định mã cần chạy ở đầu, ở giữa và ở cuối của từng dòng trong một file.ý

**Phạm Công Định**  
**PHPVietnam**  
**pcdinh@yahoo.com**

**Tham khảo:**

- Những thay đổi trong PHP 5: <http://www.php.net/ChangeLog-5.php>
- Sách PHP5 and MySQL Bible của nhà xuất bản Wiley năm 2004.
- Sách PHP and MySQL for Dummies 2nd Edition 2004 của nhà xuất bản Wiley

*Theo PCWorld*

## **SALT: Ngôn ngữ hỗ trợ tiếng nói trên web**

13/9/2004 17h:51

Việc sử dụng điện thoại để truy cập các dịch vụ tự động như kiểm tra tài khoản ngân hàng hay lịch bay không phải là điều mới mẻ. Những hệ thống như vậy cho phép người gọi trả lời các câu hỏi hay chọn các tùy chọn, và hệ thống sẽ đáp trả bằng các câu nói thâu âm sẵn hay dùng kỹ thuật tổng hợp tiếng nói.

Các hệ thống nhận dạng tiếng nói tương tác (IVR - Interactive Voice Recognition) đã xuất hiện nhiều năm nay và là trải nghiệm ban đầu về công nghệ tiếng nói đối với nhiều người.

Sau khi World Wide Web phát triển vào những năm 1990 và các công nghệ web trở nên chuẩn hóa và phổ biến, các nhà phát triển công nghệ tiếng nói bắt đầu tìm kiếm phương thức kết hợp tiếng nói với web. Tất nhiên, phương thức đầu tiên là phối hợp các công nghệ điện thoại/IVR chuyên biệt trước đây với năng lực của cơ sở hạ tầng máy chủ web. Và người ta nhanh chóng nhận ra nhu cầu về một ngôn ngữ chuẩn, chuyên biệt dựa trên XML. Ngôn ngữ này cho phép định nghĩa ngữ cảnh và luận lý điều khiển ứng dụng IVR trong môi trường web. Ngôn ngữ VoiceXML được VoiceXML Forum (<http://www.voicexml.org>) phát triển nhằm đáp ứng nhu cầu này. VoiceXML Forum được thành lập vào năm 1999 bao gồm các công ty AT&T, IBM, Lucent và Motorola, hiện nay số công ty thành viên đã lên đến hàng trăm. Việc đưa ra VoiceXML không chỉ tạo nên các giải pháp IVR mở và linh hoạt hơn, nó còn cho phép truy cập đến ứng dụng web bằng tiếng nói.

Song song với những phát triển trong việc tích hợp tiếng nói và web, máy tính cá nhân (PC) ngày càng trở nên đủ mạnh để xử lý các tác vụ cơ bản của công nghệ tiếng nói: nhận dạng tiếng nói (dữ liệu đầu vào) và thể hiện tiếng nói (dữ liệu đầu ra). Đây quả là kỳ công nếu biết rằng vào những năm 1970 người ta phải cần đến 50 máy tính cho hệ thống HAPPY của trường

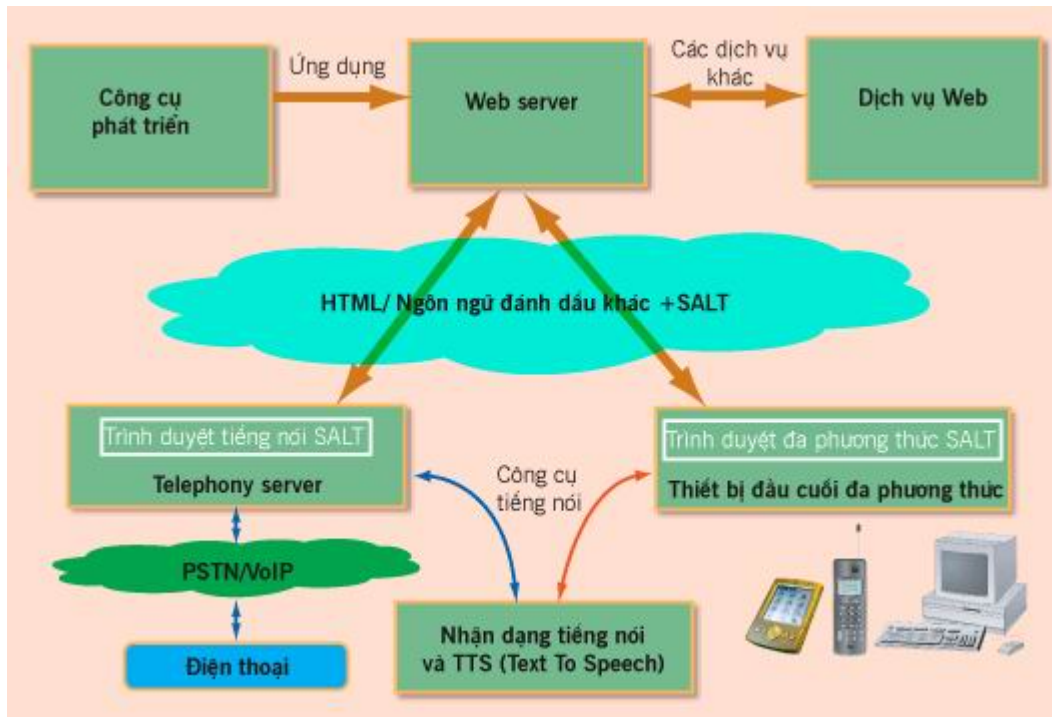
đại học Carnegie Mellon để thực hiện nhận dạng câu nói liên tục (với tốc độ đàm thoại tự nhiên).

Nhờ sự phát triển của PC cũng như sự phát triển của chính các công nghệ tiếng nói, người dùng máy tính bắt đầu có cơ hội trải nghiệm công nghệ tiếng nói trên chính PC của mình. Ngày nay, người dùng máy tính bình thường có cơ hội sử dụng các sản phẩm tiếng nói từ nhiều nhà sản xuất, hỗ trợ nhiều công việc, từ điều khiển máy tính thông qua lệnh nói đến đọc văn bản để nhập trực tiếp vào trình soạn thảo văn bản.

Các nhà phát triển phần mềm cũng được hưởng lợi việc kết hợp công nghệ tiếng nói với PC. Hiện có nhiều lựa chọn để kết hợp công nghệ tiếng nói vào ứng dụng, từ các bộ công cụ phát triển SDK của các hãng phần mềm hệ thống như Speech API (SAPI) của Microsoft và PlainTalk của Apple, đến các công cụ của các hãng phần mềm độc lập như SDK sinh trắc và tiếng nói của NuVoice.

Với sự gia tăng năng lực và khả năng xử lý tiếng nói của PC, một khái niệm mới trong việc kết hợp tiếng nói bắt đầu nổi lên - ứng dụng 'multimodal' (đa phương thức), có khả năng hỗ trợ giao tiếp người dùng phía client bằng tiếng nói, ngoài các phương thức nhập/xuất thông thường là bàn phím, chuột và màn hình. Vì trình duyệt web ngày càng trở nên phổ biến và có khả năng thể hiện giao diện người dùng phong phú, rõ ràng việc phát triển ứng dụng đa phương thức yêu cầu kết hợp chặt chẽ hơn công nghệ tiếng nói với dữ liệu và mô hình thực thi trên web, như các ngôn ngữ kịch bản theo sự kiện, DOM (Document Object Model) và HTML.

Mặc dù VoiceXML đã được mở rộng để cung cấp mức kết hợp này, nhưng một số người trong ngành công nghiệp cho rằng cần có một mô hình khác biệt từ nền tảng. Và điều này dẫn đến việc phát triển SALT (Speech Application Language Tags) của tổ chức SAL Forum ([www.saltforum.org](http://www.saltforum.org)), được thành lập vào năm 2002 với các công ty Microsoft, Intel, Cisco và Philips, hiện nay số công ty thành viên đã tăng hơn 70. Mục tiêu của SALT là kết hợp các công nghệ tiếng nói vào nhiều loại thiết bị điện toán, từ PC đến PDA, thông qua việc hỗ trợ các ứng dụng đa phương thức. Để đạt được mục tiêu này, SALT sử dụng phương thức khác với VoiceXML.



### Khác biệt giữa VoiceXML và SALT

VoiceXML ban đầu được thiết kế nhằm cung cấp môi trường bao quát cho việc xây dựng ứng dụng tiếng nói. Nó đưa ra các thành phần định nghĩa dữ liệu (form và các vùng trong form), lộ trình thực thi và cung cấp môi trường thực thi để dịch lệnh VoiceXML lúc thực thi. Nói chung, việc kết hợp web/tiếng nói được thực hiện thông qua việc tương tác giữa máy chủ web và máy chủ VoiceXML.

