RED HAT :: BOSTON :: 2008

SUMMIT

JUNE 18–20, 2008

Karsten Wade

Sr. Developer Community Mgr, Red Hat

Fedora Packages for Red Hat Enterprise Linux

What am I here to talk about?

- RPM Packages
  - ... that are community-set best practices
  - ... that are across Fedora and RHEL
  - ... that are the way to get “in” to Fedora and EPEL
  - ... may be the way to get “in” to RHEL
- RPM is the center of an ecosystem
  - History == important
  - Future == crucial
  - ... requirement to be in Fedora
    - ... built from source

This talk is about EPEL

- Extra Packages for Enterprise Linux
    - Started at FUDCon Boston 2007
- More than just another maintenance branch
    - Best way to get software ready for RHEL
        - Six to thirty six month lead time

Covering EPEL ground today

- What is EPEL?
- How do you get a package in to EPEL?
- What does the lifecycle look like?
- Why is it worth the effort?

Fedora wants to work with open source businesses

- It's in our blood :)
- Connect with the full lifecycle – devel, qa, releng, end-user testers, etc.
- Some knowledgeable folks here at the presentation
- This is a discussion that I'm leading, so ask away

What is EPEL?

- Common ground
- Set of packages with maintainers specifically interested in supporting a package for RHEL and friends
- Follow a RHEL-like philosophy – EPEL maintainers must agree to this
  - Longer testing cycle
  - Focus on security and bugfixes over enhancements
  - Don't drag in upstream code just because it is the “latest”

What does it mean to “be in Fedora/EPEL”?

■ Your goal is 'yum install foo', where foo is your app

b) Pristine source tarball/zipfile from an upstream project

c) Minimal, separated patches (best practice = fix the upstream)

d) Spec file

e) All of that is in cvs.fedoraproject.org

f) FAS knows you own the package and the traffic it generates

g) The package can then be:

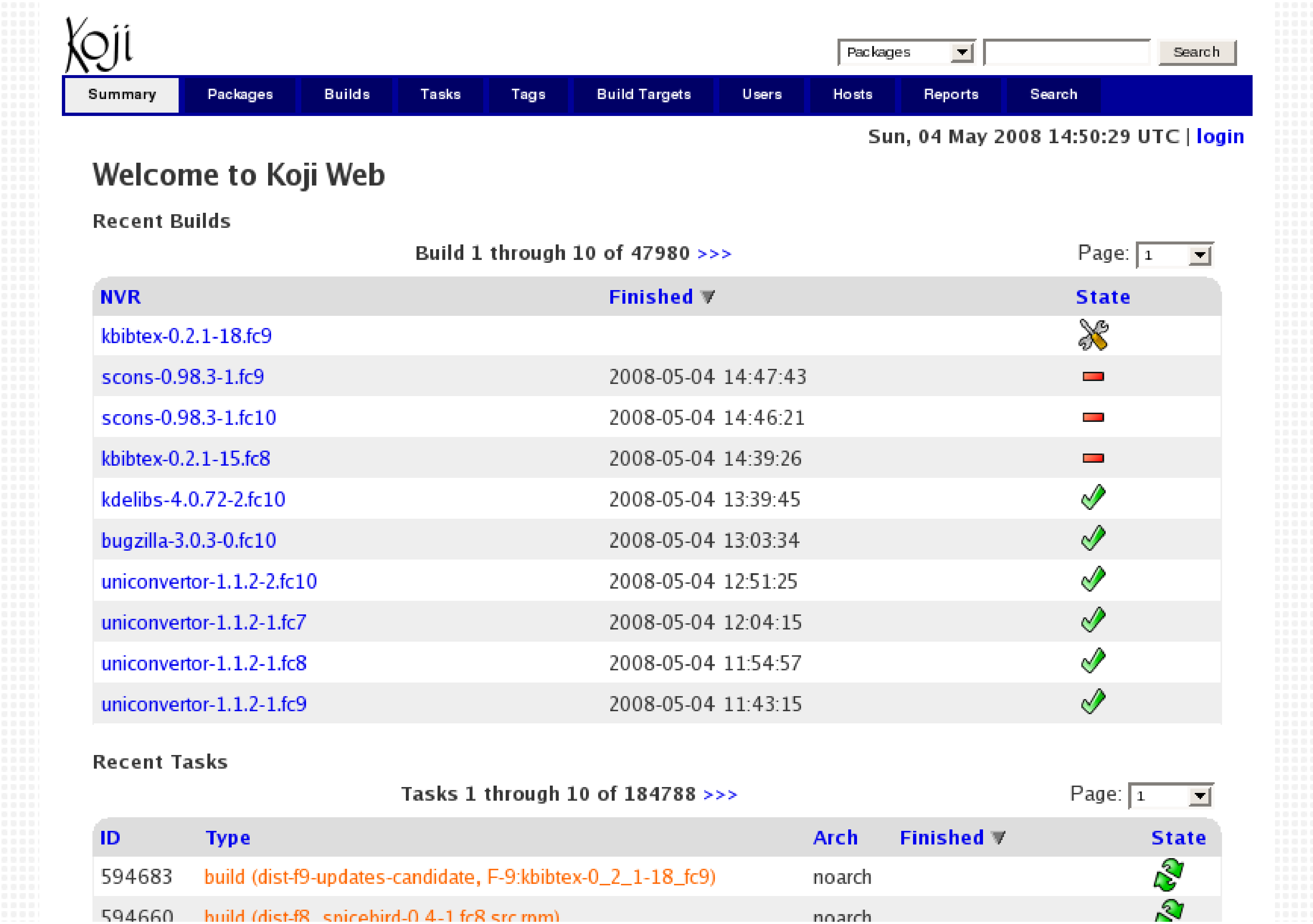
- Built from source
- Signed
- Made available for testing
- Included in the main repository

Result:

```
yum install foo
```

http://koji.fedoraproject.org - view it in the build system

http://bodhi.fedoraproject.org - dev/user social package network



Sun, 04 May 2008 14:50:29 UTC | login

Welcome to Koji Web

Recent Builds

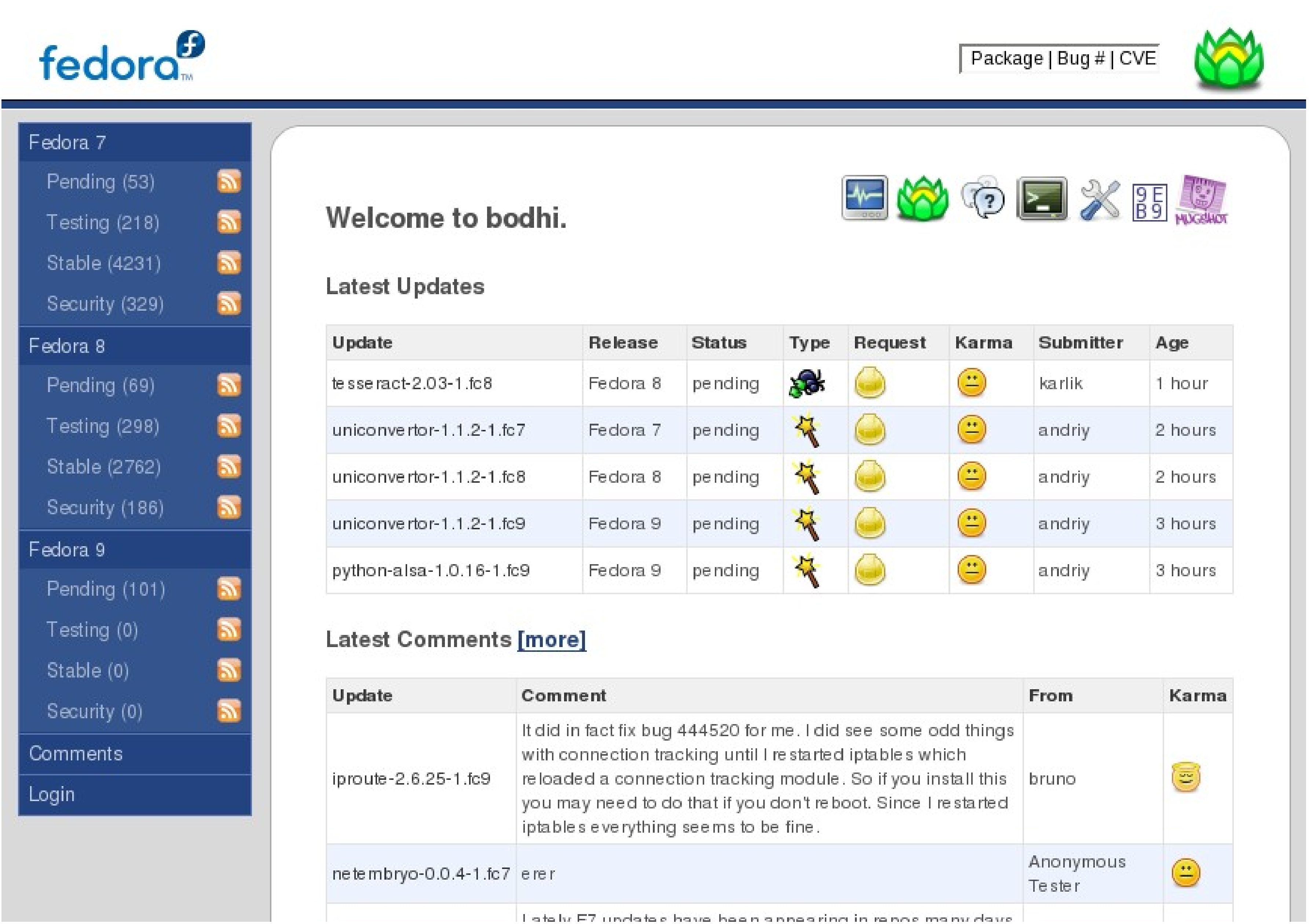
Build 1 through 10 of 47980 >>> Page: 1

NVR	Finished ▼	State
kbibtex-0.2.1-18.fc9		🛠
scons-0.98.3-1.fc9	2008-05-04 14:47:43	▬
scons-0.98.3-1.fc10	2008-05-04 14:46:21	▬
kbibtex-0.2.1-15.fc8	2008-05-04 14:39:26	▬
kdelibs-4.0.72-2.fc10	2008-05-04 13:39:45	✓
bugzilla-3.0.3-0.fc10	2008-05-04 13:03:34	✓
uniconvertor-1.1.2-2.fc10	2008-05-04 12:51:25	✓
uniconvertor-1.1.2-1.fc7	2008-05-04 12:04:15	✓
uniconvertor-1.1.2-1.fc8	2008-05-04 11:54:57	✓
uniconvertor-1.1.2-1.fc9	2008-05-04 11:43:15	✓

Recent Tasks

Tasks 1 through 10 of 184788 >>> Page: 1

ID	Type	Arch	Finished ▼	State
594683	build (dist-f9-updates-candidate, F-9:kbibtex-0_2_1-18_fc9)	noarch		🔄
594660	build (dist-f8, spicebird-0.4-1.fc8.src.rpm)	noarch		🔄



Fedora 7
- Pending (53)
- Testing (218)
- Stable (4231)
- Security (329)

Fedora 8
- Pending (69)
- Testing (298)
- Stable (2762)
- Security (186)

Fedora 9
- Pending (101)
- Testing (0)
- Stable (0)
- Security (0)

Comments

Login

Welcome to bodhi.

