



BYTE-CODE
Technology Solutions

RPM: Speed up your deploy

Francesco Crippa - fcrippa@byte-code.com

RPM: RedHat Package Manager

- Make your deploy:
 - Fast
 - Easy
 - Standard
 - Secure
 - **Predictable**
- Every time it's the same deploy



Install, update and remove

- An RPM package contains files, directories, permissions and some scripts to easy manage software without interactive process

```
[root@a~]# rpm -ih witick-1.5-4.noarch.rpm  
[root@a~]# rpm -Uh witick-1.5-4.noarch.rpm  
[root@a~]# rpm -e witick
```



Important Note...

- Please, be careful to the difference between “**Package Name**” and “**Package File Name**”:
 - witick-1.5-4.noarch.rpm is the filename of “witick” package
 - `rpm -Uh <file_name>`
 - `rpm -e <package_name>`
 - `rpm -ql <package_name>`
 - `yum install <package_name>`



Yum: dependency simplified

- One of the most terrible pain with rpm is the dependency between packages.
- Yum help you to solve this step...

```
[root@a~]# rpm -Uh krb5-server-1.5-17.i386.rpm
```

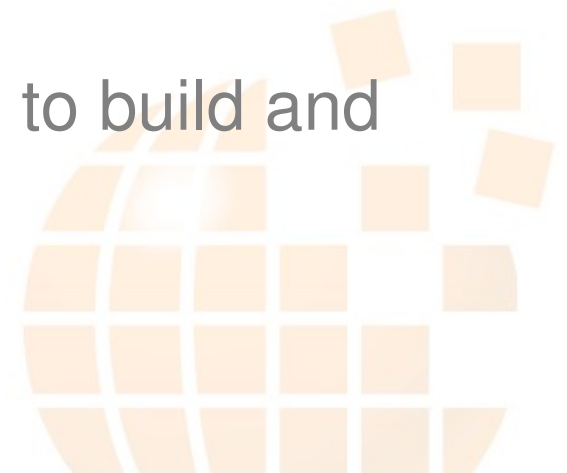
```
Error! Failed Dependencies
```

```
krb5-libs = 1.5-17 is needed by krb5-server-1.5-17
```

