RPM 사용하기

김병찬 redhands@soback.kornet.nm.kr

V1.0, September 30, 1997

목 차

제 1 절 소개

RPM 은 아주 간편하고 사용하기 편한 패키지 관리자이다. 사용자 뿐만 아니라 만드는 사람에게 있어서도 좋은 패키징 툴이다. 여기선 RPM 을 사용하고 관리하는 방법을 설명하겠다.

제 2 절 일반적인 정보

2.1 RPM 구하기

현재 RPM 은 레드햇사에서 만들어 배포하고 있다. 물론 GPL 이다. 가장 최신의 RPM 패키지는 *ftp.redhat.com* <ftp://ftp.redhat.com> 에서 찾아볼 수 있다. 국내에서는 알짜 레드햇 이라는 이름으로 더욱 많이 알려져 있고, 또한 대부분의 레드햇 사용자가 알짜팀에서 제작한 알짜 레드햇을 사용한다. 한글 관련 패키지는 *ftp.kreonet.re.kr* <ftp://ftp.kreonet.re.kr/pub/Linux/hangul> 구할수 있을것이다. 알짜 레드햇에 대한 정보는 미약하나마 *적수네 동네* <http://www.linux.sarang.net> 에서 구할수 있을것이다.

제 3 절 RPM 사용하기

RPM 사용은 아주 간단하다. 설치, 제거와 질문 그리고 패키지 제작 이렇게 세부분으로 나눌수 있는데 여기서는 설치와 질문 부분만 살펴 보기로 하자. 패키지 제작에 관한것은 이미 적은 글도 있고 (엉성하지만) 초보에게는 아직은 골치아픈 이야기가 될테니까 말이다. 사실 RPM 관련의 명령들은 이 글이 아니라도 많은곳에서 보고 배울수 있다. `man rpm` 으로 볼수도 있을것이고 RPM HOWTO 도 이미 번역이 되어 있다. 또는 `rpm --help` 를 봐도 된다. 참고 바란다.

RPM 패키지의 이름에 대해서 잠시 알아보고 넘어가도록 하자.

```
foobar-1.0-1.i386.rpm
```

RPM 패키지 이름에서 - 는 각 필드를 구분하는 것이다. ^^

- 우선 맨 앞부분의 *foorbar* 는 패키지의 이름이다. 가장 기본적인 것이다.
- 두번째 *1.0* 은 패키지의 버전이다. 프로그램 버전과 일치하는 것이다.
- 세번째의 *1* 은 패키지의 릴리즈 번호이다. 이 패키지가 몇번째로 만들어진 것인지를 나타낸다. 똑같은 프로그램으로 버전이 같다고 해도 이전에 만든 패키지를 다시 재 패키징을 하였다면 (뭔가 변화가 있고) 릴리즈 번호를 올리게 된다.
- 네번째의 *i386* 은 어떤 시스템에서 쓰이는것인지를 나타내는 것이다. *i386* 이라면 당연히 PC 계열을 나타네는 것이고 *sparc* 이라면 스팍 리눅스용, *alpha* 라면 알파 리눅스용일것이다. (현재 레드햇은 이 세개의 시스템용으로 나온다.) 이 네번째 필드가 *src* 라고 적힌것이 있는데 그것은 소스 RPM 이다. 바이너리 패키지를 만들기 위해 필요한것이다. ^^
- 마지막의 `rpm`은 소위 말하는 확장자이다. RPM 패키지라는것을 나타내 준다.

3.1 설치, 제거

RPM 패키지의 설치와 제거는 아주 간단하다. 밑의 예를 보자.

```
rpm -i foobar-1.0-1.i386.rpm
```

보통은 이런식으로 사용하게 된다.

```
rpm -ivh foobar-1.0-1.i386.rpm
```

이 명령으로 하면 설치 되는 모습이 # 마크로 표시 될것이다. 썰렁함을 면해 볼 수 있을것이다.

RPM 으로 패키지를 설치할 때는 사실 위의 명령보다는 `-Uvh` 옵션을 사용하는 습관을 익히는것이 좋을것이다.

```
rpm -Uvh foobar-1.0-1.i386.rpm
```

이 명령을 사용한다면 rpm 은 이 패키지의 이전 버전이 설치 되었는지를 보고 가 되어 있다면 업그레이드를 할것이다. 또 설沈되어 있지 않드면 설치 할 것이다. 그냥 `-i` 또는 `-ivh` 로 설치 한다면 이전 버전의 같은 패키지에 대한 정보는 사라지지 않을것이고 또한 이전의 설정 파일도 백업되지 않는다. 다시 한번 강조하지만 되도록 `rpm -Uvh` 를 사용하라.