Latest Updates

Update	Release	Status	Type	Request	Karma	Submitter	Age
tesseract-2.03-1.fc8	Fedora 8	pending	🐛	●	😐	karlik	1 hour
uniconvertor-1.1.2-1.fc7	Fedora 7	pending	🪄	●	😐	andriy	2 hours
uniconvertor-1.1.2-1.fc8	Fedora 8	pending	🪄	●	😐	andriy	2 hours
uniconvertor-1.1.2-1.fc9	Fedora 9	pending	🪄	●	😐	andriy	3 hours
python-alsa-1.0.16-1.fc9	Fedora 9	pending	🪄	●	😐	andriy	3 hours

Latest Comments [more]

Update	Comment	From	Karma
iproute-2.6.25-1.fc9	It did in fact fix bug 444520 for me. I did see some odd things with connection tracking until I restarted iptables which reloaded a connection tracking module. So if you install this you may need to do that if you don't reboot. Since I restarted iptables everything seems to be fine.	bruno	😇
netembryo-0.0.4-1.fc7	erer	Anonymous Tester	😐
	Lately F7 updates have been appearing in repos many days		

All dependencies must be in EPEL

- Be a co-maintainer, or
- Convince someone else to do it, or
- Best – create a community around the packages
- Non-dependency add-on packages can be external to EPEL
  - For example, a game engine with proprietary game levels in an outside repo

EPEL packages do not conflict with RHEL

- Some packages are taken up into RHEL updates
- Natural evolution

Do packages need to be in Fedora to be in EPEL?

- No, although highly recommended
- Just specify EPEL as the sole build repo

EPEL is downstream agnostic

- RHEL, CentOS, others are downstream
- Similar faces

Fedorans are not assigned to work with ISVs

- Community project, people work on what they are passionate about
- Right now you can get attention from those of us who care about your problems (Spot, Greg, Paul, Stephen, Karsten, etc.):
  - Mentor
  - Enable
  - Knock down walls
  - Advocate

How does software make it in to EPEL?

- Tools and community knowledge to make you successful
- Not specialized knowledge, just a reworking of what you learn in other FLOSS projects
- For example: libwidget
  - Begin with the Packaging guidelines:
  - http://fedoraproject.org/wiki/Packaging/Guidelines

How does software make it in to EPEL? (2)

0. You need to be a user in the Fedora account system (FAS), which includes tracking that you have agreed to the Contributor License Agreement (CLA). You also need a bugzilla account for bugzilla.redhat.com

How does software make it in to EPEL? (3)

1. Read the guidelines. They are very thorough. Also understand that you are not likely to get everything correct from the very start. Some of the complexity of turning an interworking set of RPMs in to a distro cannot be conveyed easily in a wiki page.

How does software make it in to EPEL? (4)

2. Put together the first version of your package. Make sure that it builds from source, with all dependencies in Fedora and/or EPEL.

How does software make it in to EPEL? (5)

3. Submit the package for review, as per the process covered in [[Packaging/Guidelines]]. This is a good place where the SIG you are working with can help find other community members to review the package(s). Or you can offer to trade reviews with another SIG member. A newly started SIG that might be perfect for you is the ISV SIG.

   http://fedoraproject.org/wiki/SIGs/ISV

   Another helpful URL:

   http://fedoraproject.org/wiki/PackageMaintainers

How does software make it in to EPEL? (6)

4. Iterate through the package review process until complete.

What is going to catch you?

- Spot's talk from yesterday plus video are coming to the Fedora wiki + a link from the ISV SIG page
- Specific examples to discuss here
  - Bundled binaries of other open source packages
  - Compatibility packages for older versions of software you require

Bundled binaries of other open source software

- Not sustainable over the medium or long term
- You have to keep track of N binaries
  - In Fedora, packages get a wider audience of contributors, testers, and users
- You have to make sure those binaries work with your application ... and the underlying OS
  - In Fedora, you gain the collective testing systems of the Fedora community; what you add to the pool is returned many-fold
- You have to maintain a derived package in your codebase, where no one else can benefit ... or help
  - In Fedora, the burden is shared wider as the community grows

Compat packages

- Compat packages, such as to get older versions of a language (Perl, Python, PHP) are acceptible
- Not encouraged
- You likely have to maintain the package yourself

Why do all this work?

- Initial work == front load effort that pays off many times over the years

Questions, references

- Questions?
- Upstairs in 302 for FUDCon
- fedoraproject.org/wiki/SIGs/ISV
- quaid.fedorapeople.org/presentations/Red-Hat-Summit-2008

RED HAT :: BOSTON :: 2008

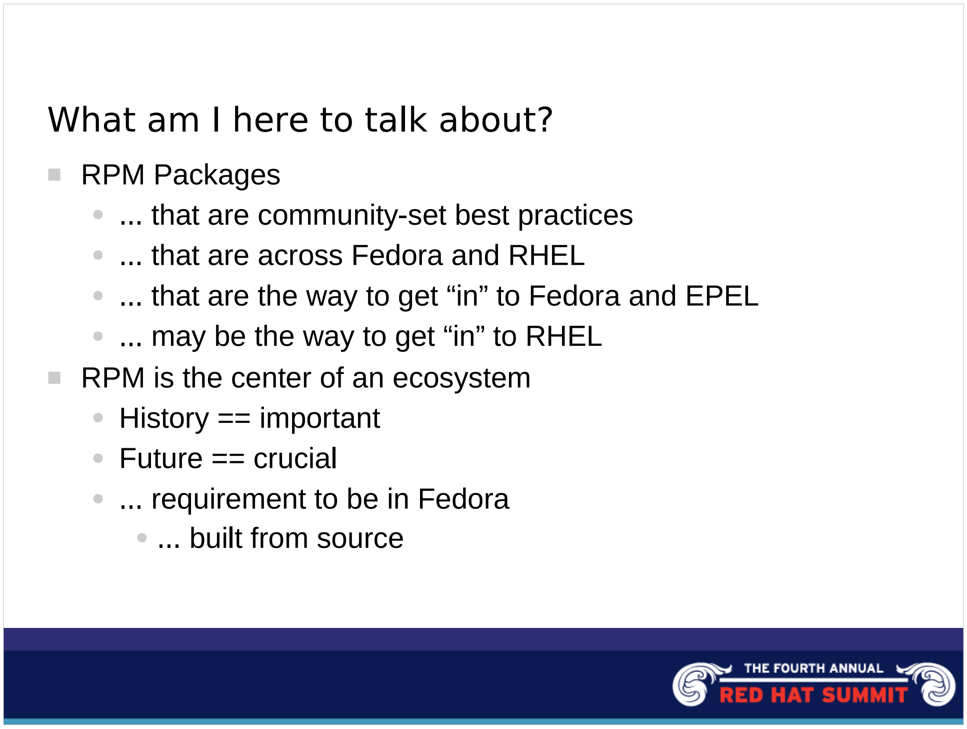
SUMMIT

JUNE 18–20, 2008

Karsten Wade

Sr. Developer Community Mgr, Red Hat

Fedora Packages for Red Hat Enterprise Linux



What am I here to talk about?

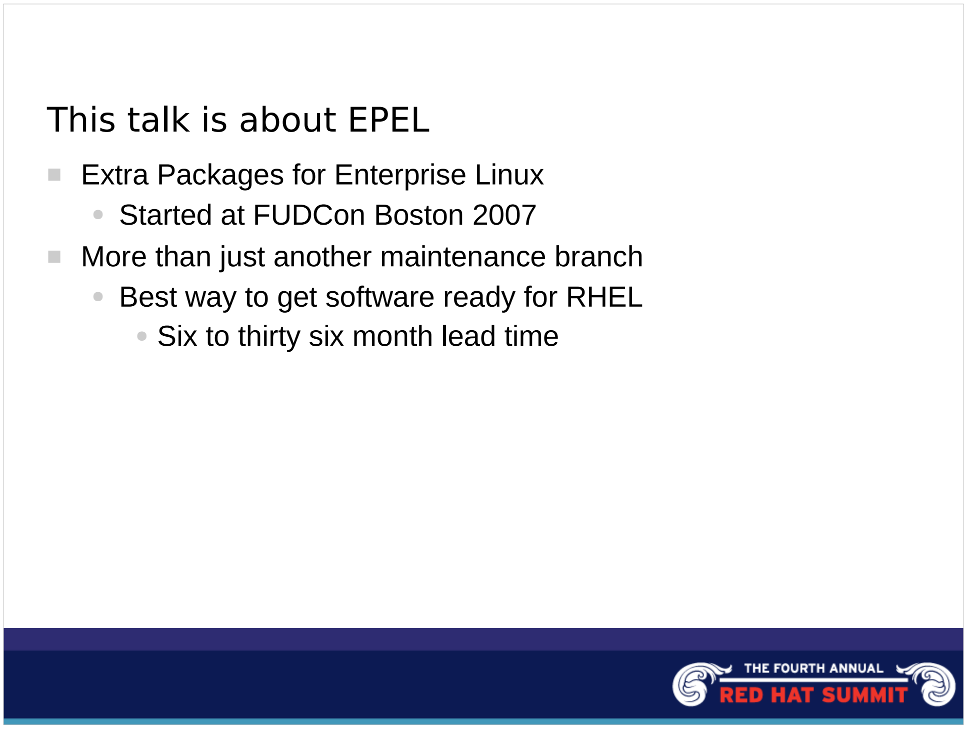
- RPM Packages
  - ... that are community-set best practices
  - ... that are across Fedora and RHEL
  - ... that are the way to get “in” to Fedora and EPEL
  - ... may be the way to get “in” to RHEL
- RPM is the center of an ecosystem
  - History == important
  - Future == crucial
  - ... requirement to be in Fedora
    - ... built from source

THE FOURTH ANNUAL RED HAT SUMMIT

What do I want to talk about?