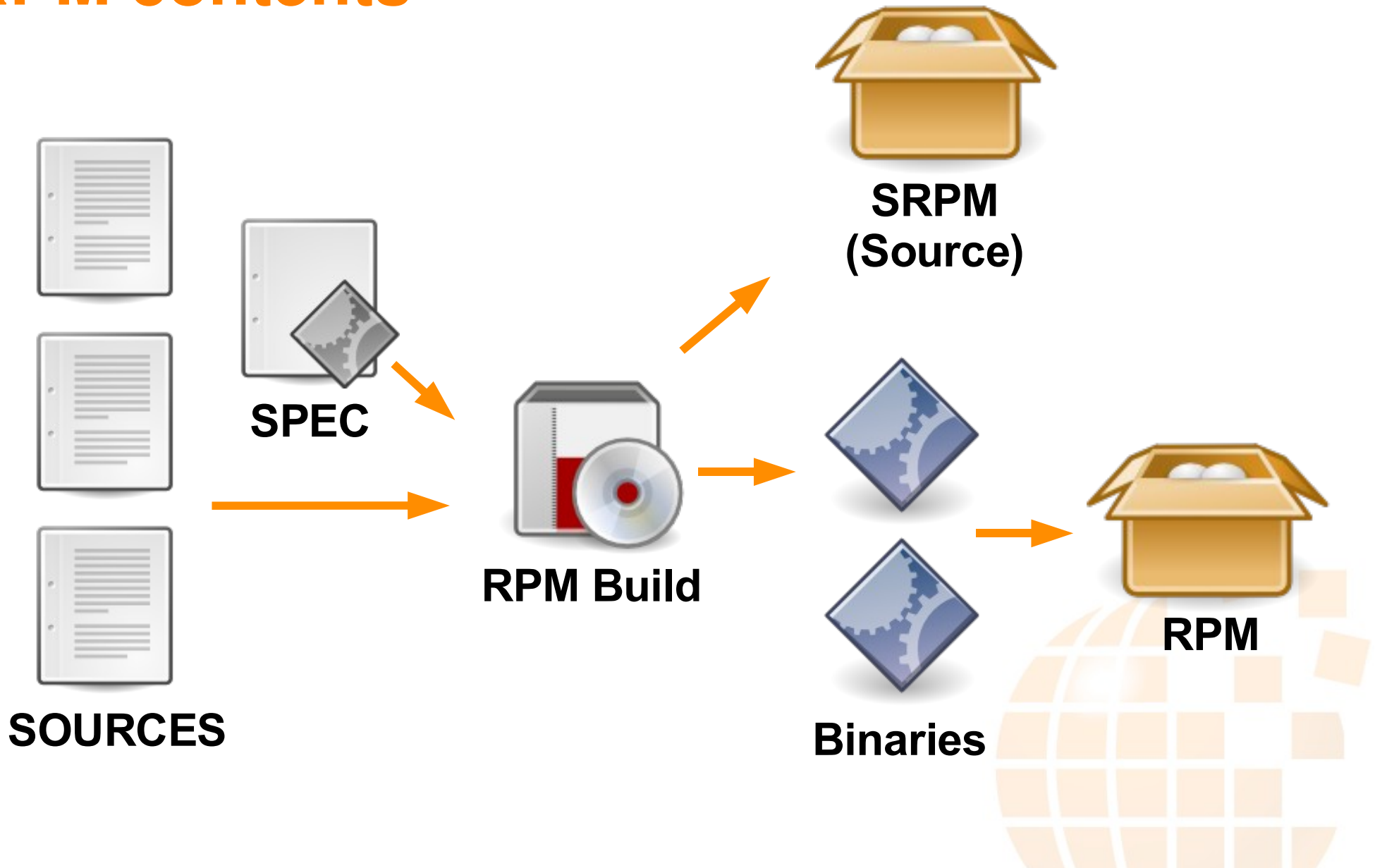


RPM: Make it standard

- Java, C++ and Ruby applications could use very different deployment and building processes:
 - Ant
 - Maven
 - MakeFile
 - Rake
 - ...
- RPM define the process to use right tools to build and deploy right application



RPM contents



RPM Build infrastructure

- To build an RPM you need a specific infrastructure:
- /~
 - .rpmmacros
 - /rpmbuild
 - /SOURCES
 - /BUILD
 - /RPMS
 - /i386
 - /i686
 - /x86_64
 - /noarch
 - SRPMS
 - SPECS



.rpmmacros

- `~/ .rpmmacros`: Contains specific configuration for your user:

```
%packager      Francesco Crippa <fcrippa@byte-code.com>
%vendor        Byte-Code srl
%_topdir       /home/fcrippa/rpmbuild
%_signature    gpg
%_gpg_name     Francesco Crippa <fcrippa@byte-code.com>
```



All the magic is SPEC file

- All you need is SPEC file. It contains:
 - Source files location
 - Patches
 - Meta-information about package (name, author, version...)
 - Changelog
 - Build process
 - Install process
 - Uninstall process
 - Requirements for building
 - Requirements for deploying



SPEC Head

Summary: Hello world

Name: hello-c

Version: 1.0

Release: 1

License: GPL

Group: Amusements/Games

URL: <http://www.byte-code.com>

Source0: `%{name}-%{version}.tar.gz`

BuildRoot: `%{_tmppath}/%{name}-%{version}-%{release}-root`

BuildArch: i386

`%description`

Hello world

...



SPEC: build and install

```
%prep
```

```
%setup -q
```

```
%build
```

```
gcc -o hello-c hello.c
```

```
%install
```

```
rm -rf $RPM_BUILD_ROOT
```

```
mkdir -p $RPM_BUILD_ROOT/bin
```

```
cp hello-c $RPM_BUILD_ROOT/bin
```

```
%clean
```

```
rm -rf $RPM_BUILD_ROOT
```



Files and changelog

```
%files
```

```
%defattr(-,root,root,-)
```

```
/bin/*
```

```
%doc README.txt
```

```
%config /etc/hello.conf
```

```
%changelog
```

```
* Sun Sep 16 2007 Francesco Crippa <fcrippa@byte-  
code.com> - 1.0
```

```
- Initial build.
```



Building RPM

- “rpm-build” must be installed on your system
- --rmsource help you to keep free your hard disk
- --sign help you to sign package during building

```
[root@a~]# rpmbuild -ba hello.spec  
[root@a~]# rpm -ba --sign -rmsource hello.spec
```



Sign your security

- It's possible to sign (or re-sign) your preferred rpms.
- Sign is made by GnuPG
- You can resign a signed package
- You can sign a package during rpm building



Racommendations

- You MUST not use RPM in %pre and %post
- Never use “cp” and “mv” in %pre and %post
- Increment version or release on every build
- You MUST not ask any question to the user during %pre and %post (process must be non-interactive!)



Next Step

- Continuous Build:
 - **Koji** <https://hosted.fedoraproject.org/projects/koji/>
- Version Tree Management:
 - **Bodhi** <https://hosted.fedoraproject.org/projects/bodhi/>





www.byte-code.com



redhat.
ADVANCED
BUSINESS PARTNER