3.1.1 설치, 제거의 여러 옵션들

그외 몇가지 부수적인 옵션이 있는데 여기서 설명하도록 하겠다. 이 부수적인 옵션들은 설치 또는 업그레이드 또는 제거 옵션뒤에 붙이게 된다. (제거 옵션에서는 `--nodeps` , `--noscripts` , `--test` 만을 사용한다.)

- `--nodeps` :
  의존성을 무시하고 설치한다. 가장 많이 겪게 되는 문제인데 RPM 에서는 어떠한 패키지가 깔려 있지 않으면 그거에 영향을 받는 패키지는 못 까는 경우가 있다. 그럴때 사용하는것이 `--nodeps` 이다.
- `--force` :
  강제로 설치 하게 된다. 보통 패키지 설치시에 현재의 패키지에 포함된 파일이 이미 다른 패키지에 의해 설치 되 있을때 충돌을 한다며 에러가 나는데 이 옵션으로 설치 할수 있다. 참 이 옵션은 이미 있는 파일은 덮어 쓰지 않는다. 이미 있는 파일마저 덮어 쓸려면 `--replcaefiles` 를 사용하면 된다.
- `--oldpackage` :
  새로운 패키지를 지우고 구버전의 패키지로 교체할 때에 사용한다.
- `--percent` :
  패키지 파일을 설치하는것을 퍼센트로 표시해준다. 멋있게 나오는것은 전혀 아니니 기대하지 말자! (ftp 설치 같은것을 할때 유용할 것이다.)
- `--replacepkgs` :
  이미 같은 패키지가 설치되 있더라도 다시 설치한다.
- `--replacefiles` :
  이미 설치된 다른 패키지의 파일을 덮어 쓰면서라도 목숨 걸고 설치한다. (웬만하면 쓰지 말자 무섭다!)
- `--root` <디렉토리> :
  <디렉토리>를 마치 / 처럼 생각하고 설치를 한다. 즉
- `--root /tmp` 라고 한다면 `/tmp` 가 / 인 것으로 생각하고 그 이하로 설치하게 될것이다.
  한가지 문제가 있다면 이 명령을 사용하면 RPM 정보를 기록하는 파일을 지정한 디렉토리 밑에서 찾게 된다. 흐흐
- `--test` :
  패키지를 실제로 설치하지는 않고 충돌이나 의존성 문제가 있는지만을 검사한다.
- `--noscripts` :
  스크립트를 실행하지 않는다. (레드햇 패키지에는 4개의 스크립트가 들어간다. 설치 전후, 제거 전후 이렇게 4개이다.)
- `--excludedocs` :
  문서 파일은 설치 하지 않는다.

이 옵션들을 모두 기억할 필요는 없다. 단 `--nodeps` 만은 꼭 머리 속에 넣어 두기 바란다. 흘려 듣지 말기를

한가지 매우 재밌는것이 있다. 윈도우 같은 상용 프로그램들에서는 상상도 못할 일이다. 바로 FTP 를 이용한 설치이다. (상용 프로그램 어떤것이 그냥 FTP 로 받아서 설치 하도록 해주겠는가?)
FTP 를 이용해서 설치 하기 위해서는 네트웍에 연결이 되어야 하고 정확한 위치를 알아야만 한다.

```
rpm -i ftp://ftp.foobar.com/pub/redhat/foobar-1.0-1.i386.rpm
```

`-i` 가 무엇을 뜻하는지를 까먹었다면 다시 처음부터 읽기 바란다. -.-+ 여기서 –precent 를 사용한다면 재밌을 것이다. 그외의 질문도 가능한데 그건 좀 있다가 얘기하자. 아 골아퍼!

3.2 질문 하기

패키지의 장점을 그대로 드러내는 곳이다. 현재 설치된 패키지나 설치하지 않은 패키지에 대한 정보를 보고 그 패키지의 구성을 살펴 볼수도 있고, 무슨 파일이 있는지 어떤 패키지와 의존성이 걸리는지 누가 만들었는지 등등 많은 것을 알수 있다. 하나 하나 설명해 보기로 하겠다.
RPM 에서 질문을 수행하기 위해서는 -q 라는 옵션을 쓴다. rpm -qa 이런식으로 -q 이후에 다른 옵션을 붙여 쓰는 것이다. 기억하라 RPM 에서 옵션을 적을때 꼭 rpm -q -a -l 이런식으로 뛰어 써야 하는것은 아니다 그냥 rpm -qal 이렇게 붙여 쓰면 된다.