Fedora packages, that's mainly it. RPM packages, built to a standard that has evolved as a set of best practices across Fedora and Red Hat. These are community-set standards, standards that have filtered up in to Red Hat engineering.

This package, with all its glory and flaws, is the center of an ecosystem started by Red Hat many years ago. The RPM may not represent the history of all software that runs on top of an RPM-based distro, but it definitely represents the future. It is a requirement for being in Fedora and the additional repository I'm here to talk about today, Extra Packages for Enterprise Linux, or EPEL (ee-pul). Your software must build entirely from source, in Fedora or EPEL, using components and fulfilling dependencies entirely from the Fedora/EPEL package set. More about the impact of this later.



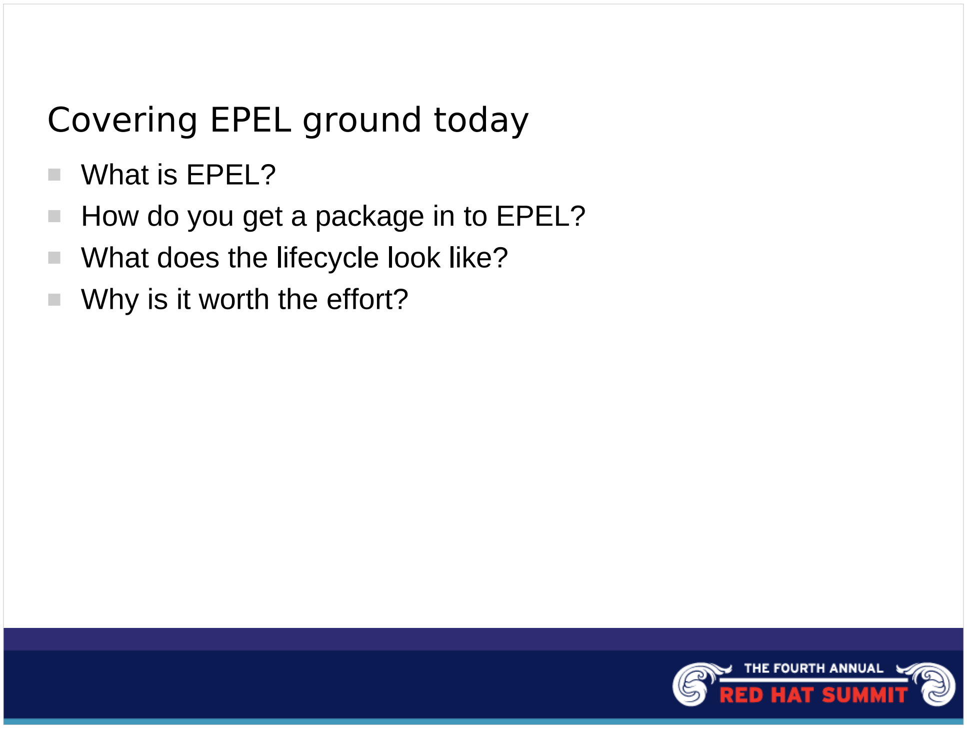
This talk is about EPEL

- Extra Packages for Enterprise Linux
  - Started at FUDCon Boston 2007
- More than just another maintenance branch
  - Best way to get software ready for RHEL
    - Six to thirty six month lead time

THE FOURTH ANNUAL RED HAT SUMMIT

This talk is about EPEL, which is an effort began 18 months ago at FUDCon Boston just up the road at Boston University, and born of years of desire in the RHEL and derivatives communities.

EPEL packages are more than a maintained branch of software from the Fedora package set. EPEL is more than a way to get software from Fedora that Red Hat didn't include in RHEL or an update. EPEL is a pathway to get software ready-for-RHEL, with a lead time of twelve months to three years.

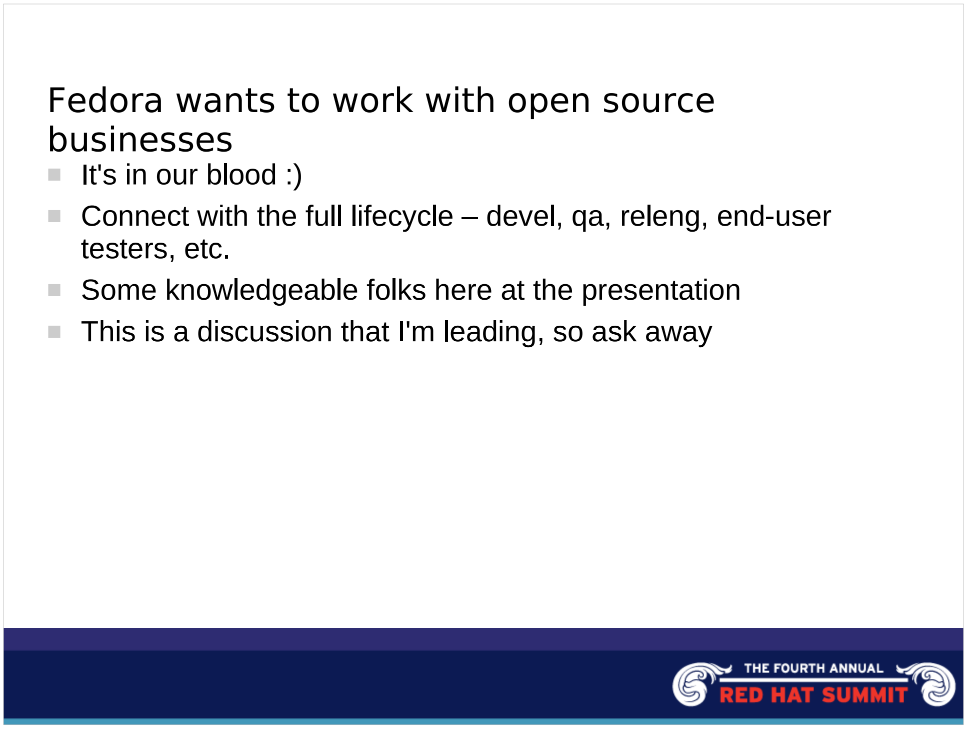


Covering EPEL ground today

- What is EPEL?
- How do you get a package in to EPEL?
- What does the lifecycle look like?
- Why is it worth the effort?

THE FOURTH ANNUAL
RED HAT SUMMIT

This talk is going to cover a lot of ground of EPEL, and only at a fairly high level. What is EPEL, how do you get a package in to EPEL, what that lifecycle looks lie, and why it's worth the effort.

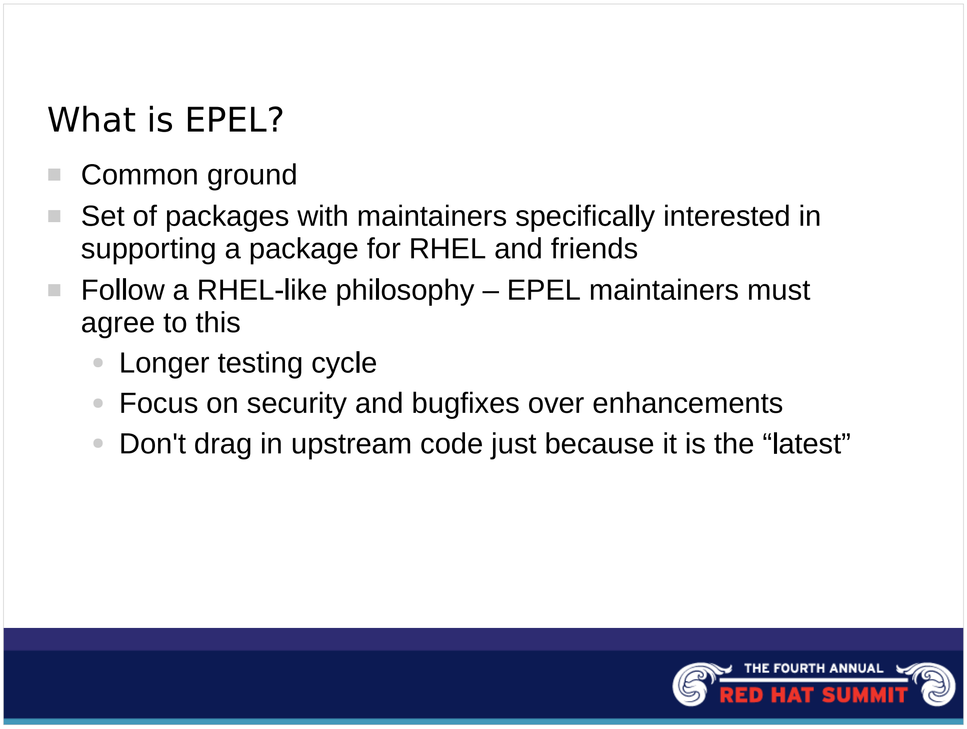


Fedora wants to work with open source businesses

- It's in our blood :)
- Connect with the full lifecycle – devel, qa, releng, end-user testers, etc.
- Some knowledgeable folks here at the presentation
- This is a discussion that I'm leading, so ask away

THE FOURTH ANNUAL
RED HAT SUMMIT

We are trying to attract business interest, to connect with software product managers, developers, QA, release engineers, the whole software lifecycle. To some degree, this talk is a clearing house to point you in the next direction. I hope to raise more questions in the opening than I answer, thus I'll try to be brief and leave plenty of time for the questions. I've invited knowledgeable folks from Red Hat and Fedora to be here, so if you have questions that are technical, business, community, or whatever, ask away at any time. I don't hold to making you wait until the end, this is a discussion with me in the front of the room.

