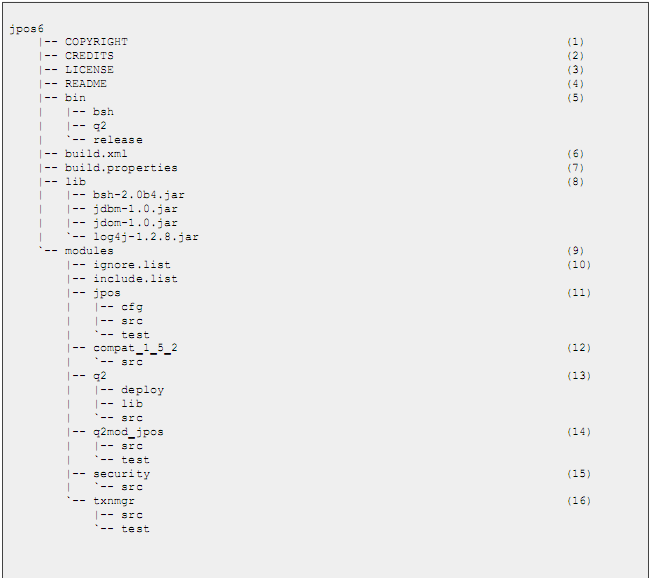
**Cách Thức Tổ Chức Biên Dịch Trong JPos**

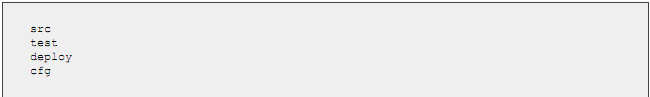
# Tổ chức thư mục của JPos Project



1. JPos’ copyright và Software copyright
2. Danh sách những người tạo nên JPos
3. Bản quyền GNU và AGNU
4. Thông tin để biên dịch JPos
5. Chứa mã Unix/Cygwin dùng để khởi chạy Q2
6. File cấu hình hỗ trợ biên dịch sử dụng Apache Ant
7. Định nghĩa các thuộc tính cho file cấu hình ở mục (6)
8. Các thư viện liện được cung cấp để chaỵ project
9. Chứa các module cần thiết cho quá trình biên dịch. Chúng ta có thể thêm các module khác do chúng ta tự định nghĩa vào đây.
10. File ignore.list cũng như include.list giúp file cấu hình ant biết được những module nào cần loại bỏ cũng như là cần biên dịch.
11. Jpos module là module chính của project chứa đựng các thứ khác liện quan tới những gói thao tác trên thoông điệp ISO8583
12. Compat\_1\_5\_2 chứa đựng các thành phần thừa kế trong các phiên bản cũ. Trong các phiên bản hiện tại ít được mở rộng.
13. Chứa gói Q2.
14. Q2 sử dụng một dịch vụ tên là QBean.QBean có thể sử dụng cho nhiều mục đích khác nhau.Thực hiện cấu hình và thực hiện đa tương tác với JPos component.
15. Chứa các gói liên quan đến bảo mật
16. Chứa các gói liên quan đến việc quản lý phiên giao dịch.

# Cơ chế biên dịch của JPos

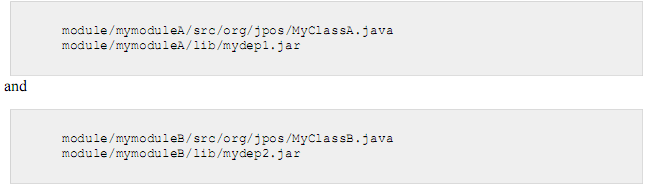
Trong bất kì module nào của JPos cũng chứa đựng các thư mục sau. Module chứa một vài thư mục sử dụng trong quá trình thực thi(runtime directory) ví dụ như **deloy**.Q2 quét vào thư mục này để khởi chạy cũng như là dừng các dịch vụ (QBean). Trong khi đó thư mục **cfg** chứa các file cấu hình lúc đang thực thi (runtime configuration) và các file định nghĩa các thuộc tính. Thư mục **cfg/lib** chứa các file jar được cung cấp để chạy các ứng dụng. Thư mục test dung đề chứa các Unit test.



JPos sẽ liên kết tất cả các file trong các module (bao gồm các module mặc định của JPos và các Module khác được chỉ ra trong **include.list** file) . Sau đó tự động copy chúng vào thư mục **build/module** trong quá trình biên dịch.