도데체 이놈의 컴에 어떤 패키지가 깔려 있을까? 답답할때가 있을것이다. 내가 이걸 깔았나 안 깔았나 패키지가 몇개나 깔려 있는건가... 대충 프로그램 이름은 아는데 패키지명은 정확히 모르겠고 아주 답답할때가 ^^ 이렇게 해보자.

```
rpm -qa |more
```

위에서 질문을 수행하기 위해서는 -q 를 사용한다고 하였다. 다시 한번 강조 하지만 기억하라. -a 옵션은 모든 패키지를 보여 달라는 것이다. 다른 명령과 조합해서 쓸수도 있다. 참 뒤에 |more 는 너무 많이 출력되니까 한 페이지씩 보기 위해서 사용한거다. ^^
심심할땐 이런 명령을 사용해 볼수도 있을것이다.

```
rpm -qal
```

-l 옵션은 어떤 패키지에 속한 파일들을 모두 출력하는 것이다. 여기서 -a 옵션과 같이 사용했으니 설치된 모든 패키지에 속한 파일들을 보여줄 것이다. 꽤 오래 걸리겠지만... 이런식으로 여러개의 옵션을 조합해서 사용함으로써 더욱 많은 정보를 더 쉽게, 또한 간단하게 알아 낼수 있다.

어떤 패키지에 대한 대략적 정보를 알고 싶으면 어떻게 할것인가?

```
rpm -qi foobar
```

이런식으로 사용한다. -q 는 다시 안 말해도 알것이고 -i 는 info 의 약자일것이다. 그 패키지에 대한 약간의 정보를 제공하게 된다. 한가지 기억해야 할것은 패키지의 이름만을 적는것이다. 저 위에서 설명 하였다. ^^ 다시 한번 보든지 기억을 해 내든지 하기를...

```
$ rpm -qi howto-ko

Name        : howto-ko                    Distribution: (none)
Version     : 1.1                               Vendor: Byeng-Chan, Kim <redhands@soback.kornet.nm.kr>
Release     : 3                             Build Date: Tue Sep 30 19:18:42 1997
Install date: Tue Sep 30 19:19:23 1997      Build Host: issue.ml.org
Group       : Extensions/Korean             Source RPM: howto-ko-1.1-3.src.rpm
Size        : 612864
Summary     : Various HOWTOs from the KLUG Document Project
Description :
This is the best collection of Korean Linux documentation there is.
If you want to find newer versions of these documents, see
http://klug.ajou.ac.kr or http://members.iWorld.net/mangchi
For the versions in this package see /usr/doc/HOWTO/translations/ko(rea).
Good Time!  (Sorry, I don't hanv Tex. so, I don't make ps and dvi...)
```

이런식의 화면을 볼수 있을것이다. 각각 무엇을 나타내는지 파악이 되리라 싶다. 패키지의 이름, 버전, 릴리즈 번호, 만든 사람, 만든곳, 만든 시간, 크기, 설명 등을 볼수 있다. ^^

이렇게 사용해 보면 또 어떨까?

```
rpm -qif /bin/ls
```

-i 옵션은 위에서 설명 했는데 -f 옵션은 또 무엇인지? 음 머리가 좋은 사람은 직접 해봤거나 대충 짐작을 할 것이다. 흐흐 (머리 좋은가?) -f 는 지정한 파일을 포함하는 패키지를 알려준다. 그런데 -i 옵션을 사용했으니까 그 패키지에 대한 대략적인 정보를 보여 줄것이다. 그냥 -f 만 쓰면 패키지 이름만 보여주고 말겠지만 재밌지 않은가? 이건 또 어떤가?

```
rpm -qlf /bin/ls
```

음 무슨 명령인지 머리속을 팍 스쳐 지나갔으리라고 생각한다. 아까 -l 옵션은 그 패키지에 속한 파일들을 다 보여 준다고 했다. 그렇다면? 후후 /bin/ls 파일을 포함한 패키지에 속한 모든 파일을 보여 줄 것이다. 아차차 다시 강조하자 얍! (뭐 말할지 알겟지만) 질문은 언제나 -q 옵션을 사용한다. 기억 기억

-f 명령은 아주 유용하게 쓰인다. 어떤 파일이 도데체 어느 패키지에 있는지 알수가 없을때, 에러가 났을때, 연관된 파일을 찾아 낼때 등등 생각해보면 아주 많은 경우에 쓰일것이다. 어떤 프로그램의 패키지가 있는데 이게 에러가 계속 나고 실행이 안된다. 요리조리 궁리해보면 이런 방법을 사용해 볼수 있을것이다.

```
rpm -qsRf /bin/foobar
```

무엇일까? (흐흐 또 새로운 옵션이 등장) -qf 는 알겠는데 -sR 은 또 뭔지..... 자자 성급하게 알려고 하지 말자 설명할거니까... 방금 패키지 에러 예기를 했으니 역시 똑똑한 사람은 짐작을 할수도 (난 멍청한가?)