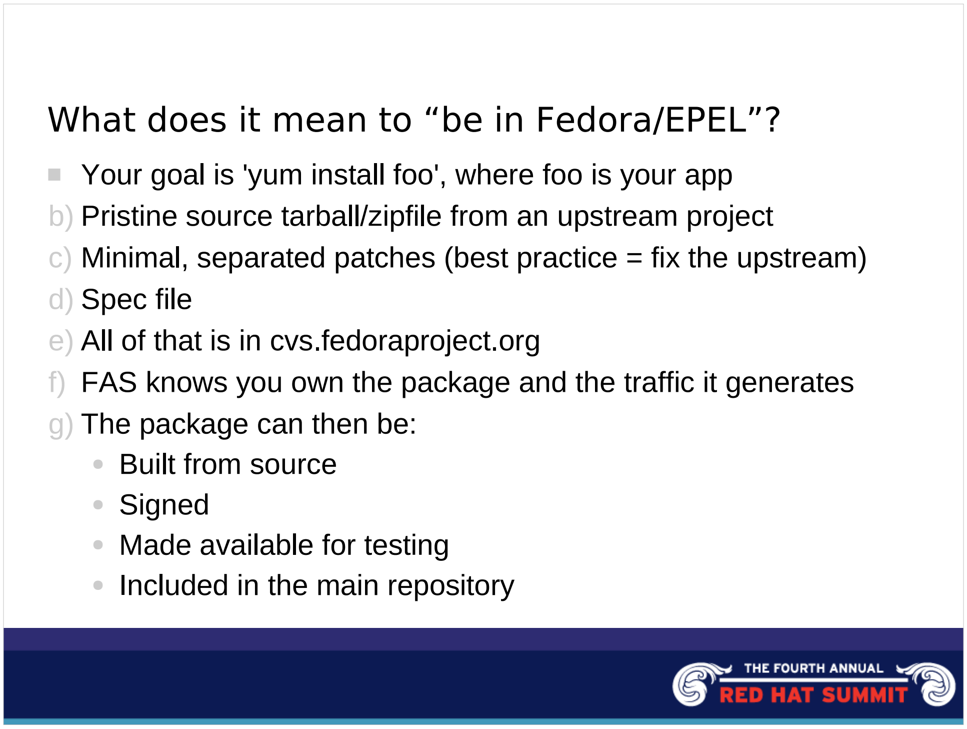


What is EPEL?

- Common ground
- Set of packages with maintainers specifically interested in supporting a package for RHEL and friends
- Follow a RHEL-like philosophy – EPEL maintainers must agree to this
  - Longer testing cycle
  - Focus on security and bugfixes over enhancements
  - Don't drag in upstream code just because it is the “latest”

THE FOURTH ANNUAL
RED HAT SUMMIT

To establish some common ground, what is EPEL? It is a set of packages that have maintainers who are specifically interested in supporting a package for RHEL and friends. (More about those friends in a minute.) EPEL packages follow a RHEL-like philosophy, and do not follow the same pace that packages do in the Fedora distro. EPEL maintainers have to agree to follow this philosophy. The philosophy includes a longer testing cycle, focus on security and bugfix updates over enhancements, and new versions of code from upstream are not shoved in to an update of the package without regard to how that impacts the rest of the repo.



What does it mean to “be in Fedora/EPEL”?

- Your goal is 'yum install foo', where foo is your app

b) Pristine source tarball/zipfile from an upstream project

c) Minimal, separated patches (best practice = fix the upstream)

d) Spec file

e) All of that is in cvs.fedoraproject.org

f) FAS knows you own the package and the traffic it generates

g) The package can then be:
   - Built from source
   - Signed
   - Made available for testing
   - Included in the main repository

THE FOURTH ANNUAL
RED HAT SUMMIT

What does it mean to "be in Fedora" or "be in EPEL"? This will be more clear as I continue, but it essentially means that you have:

a. pristine source tarball/zipfile from an upstream project

b. minimal, separated patches

   - best practice is to fix the upstream

c. spec file

d. all of that is in cvs.fedoraproject.org

e. FAS knows you own the package and all the traffic it generates

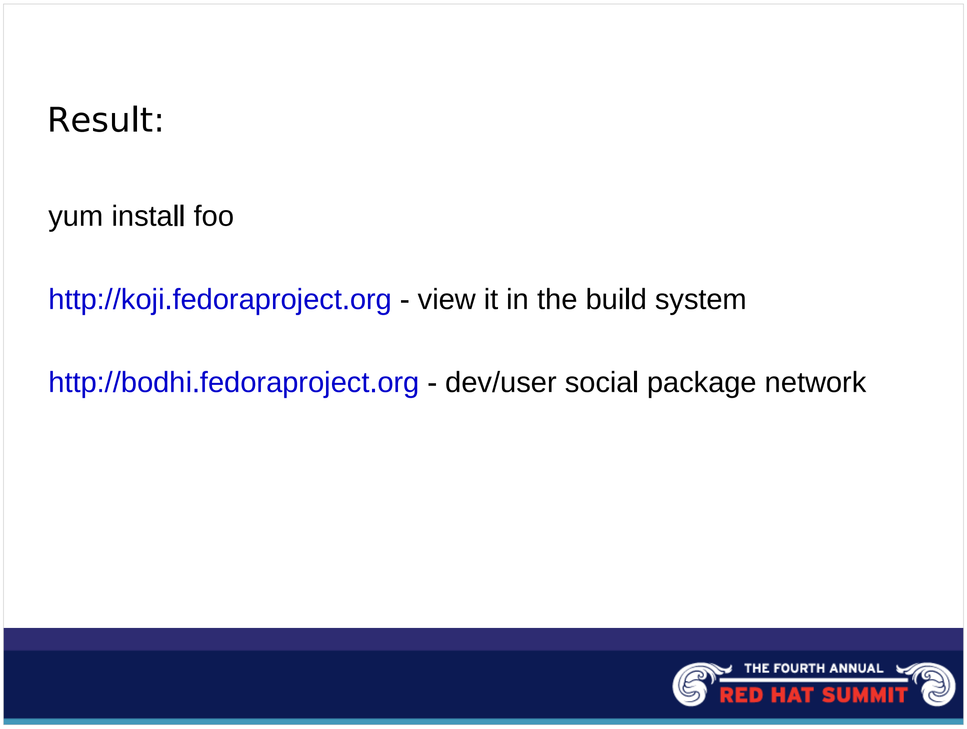
f. the package can then be:

   * built from source

   * signed

   * made available for testing

   * then included in the main repository



Result:

yum install foo

http://koji.fedoraproject.org - view it in the build system

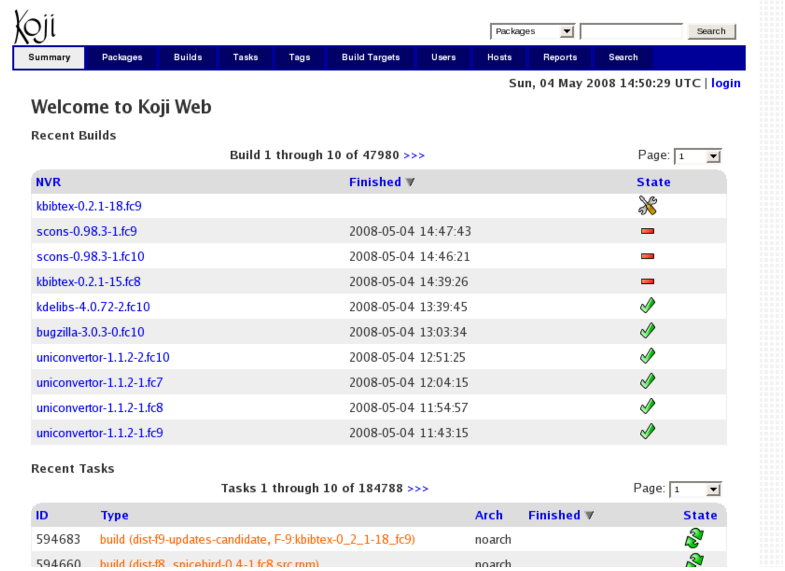
http://bodhi.fedoraproject.org - dev/user social package network

THE FOURTH ANNUAL
RED HAT SUMMIT

Result? 'yum install foo'

http://koji.fedoraproject.org -- view in to build system

http://bodhi.fedoraproject.org -- dev/user social package network



Sun, 04 May 2008 14:50:29 UTC | login

Welcome to Koji Web

Recent Builds

Build 1 through 10 of 47980 >>>   Page: 1

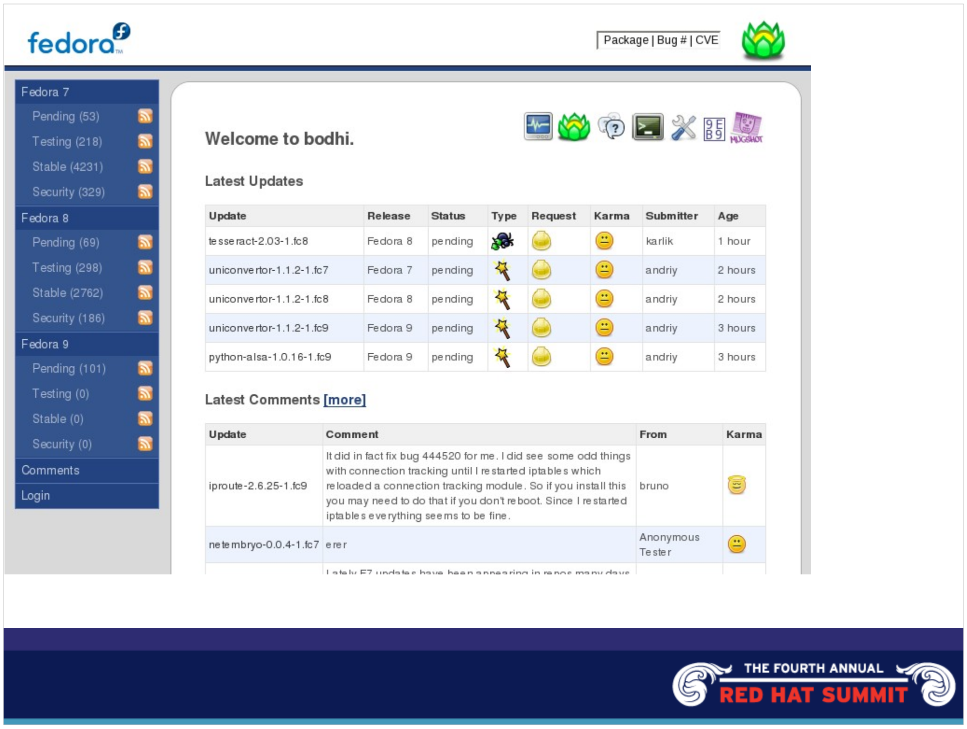
NVR	Finished ▼	State
kbibtex-0.2.1-18.fc9		🛠
scons-0.98.3-1.fc9	2008-05-04 14:47:43	▬
scons-0.98.3-1.fc10	2008-05-04 14:46:21	▬
kbibtex-0.2.1-15.fc8	2008-05-04 14:39:26	▬
kdelibs-4.0.72-2.fc10	2008-05-04 13:39:45	✓
bugzilla-3.0.3-0.fc10	2008-05-04 13:03:34	✓
uniconvertor-1.1.2-2.fc10	2008-05-04 12:51:25	✓
uniconvertor-1.1.2-1.fc7	2008-05-04 12:04:15	✓
uniconvertor-1.1.2-1.fc8	2008-05-04 11:54:57	✓
uniconvertor-1.1.2-1.fc9	2008-05-04 11:43:15	✓

Recent Tasks

Tasks 1 through 10 of 184788 >>>   Page: 1

ID	Type	Arch	Finished ▼	State
594683	build (dist-f9-updates-candidate, F-9:kbibtex-0_2_1-18_fc9)	noarch		🔄
594660	build (dist-f8, spicebird-0.4-1.fc8.src.rpm)	noarch		🔄

http://koji.fedoraproject.org -- view in to build system



Welcome to bodhi.