Ví dụ, trong một ứng dụng VoiceXML tiêu biểu, điện thoại thực hiện chức năng như công cụ duyệt web, dữ liệu vào là lời nói của người dùng được gửi đến máy chủ VoiceXML để dịch lời nói dựa trên tài liệu lệnh bao gồm các thành phần VoiceXML. Lệnh trong tài liệu có thể chỉ thị cho máy chủ VoiceXML kết nối tới một địa chỉ URL trên máy chủ web, ở đó kịch bản web (ví dụ, JavaScript) tương tác với ứng dụng/website và đáp trả lệnh VoiceXML tương ứng. Máy chủ VoiceXML nhận thông tin lệnh từ máy chủ web, dịch nghĩa nó, và gửi thông tin lời nói đầu ra thích hợp đến người dùng thông qua điện thoại.

Ngược lại, SALT cung cấp môi trường tối thiểu cho việc xây dựng ứng dụng tiếng nói. Nó giới hạn ở việc định nghĩa dữ liệu và hành vi chuyên biệt theo giao tiếp tiếng nói, như là lắng nghe đầu vào và xác định văn phạm dùng để dịch nghĩa thông tin đầu vào. Tất cả dữ liệu khác, như định nghĩa form và thành phần của form, giao cho ngôn ngữ đánh dấu mà SALT nhúng trong đó, như HTML.

#### Ví dụ 1

```
<salt:prompt id='promptWelcome' oncomplete='listenTag.Start()'>
  Welcome to a SALT multi-modal sample application.
</salt:prompt>
```

Để cung cấp mức kết hợp này, các thẻ lệnh SALT được xây dựng theo dạng các thành phần XML có chức năng giống như phần mở rộng của các ngôn ngữ đánh dấu hiện nay như HTML, XHTML và WML. Hơn nữa, các thành phần SALT còn dùng giao tiếp DOM để làm việc như là thành viên có toàn quyền trong mô hình dữ liệu của trang chủ. Điều này có nghĩa là chúng sở hữu các phương thức, thuộc tính và sự kiện có thể được truy xuất trực tiếp từ các ngôn ngữ kịch bản tương thích ECMA như JavaScript theo cùng cách thức như những thành phần khác trong trang web tương thích DOM.

Nói tóm lại, SALT được thiết kế đặc biệt để thích hợp với mô hình dữ liệu và môi trường thực thi được các nhà phát triển web sử dụng hiện nay. Toàn bộ đặc tả kỹ thuật của SALT có ở địa chỉ <http://saltforum.org/devforum/default.asp>.

## Ứng dụng SALT

Chúng ta hãy xem xét ứng dụng đa phương thức cơ bản dùng 4 thành phần thẻ lệnh quan trọng nhất của SALT.

Trước hết, bạn cần có một phương thức nhắc người dùng đưa thông tin vào. Thẻ lệnh SALT `<prompt>` được dùng để xác định nội dung phát âm như lời chào, hướng dẫn hay xác nhận. Nguồn nội dung có thể là văn bản được phát âm hay tập tin âm thanh. Với tập tin âm thanh thì công việc chỉ đơn giản là phát lại tập tin này, nhưng với văn bản thì phải cần đến công cụ TTS (text-to-speech) để chuyển thành tiếng nói.

Ví dụ 1 minh họa việc sử dụng thành phần `<prompt>` để phát âm văn bản 'Welcome to a SALT multimodal application'. Sự kiện `oncomplete` kích hoạt thành phần `<listen>` khi việc phát thông báo hoàn tất.

### Ví dụ 2

```
<salt:listen id='listenTag'>
  <salt:grammar src='TagGrammar.xml' />
  <salt:bind value='//Tag' targetElement='selectTag' />
</salt:listen>
```

Thành phần `<listen>` cung cấp điều khiển nhận dạng tiếng nói, chuyển tiếng nói thành văn bản và xử lý kết quả nhận dạng. Ví dụ 2 minh họa việc sử dụng thành phần `<listen>`. Trong ví dụ này, có 2 thành phần con: Thành phần `<grammar>` xác định từ khóa nhận dạng, thành phần `<bind>` chỉ đến thành phần khác của trang web, ví dụ như một thẻ lệnh HTML.

Thành phần `<grammar>` xác định chính xác cái mà chương trình có thể nhận dạng và có thể xem như là danh sách từ điển. Trình nhận dạng tiếng nói dịch tín hiệu số thể hiện câu nói thành các âm và so khớp với thể hiện âm của các từ trong danh sách được phân định theo văn phạm. Nếu tìm thấy từ trùng khớp thì kết quả được trả về. Văn phạm có thể được xác định ngay trong dòng lệnh hoặc có thể nằm ở một tập tin riêng và được chỉ đến bởi thuộc tính `src` của thành phần `<grammar>`. Trong ví dụ 2, văn phạm chứa trong tập tin ngoài 'TagGrammar.xml'; mã nguồn của tập tin này được trình bày trong ví dụ 3 định nghĩa nhận dạng 3 từ 'prompt', 'listen', 'grammar'.

Với 4 thành phần SALT cơ bản trên, chúng ta có thể ráp thành ứng dụng mẫu như ở ví dụ 4. Điều lưu ý đầu tiên là trang web này phải tham chiếu đến định nghĩa SALT ở dòng đầu.

```
<HTML xmlns:salt='http://www.saltforum.org/2002/SALT'>
```

Khi trang nạp vào trình duyệt, dòng:

```
<body onload='promptWelcome.Start()'>
```

bắt đầu phát thông báo được chỉ định là promptWelcome, công cụ TTS sẽ phát âm câu 'Welcome to a SALT multimodal sample. You may select...'. Sau khi thông báo kết thúc, sự kiện oncomplete của thành phần promptWelcome kích hoạt thành phần <listen> được chỉ định là listenTag. Việc này sẽ kích hoạt trình nhận dạng lắng nghe tiếng nói đầu vào. Khi phát hiện tín hiệu đầu vào, trình nhận dạng sẽ cố gắng nhận dạng dựa trên văn phạm đã được nạp bởi thành phần con <grammar>.

```
<salt:grammar src='TagGrammar.xml' />
```

Nếu nhận dạng thành công, thành phần con <bind> sẽ được thực thi.

```
<salt:bind value='//Tag' targetElement='selectTag' />
```

Lệnh này gán giá trị của nút <tag> ở dữ liệu XML đầu ra của trình nhận dạng cho thành phần selectTag, được định nghĩa ở dạng HTML trong ví dụ này là danh sách xổ xuống. Việc này dẫn đến tùy chọn tương ứng trong điều khiển HTML SELECT được chọn. Sau khi bind được thực thi, sự kiện nhận dạng thành công được kích hoạt với onreco, nó gọi hàm JavaScript ProcessInput(). Lệnh trong ProcessInput() khi đó sẽ hiển thị khai báo văn bản đang được hiển thị trong điều khiển HTML SELECT. Lưu ý là ProcessInput() cũng được gọi bởi sự kiện onchange của điều khiển SELECT được kích hoạt khi người dùng chọn một tùy chọn bằng chuột hay bàn phím.

Ứng dụng đa phương thức đơn giản này minh họa cách thức SALT hỗ trợ tiếng nói điều khiển các thành phần trang web trong khi vẫn cho phép người dùng tương tác với các thành phần đó bằng chuột và bàn phím theo cách truyền thống. Ở khía cạnh lập trình, ứng dụng này cũng minh họa việc tương tác thông suốt giữa các thành phần SALT với mô hình dữ liệu và môi trường thực thi của trang web thông thường.

### Ví dụ 3

```
<grammar root='root' xml:lang='en-US' version='1.0'
xmlns='http://www.w3.org/2001/06/grammar'>
  <rule id='root' scope='public'>
    <ruleref uri='#Tag' />
    <tag> $.Tag = $$ </tag>
  </rule>
  <rule id='Tag' scope='public'>
    <one-of>
      <item>prompt<tag> $. _value = 'prompt'</tag></item>
      <item>listen<tag> $. _value = 'listen'</tag></item>
      <item>grammar<tag> $. _value =
'grammar'</tag></item>
      <item>bind<tag> $. _value = 'bind'</tag></item>
    </one-of>
  </rule>
```

</grammar>

### **SALT với công nghệ WEB của Microsoft**

Việc kết hợp chặt chẽ SALT với mô hình dữ liệu và môi trường thực thi phổ biến của web (HTML, DOM, XML và ngôn ngữ kịch bản) đồng thời cũng hỗ trợ việc kết hợp SALT vào các môi trường phát triển thông dụng. Chúng ta hãy xem xét cách thức Microsoft hỗ trợ SALT trong môi trường IIS (Internet Information Server) và IE (Internet Explorer).