-s 옵션은 패키지가 포함한 파일들의 상태를 보여주게 된다. 이 패키지에 속한 파일들이 설치가 되었는지 지워졌는지 딴걸로 바뀌었는지를 알수 있다. 각각 normal, not installed, replaced 로 표시 된다. 어떤 파일이 쉬익 지워졌다면 그 프로그램이 실행되지 않을수도 있지 않는가^^

아 그리고 -R 은? 후후 이 명령은 패키지가 의존하고 있는 다른 파일 또는 패키지를 보여 준다. 의존이라고 하면 잘 이해가 가지 않을수도 있는데 이 패키지의 프로그램을 실행하기 위해서 필요한 파일, 프로그램들을 나타내는 것이다. 왠만한 프로그램은 libc 가 없으면 실행 되지 않을것이다. (이해가 되었나? 흠?) 그러니까 이 파일이 의존하는 프로그램들을 찾아서 흐흐 그 프로그램들이 또 이상은 없는지 확인하면 호호 에러를 고칠수도 있을것이다.

자자 다시 고, 못 먹어도 고!!!

```
rpm -dc foobar
```

흠 이건 뭐에 쓰는 물건 아니 옵션인고? -d 는 문서 파일만 보여준다. 흐흐 그리고 -c 는 설정 파일만 보여준다. (붙여쓰는 일은 거의 없지만 인간이 글 적다 보면 더 적기 귀찮을때가 있지 않겠는가) 이 패키지의 설정 파일만 보거나 문서 파일만 보고 싶을때 사용하면 되지 않겠는가^^

대충 정리 끝났다. 이정도만 알면? 써먹을수 있지 않을까? 아 참 중요한거 하나 빼먹은거 같다. -p 옵션이다. 설치된 패키지에 대한 옵션이야 그냥 -qi 이런식으로 사용하면 되는데 설치 되지 않은 패키지에 대해서는 -p 옵션을 사용해야 한다. 완전한 경로와 정확한 이름을 써야 한다. 패키지 이름만 적는것이 아니다.

```
rpm -qip /tmp/foobar-1.0-1.i386.rpm
```

무슨 말인지 잘 모르겠으면 손으로 해보라. 머리는 컴이 아니니까 ^^

이런, 또 빼먹었다. 이걸 잘 인식하지 않는 사람이 있을텐데 후 --scripts 라는 옵션이다. 물론 -q 와 함께 사용한다. 요런식으로,

```
rpm -qp --scripts /tmp/foobar-1.0-1/i386.rpm
```

아까 잠시 말한적이 있을것이다. RPM 패키지에는 4개의 스크립트가 포함된다고 말이다. 설치 전후, 제거 전후 그런데 왜 이렇게 이걸 설명을 하냐면 중요하기 때문이다. 패키지라는것이 아주 위험하다. 잘 사용하면 아주 편하지만 악의를 가진 사람에게 있어서는 트로이의 목마로 사용될수도 있는것이다.

패키지는 root 로 설치하게 된다. 이 명령을 사용할때가 바로 이때다 설치 전!!! (물론, 설치 후에 사용해도 상관이 없겠지만...) 왜 그런가 다시 한번 말하지만 4개의 스크립트에 무엇을 집어넣어도 상관이 없다. 설치후에 rm -rf / 가 실행되도록 한다면 당신의 시스템은? 풍지박산 개박살^^ 꼭 기억하라 설치전에 꼭 체크를 해보는 습관을 가져라. 그리고 믿을수 있는곳에서 패키지를 받아 와라. 그렇지 않고 개박살난 후에 후회해봤자 소 잃고 외양간 다 날라가고 집구석까지 폭삭 내려 앉은 후에 외양간 고친다고 발악하는 꼴이 될꺼라고 이 연사 강력하게 주장합니다. 에궁

에이 이왕 가는김에 -K 도 알아보자. 이건 -q 를 같이 사용하지 않는다.

```
rpm -K /tmp/foobar-1.0-1/i386.rpm
```