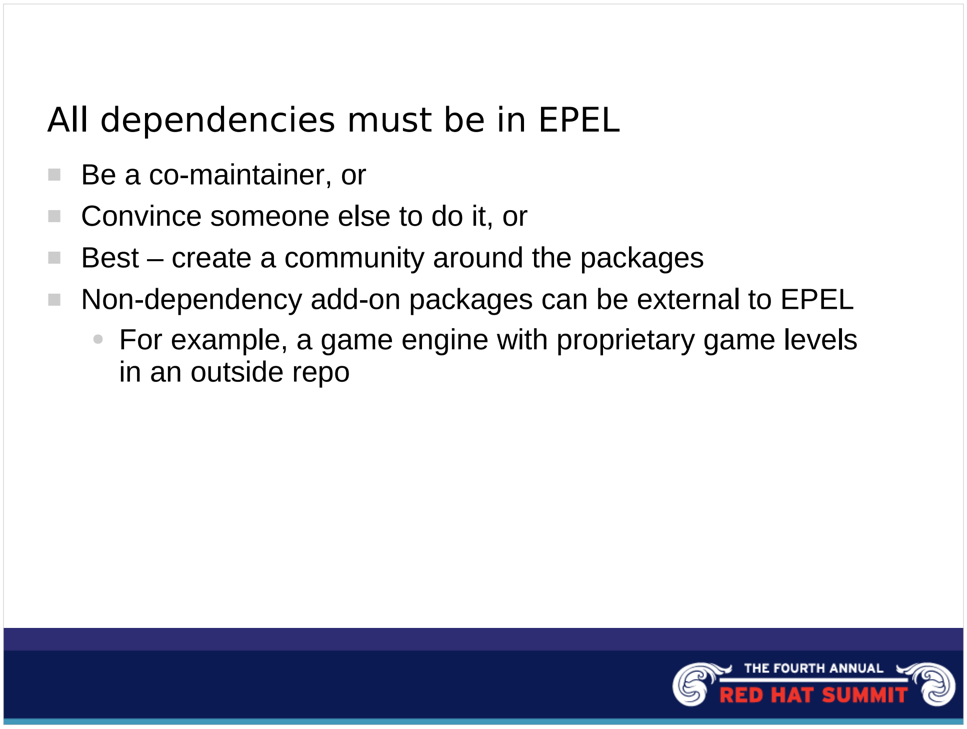
Latest Updates

Update	Release	Status	Type	Request	Karma	Submitter	Age
tesseract-2.03-1.fc8	Fedora 8	pending	🐞	⛑	😐	karlik	1 hour
uniconvertor-1.1.2-1.fc7	Fedora 7	pending	🪄	⛑	😐	andriy	2 hours
uniconvertor-1.1.2-1.fc8	Fedora 8	pending	🪄	⛑	😐	andriy	2 hours
uniconvertor-1.1.2-1.fc9	Fedora 9	pending	🪄	⛑	😐	andriy	3 hours
python-alsa-1.0.16-1.fc9	Fedora 9	pending	🪄	⛑	😐	andriy	3 hours

Latest Comments [more]

Update	Comment	From	Karma
iproute-2.6.25-1.fc9	It did in fact fix bug 444520 for me. I did see some odd things with connection tracking until I restarted iptables which reloaded a connection tracking module. So if you install this you may need to do that if you don't reboot. Since I restarted iptables everything seems to be fine.	bruno	😇
netembryo-0.0.4-1.fc7	erer	Anonymous Tester	😐
	Lately F7 updates have been appearing in repos many days		

http://bodhi.fedoraproject.org -- dev/user social package network



All dependencies must be in EPEL

- Be a co-maintainer, or
- Convince someone else to do it, or
- Best – create a community around the packages
- Non-dependency add-on packages can be external to EPEL
  - For example, a game engine with proprietary game levels in an outside repo

THE FOURTH ANNUAL RED HAT SUMMIT

All of the packages that an EPEL package requires, called dependencies, must be in EPEL. If they do not exist already, there are several options. One is to work with the maintainer of the Fedora package as a co-maintainer, specifically filling the role of EPEL maintainer. You might be able to convince that person or someone else to maintain the dependency packages in EPEL. An example of a great thing is if you find a few other people interested in the common packages, and you partner with one or more of them to be co-maintainers. In other words, you establish a small but focused Fedora/EPEL community around a set of packages.

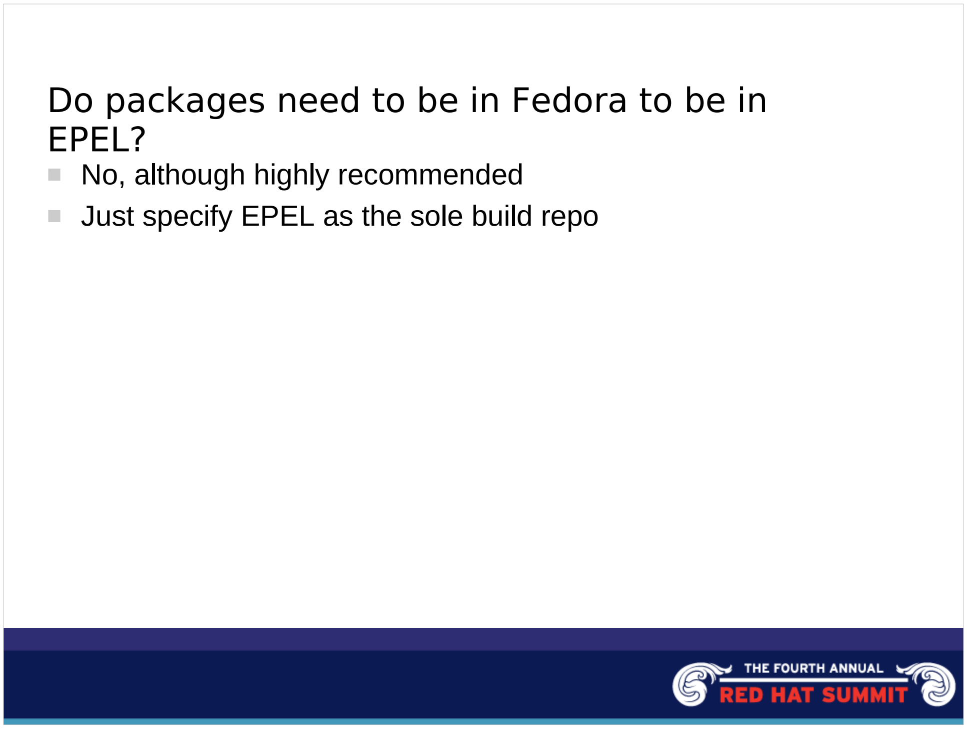
Note: add-on packages that are not dependencies can exist outside of EPEL, since they are not needed to build the package from source or to support package installation. An example of this is an open source game engine and a set of game levels that are not free and open content. The game engine can be built and installed by itself without any levels. The levels make for a nicer playing experience, but are not required, and could be in an external repository.

EPEL packages do not conflict with RHEL

- Some packages are taken up into RHEL updates
- Natural evolution

THE FOURTH ANNUAL
RED HAT SUMMIT

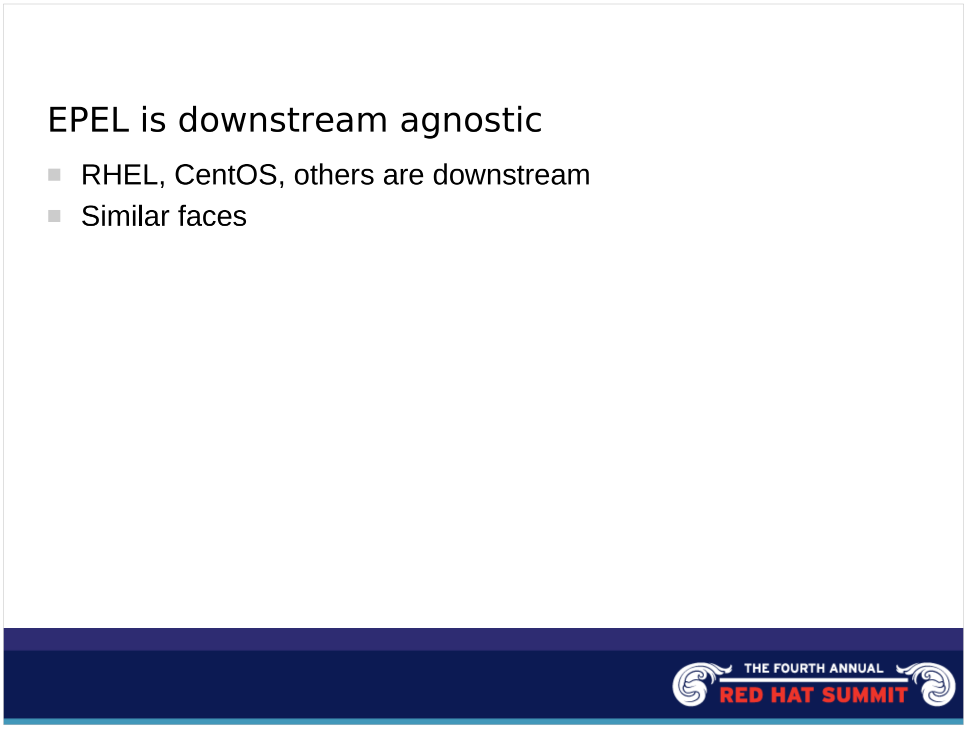
EPEL packages do not conflict with packages from RHEL. This includes packages that might be in a RHEL add-on offering. The RHEL team can bring any packages in for a RHEL update that have been in EPEL; this is inevitable in some cases, a natural part of the evolution of customer need. We haven't been tracking statistics yet, but over time it will be interesting to see how much EPEL popularity affects customer demand.



Do packages need to be in Fedora to be in EPEL?

- No, although highly recommended
- Just specify EPEL as the sole build repo

Does a package need to be in the main Fedora repository before or alongside EPEL? Actually, no. It is highly recommended for various reasons, but it is not required. You follow the same package acceptance process, and you specify EPEL as the sole build repo.