Trước hết, trên máy tính client bạn phải cài đặt Microsoft Speech Add-in cho IE. Add-in này cung cấp các thư viện DLL bổ sung cho IE 6.0 và thực hiện việc dịch lệnh SALT trong trình duyệt. Lưu ý là nếu bạn đã có cài Speech Application SDK (SASDK) thì không cần thực hiện bước này.

Bước kế tiếp, tạo một thư mục trên máy chủ IIS và thiết lập làm thư mục gốc của ứng dụng. Trong thư mục mới này, tạo một tập tin chứa mã lệnh ứng dụng SALT (dùng mã lệnh trong phần 'Ứng dụng SALT') và đặt tên với đuôi thích hợp, ví dụ SimpleSALT.slt. Tiếp theo, tạo một tập tin khác chứa văn phạm và đặt tên cho nó, ví dụ TagGrammar.xml.

Cuối cùng, thiết lập kiểu MIME trên máy chủ IIS cho trang chứa lệnh SALT. Ví dụ, nếu tập tin được đặt tên là SimpleSALT.slt thì bạn cần liên kết các tập tin \*.slt với loại SALT MIME. Một cách để làm việc này là dùng công cụ Internet Services Manager, chọn thư mục gốc của ứng dụng hay website, mở trang thuộc tính của nó, chọn mục HTTP Headers, nhấn nút File Type trong phần MIME Map, và tạo một loại mới với phần mở rộng '.slt' và dạng nội dung 'text/salt+html'. Thao tác này sẽ làm cho Microsoft Speech Add-in tự động kích hoạt các đối tượng SALT và thực hiện chức năng SALT cho IE.

Giờ bạn có thể dùng trình duyệt để mở trang web hỗ trợ SALT trong thư mục đã tạo và chạy ứng dụng. Nếu bạn 'view source' trong trình duyệt, bạn sẽ chú ý thấy do việc ánh xạ MIME, đoạn mã tương tự như dưới đây đã được IIS thêm vào trang web của bạn trước khi chuyển nó đến trình duyệt:

```
<object id='saltobject26239220' CLASSID='clsid:DCF68E5B-84A1-4047-98A4-0A72276D19CC' VIEWASTEXT WIDTH=0 HEIGHT=0></object>
```

```
<?import namespace='salt' implementation='#salobject26239220' />
```



```
<?import namespace='mm' implementation='#salobject26239220' />
```

IIS đã chèn các dòng này trực tiếp vào trình IE để báo cho Speech Add-in biết xử lý các thành phần SALT.

#### **Ví dụ 4**

```

<HTML xmlns:salt='http://www.saltforum.org/2002/SALT'>
<HEAD>
<TITLE>A SALT Multi-modal Example</TITLE>
<SCRIPT language='JavaScript'>
function ProcessInput() {
    promptWelcome.Stop();
    listenTag.Stop();
    switch(selectTag.value) {
        case 'prompt': divDefinition.innerHTML =
            'The prompt element ' +
            'is used to specify the content of audio ' +
            'output, either as inline or referenced text, ' +
            'variable values, or links to audio files.';
            break;
        case 'listen': divDefinition.innerHTML =
            'The listen element ' +
            'is used for recognition and/or recording, ' +
            'and contains one or more grammars and ' +
            'optionally a set of bind elements to inspect ' +
            'and copy input.';
            break;
        case 'grammar': divDefinition.innerHTML =
            'The grammar element ' +
            'is used to specify possible user inputs ' +
            'with rules identified either inline or ' +
            'by reference.';
            break;
        case 'bind': divDefinition.innerHTML =
            'The bind element ' +
            'is used to bind values from spoken input ' +
            'into the page, or to call methods on ' +
            'page elements.';
            break;
        default: divDefinition.innerHTML = "";
    }
}
</SCRIPT>
</HEAD>
<body onload='promptWelcome.Start()'>
Select a SALT tag to view its definition:<P>
<SELECT id='selectTag' onchange='ProcessInput()'>
    <OPTION value='prompt'>prompt</OPTION>
    <OPTION value='listen'>listen</OPTION>
    <OPTION value='grammar'>grammar</OPTION>
    <OPTION value='bind'>bind</OPTION>
    <OPTION SELECTED value=''>-- Select a Tag ---</OPTION>
</SELECT>
<P>Definition:<P>
<DIV id='divDefinition'> </DIV>
<salt:prompt id='promptWelcome' oncomplete='listenTag.Start()'>
Welcome to a SALT multi-modal sample. You may select a SALT tag from the

```

dropdown list either by speaking or by using a mouse.

```
</salt:prompt>
```

```
<salt:listen id='listenTag' onreco='ProcessInput()>
```

```
  <salt:grammar src='TagGrammar.xml' />
```

```
  <salt:bind value='//Tag' targetElement='selectTag' />
```

```
</salt:listen>
```

```
</body>
```

```
</HTML>
```

## Lời kết

Tuy bài viết này chú trọng đến phía client của ứng dụng đa phương thức, lưu ý rằng SALT cũng hỗ trợ ứng dụng thoại thuần túy, không có giao diện đồ họa. Trong trường hợp này, điện thoại thực hiện chức năng thiết bị đầu cuối, bộ thông dịch SALT chỉ hỗ trợ tiếng nói kết hợp với máy chủ tiếng nói và điện thoại theo cách thức tương tự mô hình VoiceXML truyền thống.

Ngoài công nghệ Microsoft được dùng để triển khai ứng dụng mẫu trong bài viết này, còn có một số sản phẩm của các hãng khác hỗ trợ SALT (xem trên website của SALT Forum), và có cả giải pháp nguồn mở OpenSALT (<http://hsp.speech.cs.cmu.edu/salt/>) của trường đại học Carnegie Mellon, thành viên của SALT Forum. OpenSALT đã đưa ra trình duyệt nguồn mở tương thích SALT 1.0 dựa trên trình duyệt nguồn mở Mozilla và sử dụng phần mềm nguồn mở tổng hợp tiếng nói Festival và nhận dạng Sphinx

*Theo PCWorld*

## SALT: Ngôn ngữ hỗ trợ tiếng nói trên web

13/9/2004 17h:51

**Việc sử dụng điện thoại để truy cập các dịch vụ tự động như kiểm tra tài khoản ngân hàng hay lịch bay không phải là điều mới mẻ. Những hệ thống như vậy cho phép người gọi trả lời các câu hỏi hay chọn các tùy chọn, và hệ thống sẽ đáp trả bằng các câu nói thâu âm sẵn hay dùng kỹ thuật tổng hợp tiếng nói.**

Các hệ thống nhận dạng tiếng nói tương tác (IVR - Interactive Voice Recognition) đã xuất hiện nhiều năm nay và là trải nghiệm ban đầu về công nghệ tiếng nói đối với nhiều người.

Sau khi World Wide Web phát triển vào những năm 1990 và các công nghệ web trở nên chuẩn hóa và phổ biến, các nhà phát triển công nghệ tiếng nói bắt đầu tìm kiếm phương thức kết hợp tiếng nói với web. Tất nhiên, phương thức đầu tiên là phối hợp các công nghệ điện thoại/IVR chuyên biệt trước đây với năng lực của cơ sở hạ tầng máy chủ web. Và người ta nhanh chóng nhận ra nhu cầu về một ngôn ngữ chuẩn, chuyên biệt dựa trên XML. Ngôn ngữ này cho phép định nghĩa ngữ cảnh và luận lý điều khiển ứng dụng IVR trong môi trường web. Ngôn ngữ VoiceXML được VoiceXML Forum (<http://www.voicexml.org>) phát triển nhằm đáp ứng nhu cầu này. VoiceXML Forum được thành lập vào năm 1999 bao gồm các công ty AT&T, IBM, Lucent và Motorola, hiện nay số công ty thành viên đã lên đến hàng trăm. Việc đưa ra VoiceXML không chỉ tạo nên các giải pháp IVR mở và linh hoạt hơn, nó còn cho phép truy cập đến ứng dụng web bằng tiếng nói.

Song song với những phát triển trong việc tích hợp tiếng nói và web, máy tính cá nhân (PC) ngày càng trở nên đủ mạnh để xử lý các tác vụ cơ bản của công nghệ tiếng nói: nhận dạng tiếng nói (dữ liệu đầu vào) và thể hiện tiếng nói (dữ liệu đầu ra). Đây quả là kỳ công nếu biết rằng vào những năm 1970 người ta phải cần đến 50 máy tính cho hệ thống HAPPY của trường đại học Carnegie Mellon để thực hiện nhận dạng câu nói liên tục (với tốc độ đàm thoại tự nhiên).