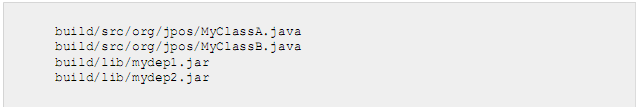
Xem xét ví dụ sau :

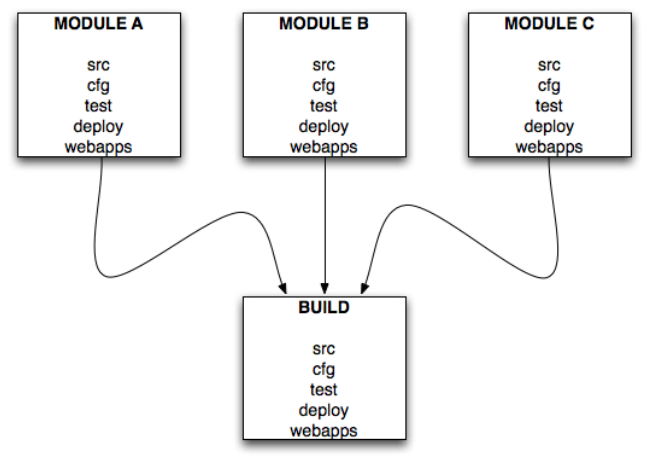
Chúng ta có moduleA và ModuleB chứa các file sau:



Khi chúng ta gọi lệnh ant để biên dịch JPos thì bước đầu tiên nó sẽ sao chép:

Tất cả các file \*.java trong thư mục moudule/\*\*/src 🡺 build/src và moudule/\*\*/cfg/lib hoặc module/\*\*/lib 🡺 build/lib





## Khởi chạy JPos

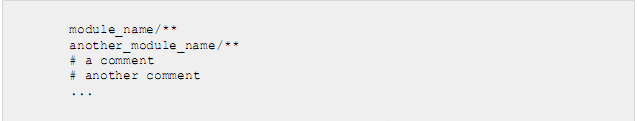
Vào thư mục build chứa file jpos.jar vừa mới biên dịch gõ lệnh sau:

java –jar jpos.jar

Có một Unix Shell Script (Chúng ta có thể chạy shell script này trên Window bằng Cygwin hay có thể dựa vào mẫu có sẵn tạo ra một file batch (\*.BAT) chạy trên window) và rồi called Q2 trong bin/q2 để khởi động Q2.

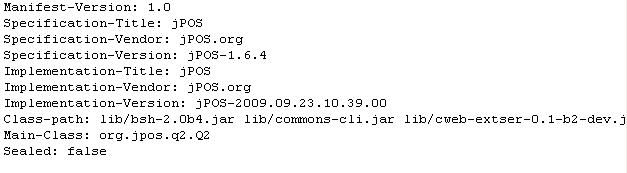
## Thêm vào hoặc loại bỏ module.

Chúng ta có thể thêm các module và gỡ bỏ các module cho quá trình biên dịch của JPos một cách dễ dàng thông qua module/include.list và module/ignore.list[[1]](#footnote-2)



## Manifest File

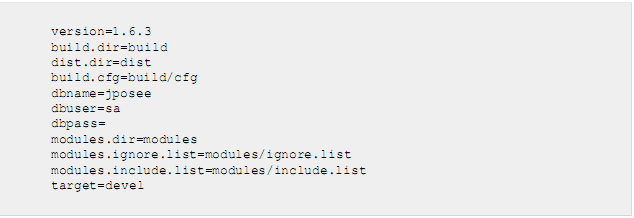
JPos tự sinh ra một file Manifest chứa thông tin của ứng dụng trong build/classes/manifest.mf



## Build.xml và build.properties

Build.xml là một file cấu hình sử dụng Apache Ant để hỗ trợ cho việc thực thi ứng dụng[[2]](#footnote-3).

File build.properties cấu hình các thư mục sẽ được thực thi trong quá trình biên dịch đã trình bày ở trên. File build.xml sẽ sử dụng các thư mục được cấu hình trong build.properties dưới dạng các biến được định nghĩa sẵn.



# Tài Liệu thàm khảo

[1] Jpos Programming Guide - Alejandro Revilla – Jpos.org

[2] JPosEE - Alejandro Revilla – Jpos.org

1. Trong JposEE cung cấp cho chúng ta một dãy các module được xây dựng sẵn.(Các module này chứa đựng trong thư mục opt ) [↑](#footnote-ref-2)
2. Nên xem file build.xml một cách cẩn thận để có hiểu quá trình sao chép file của Jpos vào các thư mục cũng như các target thể hiện trong ant -projecthelp [↑](#footnote-ref-3)