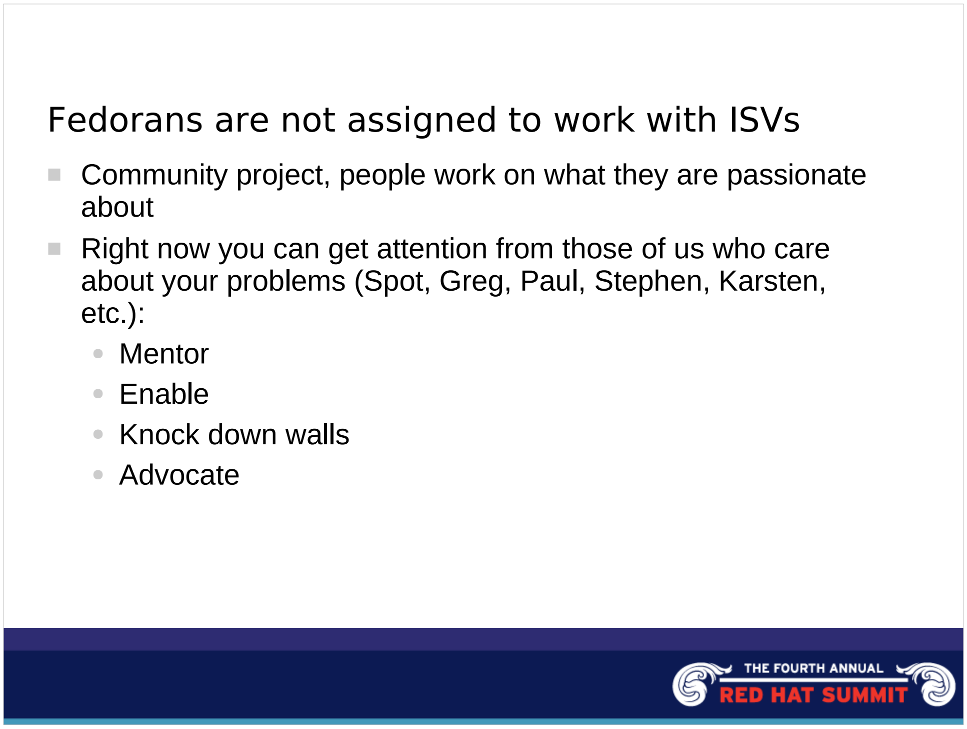


EPEL is downstream agnostic

- RHEL, CentOS, others are downstream
- Similar faces

THE FOURTH ANNUAL
RED HAT SUMMIT

As an upstream, EPEL is downstream agnostic. In Fedora, we consider RHEL-derivatives such as CentOS to be downstream of our work, and as such are explicit consumers of EPEL packages. As always, you'll see some of the same faces in the EPEL community that are also in the RHEL and friend communities.

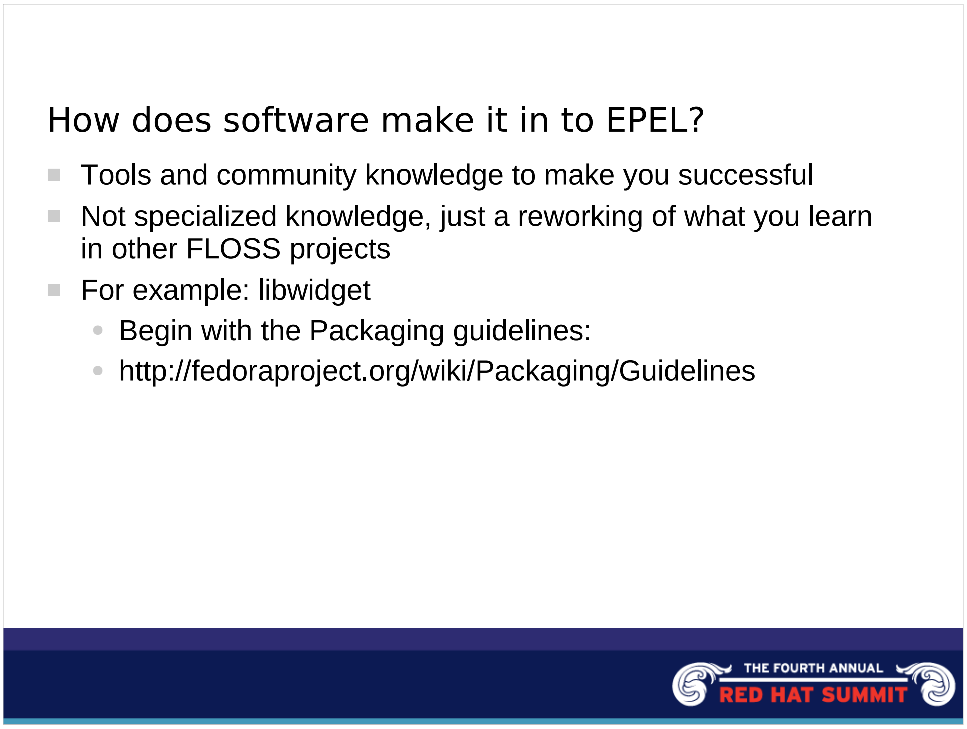


Fedorans are not assigned to work with ISVs

- Community project, people work on what they are passionate about
- Right now you can get attention from those of us who care about your problems (Spot, Greg, Paul, Stephen, Karsten, etc.):
  - Mentor
  - Enable
  - Knock down walls
  - Advocate

THE FOURTH ANNUAL RED HAT SUMMIT

As a community project, there really aren't any people in Fedora who are assigned specifically to working with ISVs. However, there are a number of folks, including myself, Tom Callaway, Greg DeKoenigsberg, Paul Frields, Steve Smoogen, and others yet unidentified who are willing mentors, enablers, wall-knocker-downers, and advocates. For our various reasons, we are here now helping. Get it while it's hot, because this level of attention won't stick around.



How does software make it in to EPEL?

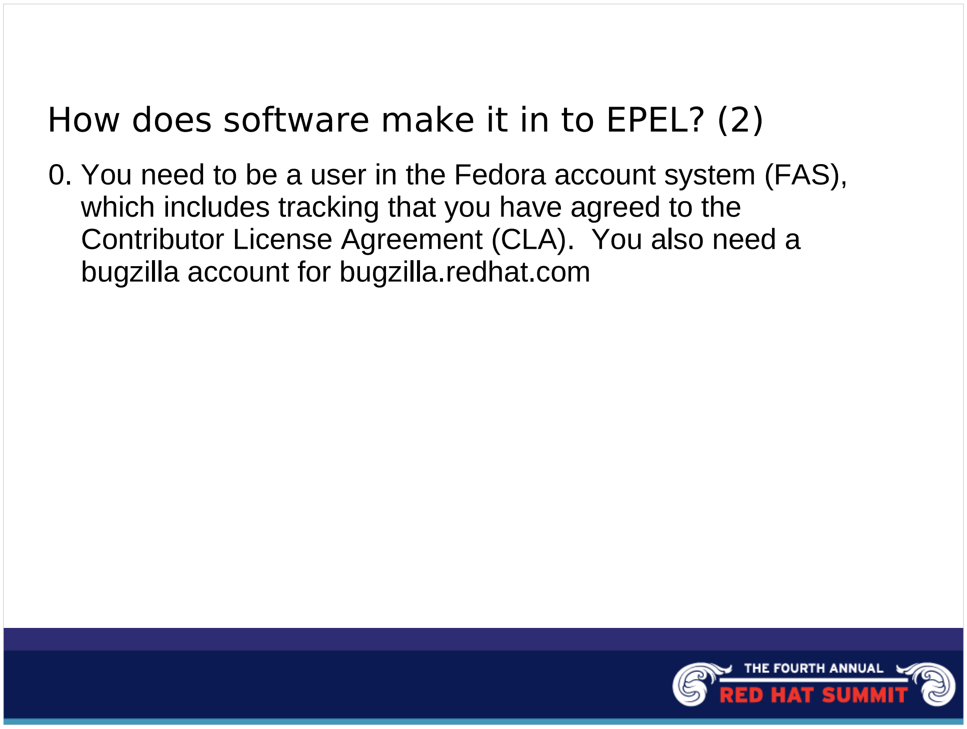
- Tools and community knowledge to make you successful
- Not specialized knowledge, just a reworking of what you learn in other FLOSS projects
- For example: libwidget
  - Begin with the Packaging guidelines:
  - http://fedoraproject.org/wiki/Packaging/Guidelines

THE FOURTH ANNUAL
RED HAT SUMMIT

How does software make it in to EPEL? For those of us who are used to the processes in Fedora, it's the norm. But there are tools and community knowledge that you need to have or obtain in order to be successful. Fortunately, it's not specialized knowledge, it's the same stuff you learn in other FLOSS projects, just sorted differently.

Let's talk about a small library your application needs, libwidget. Your journey with packaging libwidget begins in the packaging area of the Fedora wiki:

http://fedoraproject.org/wiki/Packaging/Guidelines

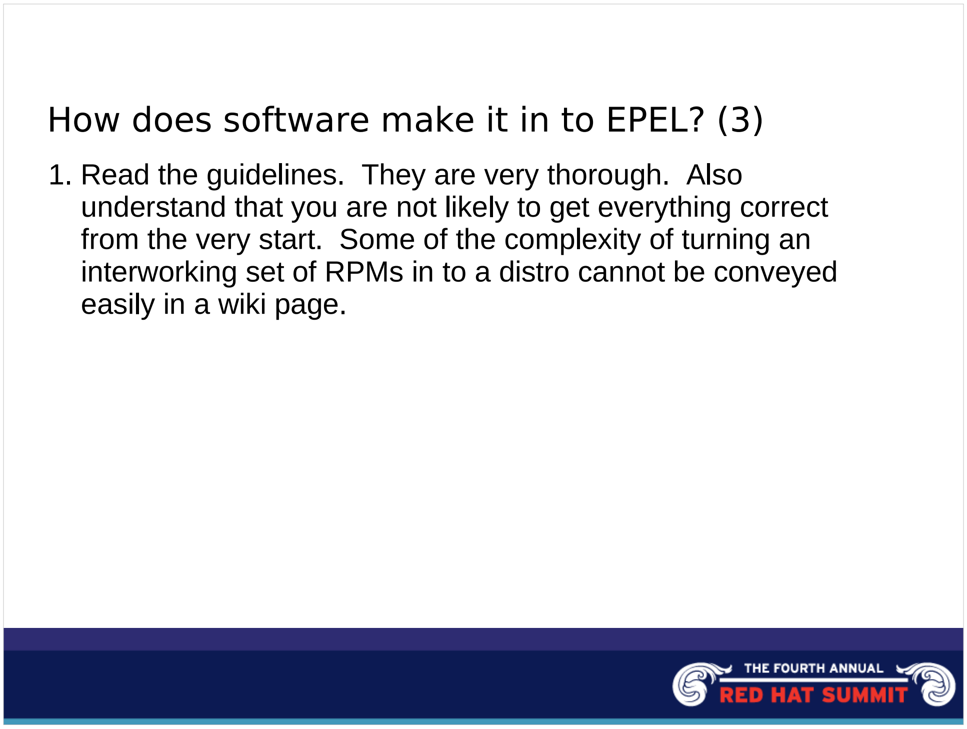


How does software make it in to EPEL? (2)

0. You need to be a user in the Fedora account system (FAS), which includes tracking that you have agreed to the Contributor License Agreement (CLA). You also need a bugzilla account for bugzilla.redhat.com

THE FOURTH ANNUAL
RED HAT SUMMIT

0. You need to be a user in the Fedora account system (FAS), which includes tracking that you have agreed to the Contributor License Agreement.