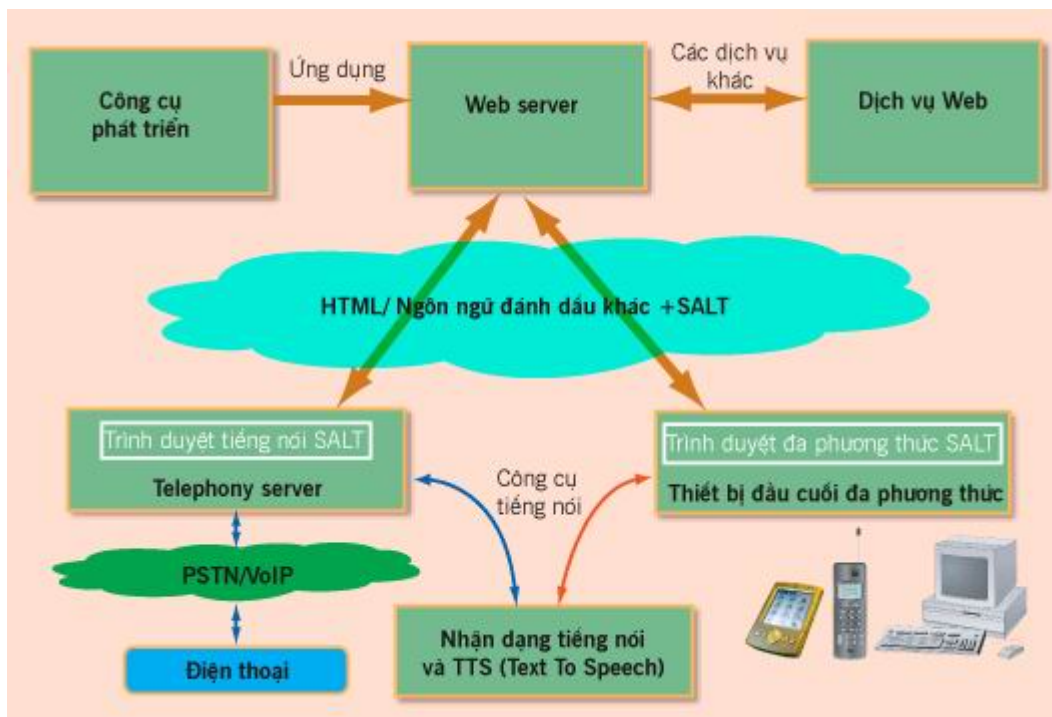
Nhờ sự phát triển của PC cũng như sự phát triển của chính các công nghệ tiếng nói, người dùng máy tính bắt đầu có cơ hội trải nghiệm công nghệ tiếng nói trên chính PC của mình.

Ngày nay, người dùng máy tính bình thường có cơ hội sử dụng các sản phẩm tiếng nói từ nhiều nhà sản xuất, hỗ trợ nhiều công việc, từ điều khiển máy tính thông qua lệnh nói đến đọc văn bản để nhập trực tiếp vào trình soạn thảo văn bản.

Các nhà phát triển phần mềm cũng được hưởng lợi việc kết hợp công nghệ tiếng nói với PC. Hiện có nhiều lựa chọn để kết hợp công nghệ tiếng nói vào ứng dụng, từ các bộ công cụ phát triển SDK của các hãng phần mềm hệ thống như Speech API (SAPI) của Microsoft và PlainTalk của Apple, đến các công cụ của các hãng phần mềm độc lập như SDK sinh trắc và tiếng nói của NeuVoice.

Với sự gia tăng năng lực và khả năng xử lý tiếng nói của PC, một khái niệm mới trong việc kết hợp tiếng nói bắt đầu nổi lên - ứng dụng 'multimodal' (đa phương thức), có khả năng hỗ trợ giao tiếp người dùng phía client bằng tiếng nói, ngoài các phương thức nhập/xuất thông thường là bàn phím, chuột và màn hình. Vì trình duyệt web ngày càng trở nên phổ biến và có khả năng thể hiện giao diện người dùng phong phú, rõ ràng việc phát triển ứng dụng đa phương thức yêu cầu kết hợp chặt chẽ hơn công nghệ tiếng nói với dữ liệu và mô hình thực thi trên web, như các ngôn ngữ kịch bản theo sự kiện, DOM (Document Object Model) và HTML.

Mặc dù VoiceXML đã được mở rộng để cung cấp mức kết hợp này, nhưng một số người trong ngành công nghiệp cho rằng cần có một mô hình khác biệt từ nền tảng. Và điều này dẫn đến việc phát triển SALT (Speech Application Language Tags) của tổ chức SAL Forum ([www.saltforum.org](http://www.saltforum.org)), được thành lập vào năm 2002 với các công ty Microsoft, Intel, Cisco và Philips, hiện nay số công ty thành viên đã tăng hơn 70. Mục tiêu của SALT là kết hợp các công nghệ tiếng nói vào nhiều loại thiết bị điện toán, từ PC đến PDA, thông qua việc hỗ trợ các ứng dụng đa phương thức. Để đạt được mục tiêu này, SALT sử dụng phương thức khác với VoiceXML.



## Khác biệt giữa VoiceXML và SALT

VoiceXML ban đầu được thiết kế nhằm cung cấp môi trường bao quát cho việc xây dựng ứng dụng tiếng nói. Nó đưa ra các thành phần định nghĩa dữ liệu (form và các vùng trong form), lộ trình thực thi và cung cấp môi trường thực thi để dịch lệnh VoiceXML lúc thực thi. Nói chung, việc kết hợp web/tiếng nói được thực hiện thông qua việc tương tác giữa máy chủ web và máy chủ VoiceXML.

Ví dụ, trong một ứng dụng VoiceXML tiêu biểu, điện thoại thực hiện chức năng như công cụ duyệt web, dữ liệu vào là lời nói của người dùng được gửi đến máy chủ VoiceXML để dịch lời nói dựa trên tài liệu lệnh bao gồm các thành phần VoiceXML. Lệnh trong tài liệu có thể chỉ thị cho máy chủ VoiceXML kết nối tới một địa chỉ URL trên máy chủ web, ở đó kịch bản web (ví dụ, JavaScript) tương tác với ứng dụng/website và đáp trả lệnh VoiceXML tương ứng. Máy chủ VoiceXML nhận thông tin lệnh từ máy chủ web, dịch nghĩa nó, và gửi thông tin lời nói đầu ra thích hợp đến người dùng thông qua điện thoại.

Ngược lại, SALT cung cấp môi trường tối thiểu cho việc xây dựng ứng dụng tiếng nói. Nó giới hạn ở việc định nghĩa dữ liệu và hành vi chuyên biệt theo giao tiếp tiếng nói, như là lắng nghe đầu vào và xác định văn phạm dùng để dịch nghĩa thông tin đầu vào. Tất cả dữ liệu khác, như định nghĩa form và thành phần của form, giao cho ngôn ngữ đánh dấu mà SALT nhúng trong đó, như HTML.

### Ví dụ 1

```
<salt:prompt id='promptWelcome' oncomplete='listenTag.Start()>  
    Welcome to a SALT multi-modal sample application.  
</salt:prompt>
```

Để cung cấp mức kết hợp này, các thẻ lệnh SALT được xây dựng theo dạng các thành phần XML có chức năng giống như phần mở rộng của các ngôn ngữ đánh dấu hiện nay như HTML, XHTML và WML. Hơn nữa, các thành phần SALT còn dùng giao tiếp DOM để làm việc như là thành viên có toàn quyền trong mô hình dữ liệu của trang chủ. Điều này có nghĩa là chúng sở hữu các phương thức, thuộc tính và sự kiện có thể được truy xuất trực tiếp từ các ngôn ngữ kịch bản tương thích ECMA như JavaScript theo cùng cách thức như những thành phần khác trong trang web tương thích DOM.

Nói tóm lại, SALT được thiết kế đặc biệt để thích hợp với mô hình dữ liệu và môi trường thực thi được các nhà phát triển web sử dụng hiện nay. Toàn bộ đặc tả kỹ thuật của SALT có ở địa chỉ <http://saltforum.org/devforum/default.asp>.

## Ứng dụng SALT

Chúng ta hãy xem xét ứng dụng đa phương thức cơ bản dùng 4 thành phần thẻ lệnh quan trọng nhất của SALT.