이 명령은 파일에 대해서만 수행하게 된다. 무슨 작동을 하는 명령이냐면 (뜸 들이기) 패키지 파일에 대한 md5, pgp 사인, 파일의 크기 에 대하여 조사 하는 명령이다.

```
/tmp/foobar-1.0-1.i386.rpm: size pgp md5 OK
```

이런식의 출력을 보게 될것이다. pgp 사인은 이 패키지의 제작자가 위에 자신이 만들었다는 표시 정도로 하게 되는것으로 내가 믿을수 있는 사람의 사인이 되어 있다면 이 패키지는 충분히 믿을수 있는것 아니겠는가? ^^ 알짜 레드햇 제작진의 PGP 공개키는 *적수네 동네* <http://www.linux.sarang.net> 에서 구할수 있다. 또한 md5 와 size 로 패키지가 변경이 되었는지 깨졌는지를 간단하게 알수 있다. 만약 저 테스트중에 크기와 md5 부분에서 이상이 난다면 누군가 악의적으로 바이너리 해킹을 했거나 전송 받는중에 깨진것일거다.

```
rpm -Kv /tmp/foobar-1.0-1/i386.rpm
```

이렇게 -v 또는 -vv 옵션을 같이 사용하면 좀더 상세한 내용을 볼수 있다. -v 나 -vv 는 어떠한 명령에도 같이 사용할수 있다. 좀 더 상세한 정보를 원할때 사용하기 바란다. ^^

```
rpm -qlv foobar
```

이런식으로 사용한다면 그 패키지에 포함된 파일들에 대한 상세한 정보를 보여줄 것이다. 크기, 퍼미션, 위치 등등

아까 설명한다고 했었는데... FTP 로도 질문이 가능하다. 다만 서버에서 배려를 했다면 말이다. ^^ 설치와 마찬가지로 정확한 위치를 알아야 한다. 또한 질문의 모든 옵션을 사용할수 있다. -K 만 빼고 ^^ 앗 -a 도 빼고 ^^

```
rpm -qip ftp://ftp.foobar.com/pub/redhat/foobar-1.0-1.i386.rpm
```

3.2.1 질문하기의 여러 옵션들...

-q와 함께 사용한다는거 잊지 말고 대략적으로 옵션들의 기능을 적겠다.

- <패키지명> :
  <패키지명> 에 대한 질문을 수행한다.
- -a :
  모든 패키지에 대하여 질문을 수행한다.
- -f <파일> :
  <파일> 을 포함하는 패키지에 대한 질문을 한다.

- -F :
  -f 와 같은 기능이지만 표준 입력으로 부터 파일명을 읽는다. 그러니까 `ls /usr/bin | rpm -F` 이런식으로 사용한다.
- -p <패키지 파일> :
  패키지 파일에 대한 질문을 한다.
- -P :
  -p 와 같은 기능이지만 표준 입력으로 부터 파일명을 읽는다. ^^
- -i :
  패키지에 대한 대략적인 정보를 출력한다.
- -R :
  패키지가 의존하고 있는 파일 또는 패키지 목록을 보여준다.
- -l :
  패키지에 속한 파일들을 보여준다.
- -s :
  패키지에 속한 파일들의 상태를 보여준다.
- -d :
  문서 화일만 보여준다.
- -c :
  설정 파일만 보여준다.
- --scripts :
  설치, 제거 과정에 사용되는 쉘 스크립트가 있다면 그 내용을 출력한다.

제 4 절 끝내면서

거의 대부분 HOWTO 와 메뉴얼 페이지에 있는 내용이지만 좀 더 쉽게 적어 볼려고 했다. 쉬울지 모르겠지만... 좀 더 쉽게 그리고 알차게 RPM 패키지를 사용하기 바란다.

4.1 질문 사항이 있다면?

RedHat 홈페이지 <http://www.redhat.com> 나 *적수네 동네* <http://www.linux.sarang.net> 에 들러보기 바란다. 또, RPM HOWTO 나 메뉴얼 페이지 등을 자세히 살펴 보기 바란다. 메일로 질문은 되도록 사양을 하며 *나우누리* <telnet://nownuri.net> 질문 게시판에 질문을 하는것도 좋은 방법이다.

4.2 저작권

이 문서는 GPL 라이센스를 따른다. 알아서 사용하기 바란다. 흐흐^^ 다시 한번 말하지만 이 인간 적수가 쓰는 글은 모두 GPL 이다. ^^