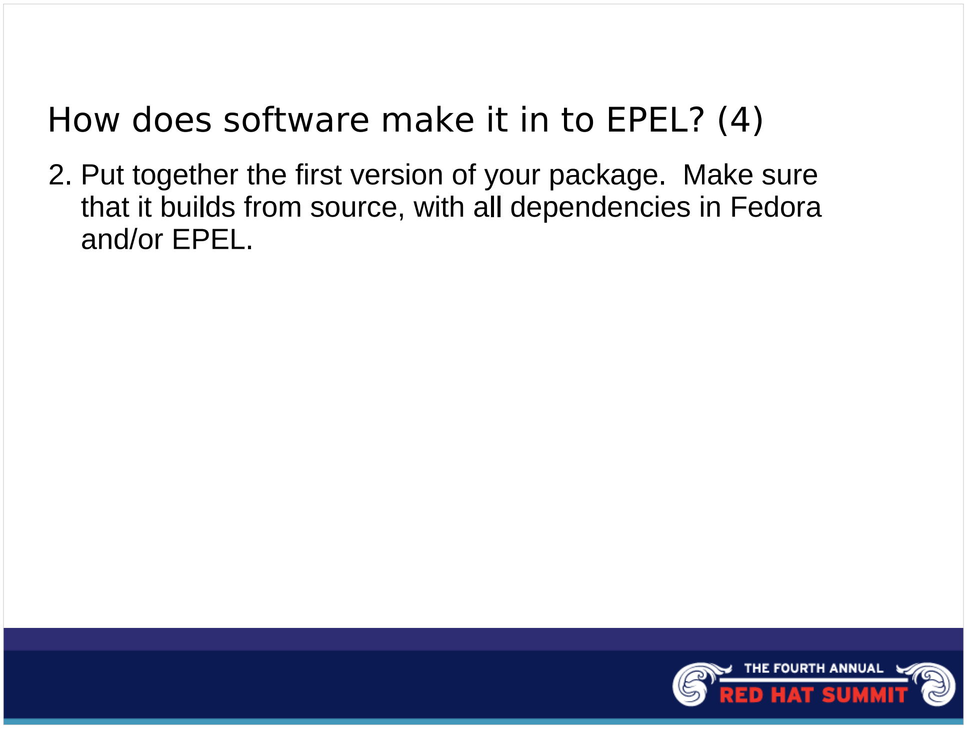


How does software make it in to EPEL? (3)

1. Read the guidelines. They are very thorough. Also understand that you are not likely to get everything correct from the very start. Some of the complexity of turning an interworking set of RPMs in to a distro cannot be conveyed easily in a wiki page.

THE FOURTH ANNUAL
RED HAT SUMMIT

1. Read the guidelines. They are very thorough. Also understand that you are not likely to get everything right from the start. Some of the complexity of turning an interworking set of RPMs in to a distro cannot be conveyed easily in a wiki page.



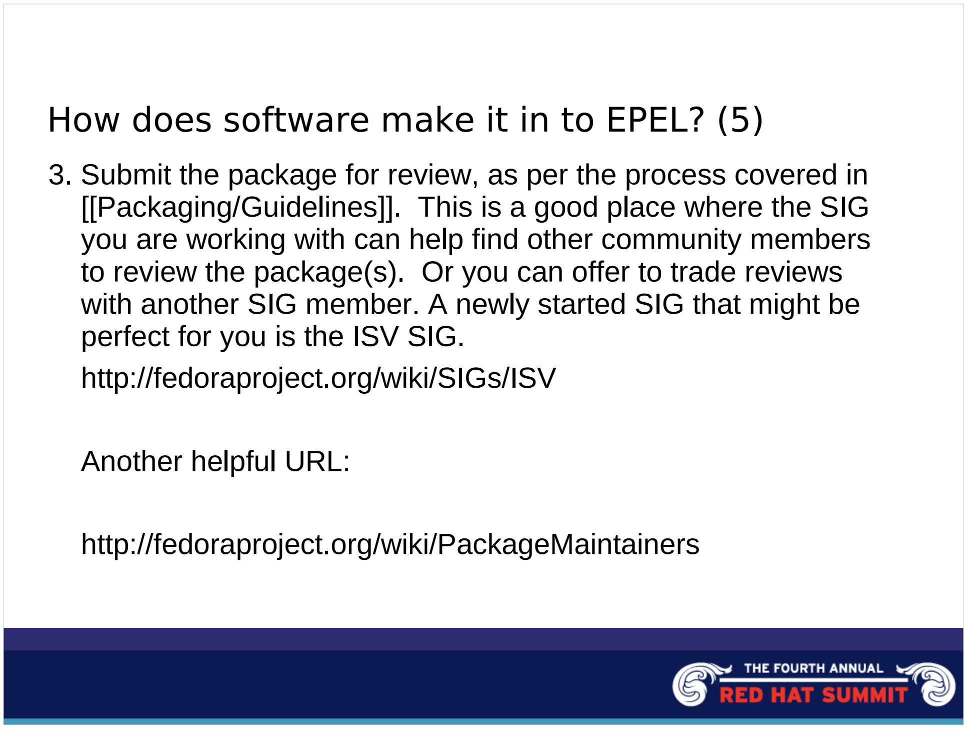
How does software make it in to EPEL? (4)

2. Put together the first version of your package. Make sure that it builds from source, with all dependencies in Fedora and/or EPEL.

THE FOURTH ANNUAL
RED HAT SUMMIT

2. Put together the first version of your package. Make sure that it builds from source, with all dependencies in Fedora.

You may find that one or more of the applications or libraries you depend on are not in Fedora. They must be in Fedora, by hook or by crook. More on that in a minute.



How does software make it in to EPEL? (5)

3. Submit the package for review, as per the process covered in [[Packaging/Guidelines]]. This is a good place where the SIG you are working with can help find other community members to review the package(s). Or you can offer to trade reviews with another SIG member. A newly started SIG that might be perfect for you is the ISV SIG.

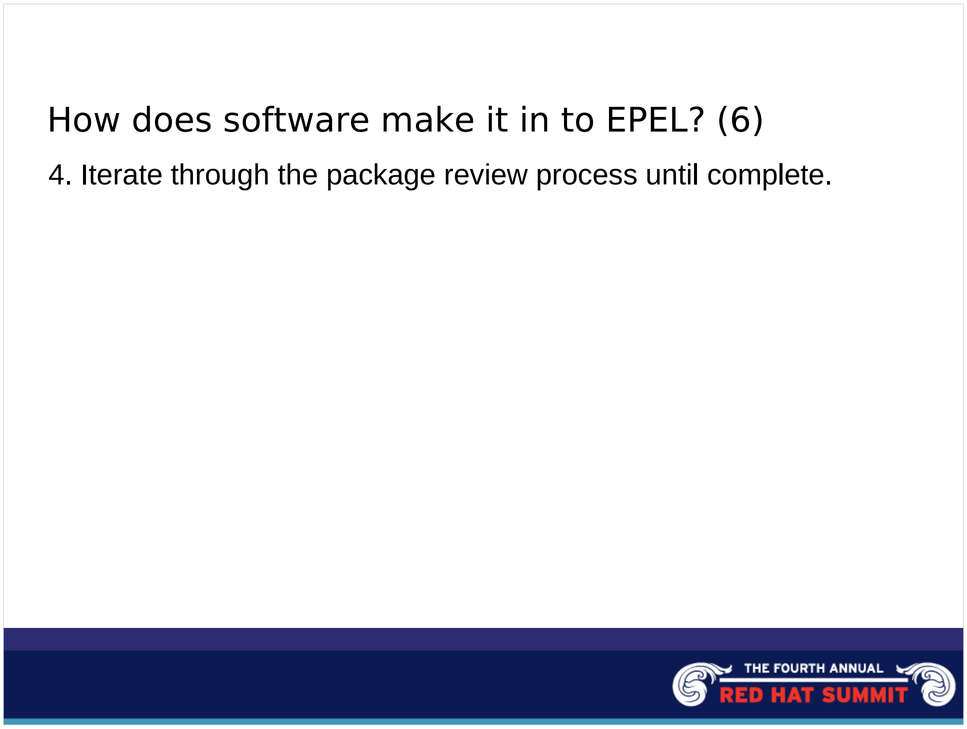
   http://fedoraproject.org/wiki/SIGs/ISV

   Another helpful URL:

   http://fedoraproject.org/wiki/PackageMaintainers

3. Submit the package for review, as per the process covered in [[PackageMaintainers]]. This is a good place where the SIG you are working with can help find other community members to review the package(s). A newly started SIG that might be perfect for you is the ISV SIG.