Trước hết, bạn cần có một phương thức nhắc người dùng đưa thông tin vào. Thẻ lệnh SALT `<prompt>` được dùng để xác định nội dung phát âm như lời chào, hướng dẫn hay xác nhận. Nguồn nội dung có thể là văn bản được phát âm hay tập tin âm thanh. Với tập tin âm thanh thì công việc chỉ đơn giản là phát lại tập tin này, nhưng với văn bản thì phải cần đến công cụ TTS (text-to-speech) để chuyển thành tiếng nói.

Ví dụ 1 minh họa việc sử dụng thành phần <prompt> để phát âm văn bản 'Welcome to a SALT multimodal application'. Sự kiện oncomplete kích hoạt thành phần <listen> khi việc phát thông báo hoàn tất.

### Ví dụ 2

```
<salt:listen id='listenTag'>
  <salt:grammar src='TagGrammar.xml' />
  <salt:bind value='//Tag' targetElement='selectTag' />
</salt:listen>
```

Thành phần <listen> cung cấp điều khiển nhận dạng tiếng nói, chuyển tiếng nói thành văn bản và xử lý kết quả nhận dạng. Ví dụ 2 minh họa việc sử dụng thành phần <listen>. Trong ví dụ này, có 2 thành phần con: Thành phần <grammar> xác định từ khóa nhận dạng, thành phần <bind> chỉ đến thành phần khác của trang web, ví dụ như một thẻ lệnh HTML.

Thành phần <grammar> xác định chính xác cái mà chương trình có thể nhận dạng và có thể xem như là danh sách từ điển. Trình nhận dạng tiếng nói dịch tín hiệu số thể hiện câu nói thành các âm và so khớp với thể hiện âm của các từ trong danh sách được phân định theo văn phạm. Nếu tìm thấy từ trùng khớp thì kết quả được trả về. Văn phạm có thể được xác định ngay trong dòng lệnh hoặc có thể nằm ở một tập tin riêng và được chỉ đến bởi thuộc tính src của thành phần <grammar>. Trong ví dụ 2, văn phạm chứa trong tập tin ngoài 'TagGrammar.xml'; mã nguồn của tập tin này được trình bày trong ví dụ 3 định nghĩa nhận dạng 3 từ 'prompt', 'listen', 'grammar'.

Với 4 thành phần SALT cơ bản trên, chúng ta có thể ráp thành ứng dụng mẫu như ở ví dụ 4. Điều lưu ý đầu tiên là trang web này phải tham chiếu đến định nghĩa SALT ở dòng đầu.

```
<HTML xmlns:salt= 'http://www.saltforum.org/2002/SALT'>
```

Khi trang nạp vào trình duyệt, dòng:

```
<body onload='promptWelcome.Start()'>
```

bắt đầu phát thông báo được chỉ định là promptWelcome, công cụ TTS sẽ phát âm câu 'Welcome to a SALT multimodal sample. You may select...'. Sau khi thông báo kết thúc, sự kiện oncomplete của thành phần promptWelcome kích hoạt thành phần <listen> được chỉ định là listenTag. Việc này sẽ kích hoạt trình nhận dạng lắng nghe tiếng nói đầu vào. Khi phát hiện tín hiệu đầu vào, trình nhận dạng sẽ cố gắng nhận dạng dựa trên văn phạm đã được nạp bởi thành phần con <grammar>.

```
<salt:grammar src='TagGrammar.xml' />
```

Nếu nhận dạng thành công, thành phần con <bind> sẽ được thực thi.

```
<salt:bind value='//Tag' targetElement='selectTag' />
```

Lệnh này gán giá trị của nút <tag> ở dữ liệu XML đầu ra của trình nhận dạng cho thành phần selectTag, được định nghĩa ở dạng HTML trong ví dụ này là danh sách xổ xuống. Việc này dẫn đến tùy chọn tương ứng trong điều khiển HTML SELECT được chọn. Sau khi bind được thực thi, sự kiện nhận dạng thành công được kích hoạt với onreco, nó gọi hàm JavaScript

ProcessInput(). Lệnh trong ProcessInput() khi đó sẽ hiển thị khai báo văn bản đang được hiển thị trong điều khiển HTML SELECT. Lưu ý là ProcessInput() cũng được gọi bởi sự kiện onchange của điều khiển SELECT được kích hoạt khi người dùng chọn một tùy chọn bằng chuột hay bàn phím.

Ứng dụng đa phương thức đơn giản này minh họa cách thức SALT hỗ trợ tiếng nói điều khiển các thành phần trang web trong khi vẫn cho phép người dùng tương tác với các thành phần đó bằng chuột và bàn phím theo cách truyền thống. Ở khía cạnh lập trình, ứng dụng này cũng minh họa việc tương tác thông suốt giữa các thành phần SALT với mô hình dữ liệu và môi trường thực thi của trang web thông thường.

### Ví dụ 3

```
<grammar root='root' xml:lang='en-US' version='1.0'
xmlns='http://www.w3.org/2001/06/grammar'>
  <rule id='root' scope='public'>
    <ruleref uri='#Tag'/>
    <tag> $.Tag = $$ </tag>
  </rule>
  <rule id='Tag' scope='public'>
    <one-of>
      <item>prompt<tag> $. _value = 'prompt'</tag></item>
      <item>listen<tag> $. _value = 'listen'</tag></item>
      <item>grammar<tag> $. _value =
'grammar'</tag></item>
      <item>bind<tag> $. _value = 'bind'</tag></item>
    </one-of>
  </rule>
</grammar>
```

## SALT với công nghệ WEB của Microsoft

Việc kết hợp chặt chẽ SALT với mô hình dữ liệu và môi trường thực thi phổ biến của web (HTML, DOM, XML và ngôn ngữ kịch bản) đồng thời cũng hỗ trợ việc kết hợp SALT vào các môi trường phát triển thông dụng. Chúng ta hãy xem xét cách thức Microsoft hỗ trợ SALT trong môi trường IIS (Internet Information Server) và IE (Internet Explorer).

Trước hết, trên máy tính client bạn phải cài đặt Microsoft Speech Add-in cho IE. Add-in này cung cấp các thư viện DLL bổ sung cho IE 6.0 và thực hiện việc dịch lệnh SALT trong trình duyệt. Lưu ý là nếu bạn đã có cài Speech Application SDK (SASDK) thì không cần thực hiện bước này.

Bước kế tiếp, tạo một thư mục trên máy chủ IIS và thiết lập làm thư mục gốc của ứng dụng. Trong thư mục mới này, tạo một tập tin chứa mã lệnh ứng dụng SALT (dùng mã lệnh trong phần 'Ứng dụng SALT') và đặt tên với đuôi thích hợp, ví dụ SimpleSALT.slt. Tiếp theo, tạo một tập tin khác chứa văn phạm và đặt tên cho nó, ví dụ TagGrammar.xml.

Cuối cùng, thiết lập kiểu MIME trên máy chủ IIS cho trang chứa lệnh SALT. Ví dụ, nếu tập tin được đặt tên là SimpleSALT.slt thì bạn cần liên kết các tập tin \*.slt với loại SALT MIME. Một cách để làm việc này là dùng công cụ Internet Services Manager, chọn thư mục gốc của ứng dụng hay website, mở trang thuộc tính của nó, chọn mục HTTP Headers, nhấn nút File



Type trong phần MIME Map, và tạo một loại mới với phần mở rộng '.slt' và dạng nội dung 'text/salt+html'. Thao tác này sẽ làm cho Microsoft Speech Add-in tự động kích hoạt các đối tượng SALT và thực hiện chức năng SALT cho IE.

Giờ bạn có thể dùng trình duyệt để mở trang web hỗ trợ SALT trong thư mục đã tạo và chạy ứng dụng. Nếu bạn 'view source' trong trình duyệt, bạn sẽ chú ý thấy do việc ánh xạ MIME, đoạn mã tương tự như dưới đây đã được IIS thêm vào trang web của bạn trước khi chuyển nó đến trình duyệt:

```
<object id='saltobject26239220' CLASSID='clsid:DCF68E5B-84A1-4047-98A4-0A72276D19CC' VIEWASTEXT WIDTH=0 HEIGHT=0></object>
```

```
<?import namespace='salt' implementation='#saltobject26239220' />
```

```
<?import namespace='mm' implementation='#saltobject26239220' />
```

IIS đã chèn các dòng này trực tiếp vào trình IE để báo cho Speech Add-in biết xử lý các thành phần SALT.

#### **Ví dụ 4**

```

<HTML xmlns:salt='http://www.saltforum.org/2002/SALT'>
<HEAD>
<TITLE>A SALT Multi-modal Example</TITLE>
<SCRIPT language='JavaScript'>
function ProcessInput() {
    promptWelcome.Stop();
    listenTag.Stop();
    switch(selectTag.value) {
        case 'prompt': divDefinition.innerHTML =
            'The prompt element ' +
            'is used to specify the content of audio ' +
            'output, either as inline or referenced text, ' +
            'variable values, or links to audio files.';
            break;
        case 'listen': divDefinition.innerHTML =
            'The listen element ' +
            'is used for recognition and/or recording, ' +
            'and contains one or more grammars and ' +
            'optionally a set of bind elements to inspect ' +
            'and copy input.';
            break;
        case 'grammar': divDefinition.innerHTML =
            'The grammar element ' +
            'is used to specify possible user inputs ' +
            'with rules identified either inline or ' +
            'by reference.';
            break;
        case 'bind': divDefinition.innerHTML =
            'The bind element ' +
            'is used to bind values from spoken input ' +
            'into the page, or to call methods on ' +
            'page elements.';
            break;
        default: divDefinition.innerHTML = "";
    }
}
</SCRIPT>
</HEAD>
<body onload='promptWelcome.Start()'>
Select a SALT tag to view its definition:<P>
<SELECT id='selectTag' onchange='ProcessInput()'>
    <OPTION value='prompt'>prompt</OPTION>
    <OPTION value='listen'>listen</OPTION>
    <OPTION value='grammar'>grammar</OPTION>
    <OPTION value='bind'>bind</OPTION>
    <OPTION SELECTED value=''>-- Select a Tag ---</OPTION>
</SELECT>
<P>Definition:<P>
<DIV id='divDefinition'> </DIV>
<salt:prompt id='promptWelcome' oncomplete='listenTag.Start()'>
Welcome to a SALT multi-modal sample. You may select a SALT tag from the

```

dropdown list either by speaking or by using a mouse.

```
</salt:prompt>
```

```
<salt:listen id='listenTag' onreco='ProcessInput()>
```

```
  <salt:grammar src='TagGrammar.xml' />
```

```
  <salt:bind value='//Tag' targetElement='selectTag' />
```

```
</salt:listen>
```

```
</body>
```

```
</HTML>
```

## Lời kết

Tuy bài viết này chú trọng đến phía client của ứng dụng đa phương thức, lưu ý rằng SALT cũng hỗ trợ ứng dụng thoại thuần túy, không có giao diện đồ họa. Trong trường hợp này, điện thoại thực hiện chức năng thiết bị đầu cuối, bộ thông dịch SALT chỉ hỗ trợ tiếng nói kết hợp với máy chủ tiếng nói và điện thoại theo cách thức tương tự mô hình VoiceXML truyền thống.

Ngoài công nghệ Microsoft được dùng để triển khai ứng dụng mẫu trong bài viết này, còn có một số sản phẩm của các hãng khác hỗ trợ SALT (xem trên website của SALT Forum), và có cả giải pháp nguồn mở OpenSALT (<http://hsp.speech.cs.cmu.edu/salt/>) của trường đại học Carnegie Mellon, thành viên của SALT Forum. OpenSALT đã đưa ra trình duyệt nguồn mở tương thích SALT 1.0 dựa trên trình duyệt nguồn mở Mozilla và sử dụng phần mềm nguồn mở tổng hợp tiếng nói Festival và nhận dạng Sphinx

*Theo PCWorld*

## SALT: Ngôn ngữ hỗ trợ tiếng nói trên web

13/9/2004 17h:51

**Việc sử dụng điện thoại để truy cập các dịch vụ tự động như kiểm tra tài khoản ngân hàng hay lịch bay không phải là điều mới mẻ. Những hệ thống như vậy cho phép người gọi trả lời các câu hỏi hay chọn các tùy chọn, và hệ thống sẽ đáp trả bằng các câu nói thâu âm sẵn hay dùng kỹ thuật tổng hợp tiếng nói.**

Các hệ thống nhận dạng tiếng nói tương tác (IVR - Interactive Voice Recognition) đã xuất hiện nhiều năm nay và là trải nghiệm ban đầu về công nghệ tiếng nói đối với nhiều người.

Sau khi World Wide Web phát triển vào những năm 1990 và các công nghệ web trở nên chuẩn hóa và phổ biến, các nhà phát triển công nghệ tiếng nói bắt đầu tìm kiếm phương thức kết hợp tiếng nói với web. Tất nhiên, phương thức đầu tiên là phối hợp các công nghệ điện thoại/IVR chuyên biệt trước đây với năng lực của cơ sở hạ tầng máy chủ web. Và người ta nhanh chóng nhận ra nhu cầu về một ngôn ngữ chuẩn, chuyên biệt dựa trên XML. Ngôn ngữ này cho phép định nghĩa ngữ cảnh và luận lý điều khiển ứng dụng IVR trong môi trường web. Ngôn ngữ VoiceXML được VoiceXML Forum (<http://www.voicexml.org>) phát triển nhằm đáp ứng nhu cầu này. VoiceXML Forum được thành lập vào năm 1999 bao gồm các công ty AT&T, IBM, Lucent và Motorola, hiện nay số công ty thành viên đã lên đến hàng trăm. Việc đưa ra VoiceXML không chỉ tạo nên các giải pháp IVR mở và linh hoạt hơn, nó còn cho phép truy cập đến ứng dụng web bằng tiếng nói.

Song song với những phát triển trong việc tích hợp tiếng nói và web, máy tính cá nhân (PC) ngày càng trở nên đủ mạnh để xử lý các tác vụ cơ bản của công nghệ tiếng nói: nhận dạng tiếng nói (dữ liệu đầu vào) và thể hiện tiếng nói (dữ liệu đầu ra). Đây quả là kỳ công nếu biết rằng vào những năm 1970 người ta phải cần đến 50 máy tính cho hệ thống HAPPY của trường đại học Carnegie Mellon để thực hiện nhận dạng câu nói liên tục (với tốc độ đàm thoại tự nhiên).

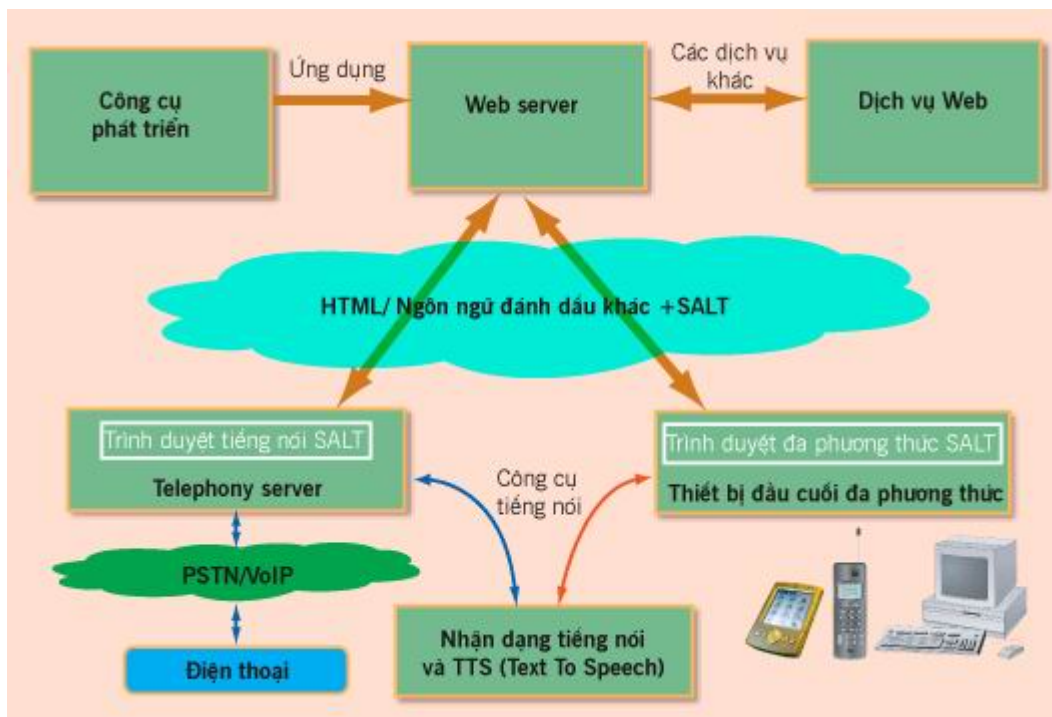
Nhờ sự phát triển của PC cũng như sự phát triển của chính các công nghệ tiếng nói, người dùng máy tính bắt đầu có cơ hội trải nghiệm công nghệ tiếng nói trên chính PC của mình.

Ngày nay, người dùng máy tính bình thường có cơ hội sử dụng các sản phẩm tiếng nói từ nhiều nhà sản xuất, hỗ trợ nhiều công việc, từ điều khiển máy tính thông qua lệnh nói đến đọc văn bản để nhập trực tiếp vào trình soạn thảo văn bản.

Các nhà phát triển phần mềm cũng được hưởng lợi việc kết hợp công nghệ tiếng nói với PC. Hiện có nhiều lựa chọn để kết hợp công nghệ tiếng nói vào ứng dụng, từ các bộ công cụ phát triển SDK của các hãng phần mềm hệ thống như Speech API (SAPI) của Microsoft và PlainTalk của Apple, đến các công cụ của các hãng phần mềm độc lập như SDK sinh trắc và tiếng nói của NeuVoice.

Với sự gia tăng năng lực và khả năng xử lý tiếng nói của PC, một khái niệm mới trong việc kết hợp tiếng nói bắt đầu nổi lên - ứng dụng 'multimodal' (đa phương thức), có khả năng hỗ trợ giao tiếp người dùng phía client bằng tiếng nói, ngoài các phương thức nhập/xuất thông thường là bàn phím, chuột và màn hình. Vì trình duyệt web ngày càng trở nên phổ biến và có khả năng thể hiện giao diện người dùng phong phú, rõ ràng việc phát triển ứng dụng đa phương thức yêu cầu kết hợp chặt chẽ hơn công nghệ tiếng nói với dữ liệu và mô hình thực thi trên web, như các ngôn ngữ kịch bản theo sự kiện, DOM (Document Object Model) và HTML.

Mặc dù VoiceXML đã được mở rộng để cung cấp mức kết hợp này, nhưng một số người trong ngành công nghiệp cho rằng cần có một mô hình khác biệt từ nền tảng. Và điều này dẫn đến việc phát triển SALT (Speech Application Language Tags) của tổ chức SAL Forum ([www.saltforum.org](http://www.saltforum.org)), được thành lập vào năm 2002 với các công ty Microsoft, Intel, Cisco và Philips, hiện nay số công ty thành viên đã tăng hơn 70. Mục tiêu của SALT là kết hợp các công nghệ tiếng nói vào nhiều loại thiết bị điện toán, từ PC đến PDA, thông qua việc hỗ trợ các ứng dụng đa phương thức. Để đạt được mục tiêu này, SALT sử dụng phương thức khác với VoiceXML.



## Khác biệt giữa VoiceXML và SALT

VoiceXML ban đầu được thiết kế nhằm cung cấp môi trường bao quát cho việc xây dựng ứng dụng tiếng nói. Nó đưa ra các thành phần định nghĩa dữ liệu (form và các vùng trong form), lộ trình thực thi và cung cấp môi trường thực thi để dịch lệnh VoiceXML lúc thực thi. Nói chung, việc kết hợp web/tiếng nói được thực hiện thông qua việc tương tác giữa máy chủ web và máy chủ VoiceXML.

Ví dụ, trong một ứng dụng VoiceXML tiêu biểu, điện thoại thực hiện chức năng như công cụ duyệt web, dữ liệu vào là lời nói của người dùng được gửi đến máy chủ VoiceXML để dịch lời nói dựa trên tài liệu lệnh bao gồm các thành phần VoiceXML. Lệnh trong tài liệu có thể chỉ thị cho máy chủ VoiceXML kết nối tới một địa chỉ URL trên máy chủ web, ở đó kịch bản web (ví dụ, JavaScript) tương tác với ứng dụng/website và đáp trả lệnh VoiceXML tương ứng. Máy chủ VoiceXML nhận thông tin lệnh từ máy chủ web, dịch nghĩa nó, và gửi thông tin lời nói đầu ra thích hợp đến người dùng thông qua điện thoại.

Ngược lại, SALT cung cấp môi trường tối thiểu cho việc xây dựng ứng dụng tiếng nói. Nó giới hạn ở việc định nghĩa dữ liệu và hành vi chuyên biệt theo giao tiếp tiếng nói, như là lắng nghe đầu vào và xác định văn phạm dùng để dịch nghĩa thông tin đầu vào. Tất cả dữ liệu khác, như định nghĩa form và thành phần của form, giao cho ngôn ngữ đánh dấu mà SALT nhúng trong đó, như HTML.

### Ví dụ 1

```
<salt:prompt id='promptWelcome' oncomplete='listenTag.Start()>  
    Welcome to a SALT multi-modal sample application.  
</salt:prompt>
```

Để cung cấp mức kết hợp này, các thẻ lệnh SALT được xây dựng theo dạng các thành phần XML có chức năng giống như phần mở rộng của các ngôn ngữ đánh dấu hiện nay như HTML, XHTML và WML. Hơn nữa, các thành phần SALT còn dùng giao tiếp DOM để làm việc như là thành viên có toàn quyền trong mô hình dữ liệu của trang chủ. Điều này có nghĩa là chúng sở hữu các phương thức, thuộc tính và sự kiện có thể được truy xuất trực tiếp từ các ngôn ngữ kịch bản tương thích ECMA như JavaScript theo cùng cách thức như những thành phần khác trong trang web tương thích DOM.

Nói tóm lại, SALT được thiết kế đặc biệt để thích hợp với mô hình dữ liệu và môi trường thực thi được các nhà phát triển web sử dụng hiện nay. Toàn bộ đặc tả kỹ thuật của SALT có ở địa chỉ <http://saltforum.org/devforum/default.asp>.

## Ứng dụng SALT

Chúng ta hãy xem xét ứng dụng đa phương thức cơ bản dùng 4 thành phần thẻ lệnh quan trọng nhất của SALT.

Trước hết, bạn cần có một phương thức nhắc người dùng đưa thông tin vào. Thẻ lệnh SALT <prompt> được dùng để xác định nội dung phát âm như lời chào, hướng dẫn hay xác nhận. Nguồn nội dung có thể là văn bản được phát âm hay tập tin âm thanh. Với tập tin âm thanh thì công việc chỉ đơn giản là phát lại tập tin này, nhưng với văn bản thì phải cần đến công cụ TTS (text-to-speech) để chuyển thành tiếng nói.

Ví dụ 1 minh họa việc sử dụng thành phần <prompt> để phát âm văn bản 'Welcome to a SALT multimodal application'. Sự kiện oncomplete kích hoạt thành phần <listen> khi việc phát thông báo hoàn tất.

### Ví dụ 2

```
<salt:listen id='listenTag'>
  <salt:grammar src='TagGrammar.xml' />
  <salt:bind value='//Tag' targetElement='selectTag' />
</salt:listen>
```

Thành phần <listen> cung cấp điều khiển nhận dạng tiếng nói, chuyển tiếng nói thành văn bản và xử lý kết quả nhận dạng. Ví dụ 2 minh họa việc sử dụng thành phần <listen>. Trong ví dụ này, có 2 thành phần con: Thành phần <grammar> xác định từ khóa nhận dạng, thành phần <bind> chỉ đến thành phần khác của trang web, ví dụ như một thẻ lệnh HTML.

Thành phần <grammar> xác định chính xác cái mà chương trình có thể nhận dạng và có thể xem như là danh sách từ điển. Trình nhận dạng tiếng nói dịch tín hiệu số thể hiện câu nói thành các âm và so khớp với thể hiện âm của các từ trong danh sách được phân định theo văn phạm. Nếu tìm thấy từ trùng khớp thì kết quả được trả về. Văn phạm có thể được xác định ngay trong dòng lệnh hoặc có thể nằm ở một tập tin riêng và được chỉ đến bởi thuộc tính src của thành phần <grammar>. Trong ví dụ 2, văn phạm chứa trong tập tin ngoài 'TagGrammar.xml'; mã nguồn của tập tin này được trình bày trong ví dụ 3 định nghĩa nhận dạng 3 từ 'prompt', 'listen', 'grammar'.

Với 4 thành phần SALT cơ bản trên, chúng ta có thể ráp thành ứng dụng mẫu như ở ví dụ 4. Điều lưu ý đầu tiên là trang web này phải tham chiếu đến định nghĩa SALT ở dòng đầu.

```
<HTML xmlns:salt= 'http://www.saltforum.org/2002/SALT'>
```

Khi trang nạp vào trình duyệt, dòng:

```
<body onload='promptWelcome.Start()'>
```

bắt đầu phát thông báo được chỉ định là promptWelcome, công cụ TTS sẽ phát âm câu 'Welcome to a SALT multimodal sample. You may select...'. Sau khi thông báo kết thúc, sự kiện oncomplete của thành phần promptWelcome kích hoạt thành phần <listen> được chỉ định là listenTag. Việc này sẽ kích hoạt trình nhận dạng lắng nghe tiếng nói đầu vào. Khi phát hiện tín hiệu đầu vào, trình nhận dạng sẽ cố gắng nhận dạng dựa trên văn phạm đã được nạp bởi thành phần con <grammar>.

```
<salt:grammar src='TagGrammar.xml' />
```

Nếu nhận dạng thành công, thành phần con <bind> sẽ được thực thi.

```
<salt:bind value='//Tag' targetElement='selectTag' />
```

Lệnh này gán giá trị của nút <tag> ở dữ liệu XML đầu ra của trình nhận dạng cho thành phần selectTag, được định nghĩa ở dạng HTML trong ví dụ này là danh sách xổ xuống. Việc này dẫn đến tùy chọn tương ứng trong điều khiển HTML SELECT được chọn. Sau khi bind được thực thi, sự kiện nhận dạng thành công được kích hoạt với onreco, nó gọi hàm JavaScript



ProcessInput(). Lệnh trong ProcessInput() khi đó sẽ hiển thị khai báo văn bản đang được hiển thị trong điều khiển HTML SELECT. Lưu ý là ProcessInput() cũng được gọi bởi sự kiện onchange của điều khiển SELECT được kích hoạt khi người dùng chọn một tùy chọn bằng chuột hay bàn phím.

Ứng dụng đa phương thức đơn giản này minh họa cách thức SALT hỗ trợ tiếng nói điều khiển các thành phần trang web trong khi vẫn cho phép người dùng tương tác với các thành phần đó bằng chuột và bàn phím theo cách truyền thống. Ở khía cạnh lập trình, ứng dụng này cũng minh họa việc tương tác thông suốt giữa các thành phần SALT với mô hình dữ liệu và môi trường thực thi của trang web thông thường.

### Ví dụ 3

```
<grammar root='root' xml:lang='en-US' version='1.0'
xmlns='http://www.w3.org/2001/06/grammar'>
  <rule id='root' scope='public'>
    <ruleref uri='#Tag'/>
    <tag> $.Tag = $$ </tag>
  </rule>
  <rule id='Tag' scope='public'>
    <one-of>
      <item>prompt<tag> $. _value = 'prompt'</tag></item>
      <item>listen<tag> $. _value = 'listen'</tag></item>
      <item>grammar<tag> $. _value =
'grammar'</tag></item>
      <item>bind<tag> $. _value = 'bind'</tag></item>
    </one-of>
  </rule>
</grammar>
```

## SALT với công nghệ WEB của Microsoft

Việc kết hợp chặt chẽ SALT với mô hình dữ liệu và môi trường thực thi phổ biến của web (HTML, DOM, XML và ngôn ngữ kịch bản) đồng thời cũng hỗ trợ việc kết hợp SALT vào các môi trường phát triển thông dụng. Chúng ta hãy xem xét cách thức Microsoft hỗ trợ SALT trong môi trường IIS (Internet Information Server) và IE (Internet Explorer).

Trước hết, trên máy tính client bạn phải cài đặt Microsoft Speech Add-in cho IE. Add-in này cung cấp các thư viện DLL bổ sung cho IE 6.0 và thực hiện việc dịch lệnh SALT trong trình duyệt. Lưu ý là nếu bạn đã có cài Speech Application SDK (SASDK) thì không cần thực hiện bước này.

Bước kế tiếp, tạo một thư mục trên máy chủ IIS và thiết lập làm thư mục gốc của ứng dụng. Trong thư mục mới này, tạo một tập tin chứa mã lệnh ứng dụng SALT (dùng mã lệnh trong phần 'Ứng dụng SALT') và đặt tên với đuôi thích hợp, ví dụ SimpleSALT.slt. Tiếp theo, tạo một tập tin khác chứa văn phạm và đặt tên cho nó, ví dụ TagGrammar.xml.

Cuối cùng, thiết lập kiểu MIME trên máy chủ IIS cho trang chứa lệnh SALT. Ví dụ, nếu tập tin được đặt tên là SimpleSALT.slt thì bạn cần liên kết các tập tin \*.slt với loại SALT MIME. Một cách để làm việc này là dùng công cụ Internet Services Manager, chọn thư mục gốc của ứng dụng hay website, mở trang thuộc tính của nó, chọn mục HTTP Headers, nhấn nút File



Type trong phần MIME Map, và tạo một loại mới với phần mở rộng '.slt' và dạng nội dung 'text/salt+html'. Thao tác này sẽ làm cho Microsoft Speech Add-in tự động kích hoạt các đối tượng SALT và thực hiện chức năng SALT cho IE.

Giờ bạn có thể dùng trình duyệt để mở trang web hỗ trợ SALT trong thư mục đã tạo và chạy ứng dụng. Nếu bạn 'view source' trong trình duyệt, bạn sẽ chú ý thấy do việc ánh xạ MIME, đoạn mã tương tự như dưới đây đã được IIS thêm vào trang web của bạn trước khi chuyển nó đến trình duyệt:

```
<object id='saltobject26239220' CLASSID='clsid:DCF68E5B-84A1-4047-98A4-0A72276D19CC' VIEWASTEXT WIDTH=0 HEIGHT=0></object>
```

```
<?import namespace='salt' implementation='#saltobject26239220' />
```

```
<?import namespace='mm' implementation='#saltobject26239220' />
```

IIS đã chèn các dòng này trực tiếp vào trình IE để báo cho Speech Add-in biết xử lý các thành phần SALT.

#### **Ví dụ 4**

```

<HTML xmlns:salt='http://www.saltforum.org/2002/SALT'>
<HEAD>
<TITLE>A SALT Multi-modal Example</TITLE>
<SCRIPT language='JavaScript'>
function ProcessInput() {
    promptWelcome.Stop();
    listenTag.Stop();
    switch(selectTag.value) {
        case 'prompt': divDefinition.innerHTML =
            'The prompt element ' +
            'is used to specify the content of audio ' +
            'output, either as inline or referenced text, ' +
            'variable values, or links to audio files.';
            break;
        case 'listen': divDefinition.innerHTML =
            'The listen element ' +
            'is used for recognition and/or recording, ' +
            'and contains one or more grammars and ' +
            'optionally a set of bind elements to inspect ' +
            'and copy input.';
            break;
        case 'grammar': divDefinition.innerHTML =
            'The grammar element ' +
            'is used to specify possible user inputs ' +
            'with rules identified either inline or ' +
            'by reference.';
            break;
        case 'bind': divDefinition.innerHTML =
            'The bind element ' +
            'is used to bind values from spoken input ' +
            'into the page, or to call methods on ' +
            'page elements.';
            break;
        default: divDefinition.innerHTML = "";
    }
}
</SCRIPT>
</HEAD>
<body onload='promptWelcome.Start()'>
Select a SALT tag to view its definition:<P>
<SELECT id='selectTag' onchange='ProcessInput()'>
    <OPTION value='prompt'>prompt</OPTION>
    <OPTION value='listen'>listen</OPTION>
    <OPTION value='grammar'>grammar</OPTION>
    <OPTION value='bind'>bind</OPTION>
    <OPTION SELECTED value=''>-- Select a Tag ---</OPTION>
</SELECT>
<P>Definition:<P>
<DIV id='divDefinition'> </DIV>
<salt:prompt id='promptWelcome' oncomplete='listenTag.Start()'>
Welcome to a SALT multi-modal sample. You may select a SALT tag from the

```

dropdown list either by speaking or by using a mouse.

```
</salt:prompt>
```

```
<salt:listen id='listenTag' onreco='ProcessInput()>
```

```
  <salt:grammar src='TagGrammar.xml' />
```

```
  <salt:bind value='//Tag' targetElement='selectTag' />
```

```
</salt:listen>
```

```
</body>
```

```
</HTML>
```

## **Lời kết**

Tuy bài viết này chú trọng đến phía client của ứng dụng đa phương thức, lưu ý rằng SALT cũng hỗ trợ ứng dụng thoại thuần túy, không có giao diện đồ họa. Trong trường hợp này, điện thoại thực hiện chức năng thiết bị đầu cuối, bộ thông dịch SALT chỉ hỗ trợ tiếng nói kết hợp với máy chủ tiếng nói và điện thoại theo cách thức tương tự mô hình VoiceXML truyền thống.

Ngoài công nghệ Microsoft được dùng để triển khai ứng dụng mẫu trong bài viết này, còn có một số sản phẩm của các hãng khác hỗ trợ SALT (xem trên website của SALT Forum), và có cả giải pháp nguồn mở OpenSALT (<http://hap.speech.cs.cmu.edu/salt/>) của trường đại học Carnegie Mellon, thành viên của SALT Forum. OpenSALT đã đưa ra trình duyệt nguồn mở tương thích SALT 1.0 dựa trên trình duyệt nguồn mở Mozilla và sử dụng phần mềm nguồn mở tổng hợp tiếng nói Festival và nhận dạng Sphinx

***Theo PCWorld***