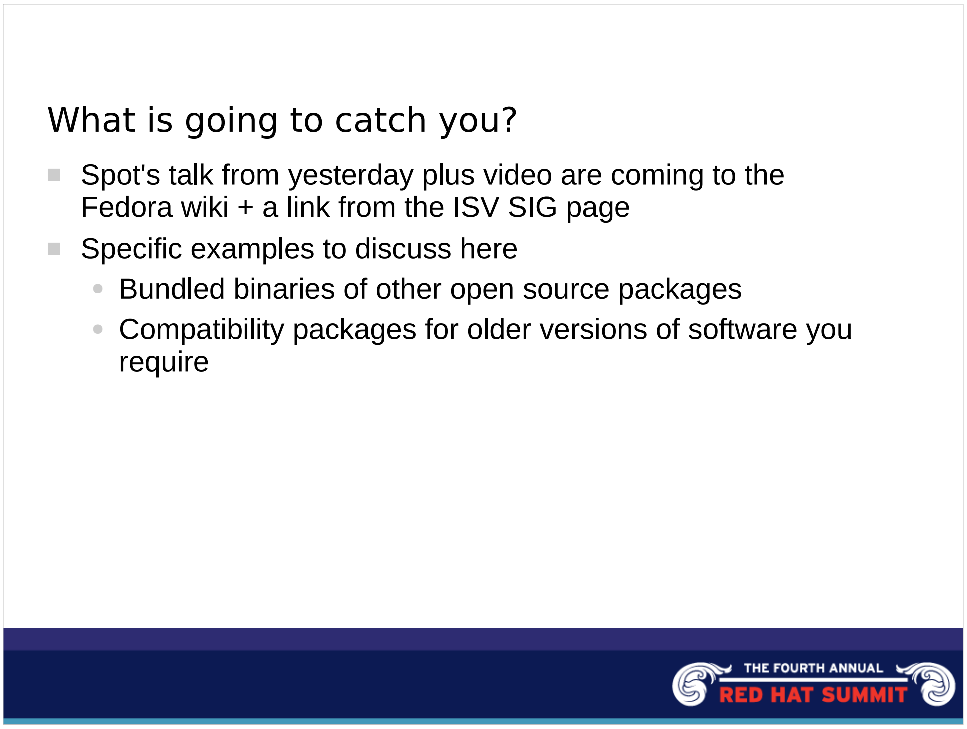
https://fedoraproject.org/wiki/SIGs/ISV



How does software make it in to EPEL? (6)

4. Iterate through the package review process until complete.

4. Iterate through the package review process until complete.

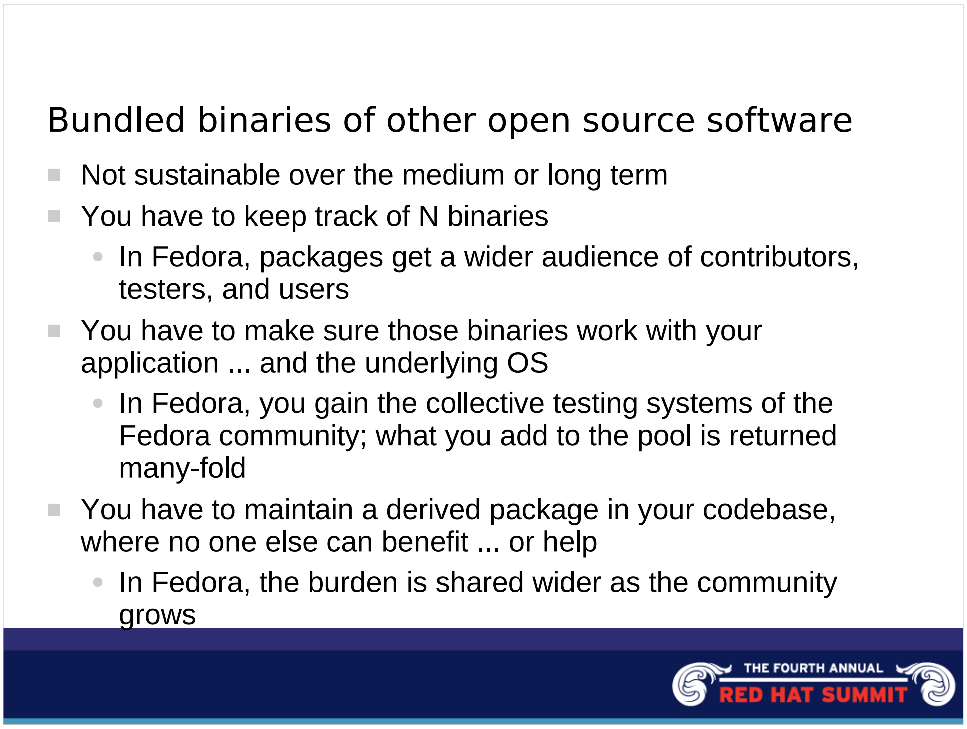


What is going to catch you?

- Spot's talk from yesterday plus video are coming to the Fedora wiki + a link from the ISV SIG page
- Specific examples to discuss here
  - Bundled binaries of other open source packages
  - Compatibility packages for older versions of software you require

THE FOURTH ANNUAL
RED HAT SUMMIT

What is going to catch you? All manner of stuff. Yesterday Spot gave a talk on how to make good RPM packages. He talked in some detail about common mistakes. I got a video tape of that, which I'll wrap up with his ultra-commented spec file and presentation, and add a link to that bundle from the ISV SIG page.



Bundled binaries of other open source software

- Not sustainable over the medium or long term
- You have to keep track of N binaries
  - In Fedora, packages get a wider audience of contributors, testers, and users
- You have to make sure those binaries work with your application ... and the underlying OS
  - In Fedora, you gain the collective testing systems of the Fedora community; what you add to the pool is returned many-fold
- You have to maintain a derived package in your codebase, where no one else can benefit ... or help
  - In Fedora, the burden is shared wider as the community grows

One that we keep hearing about is the situation with including binaries of open source software built elsewhere. This is a typical practice in many projects, in particular in Java applications. While the development methodology can be argued separately, from a system maintenance stand point, this is a challenging nightmare. Ultimately, from any view, it is not a long-term and tenable position. Why?

* You have to keep track of N binaries

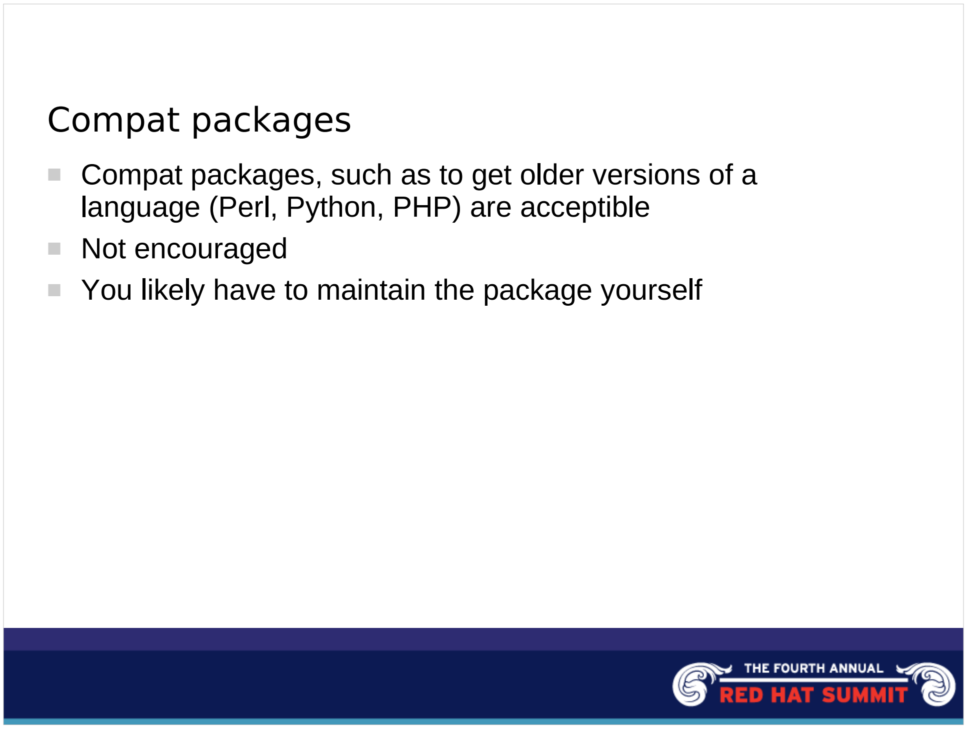
  - In Fedora, packages get a wider audience of contributors, testers, and users

* You have to make sure those binaries work with your application ... and the underlying OS

  - In Fedora, you gain the collective testing systems of the Fedora community; what you add to that pool, is returned many-fold.

* You have to maintain a derived package in your codebase, where no one else can benefit or help

  - In Fedora, the burden is shared wider as the community increases

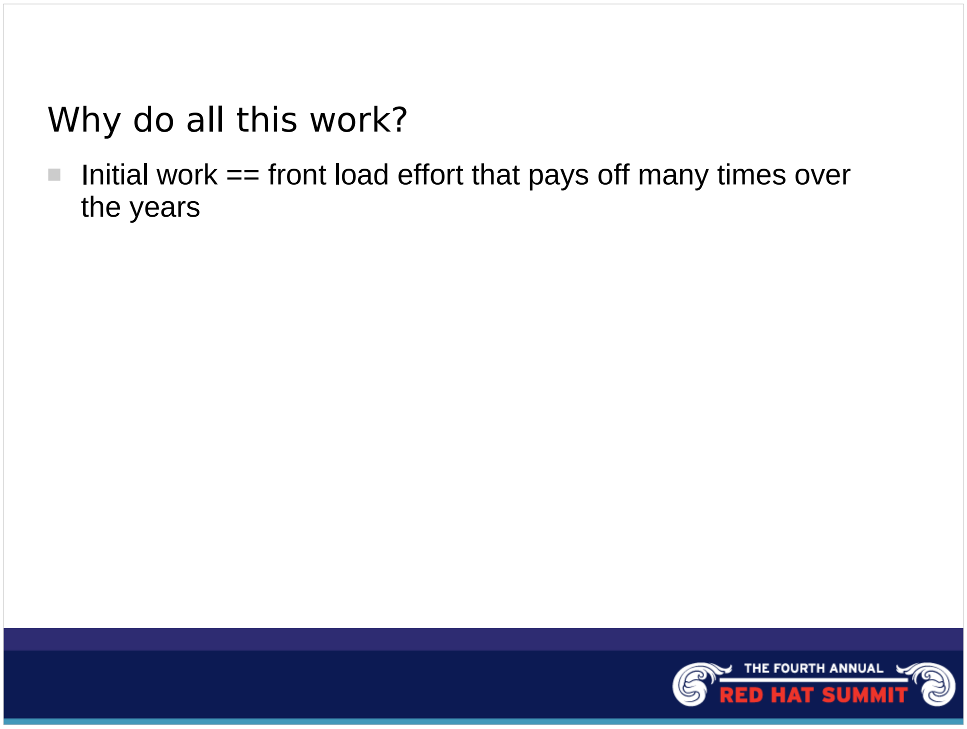


Compat packages

- Compat packages, such as to get older versions of a language (Perl, Python, PHP) are acceptible
- Not encouraged
- You likely have to maintain the package yourself

THE FOURTH ANNUAL
RED HAT SUMMIT

Another situation is when an older version of a package is needed to make your software work. It is possible to maintain a compatibility package (compat package) that is the older version.

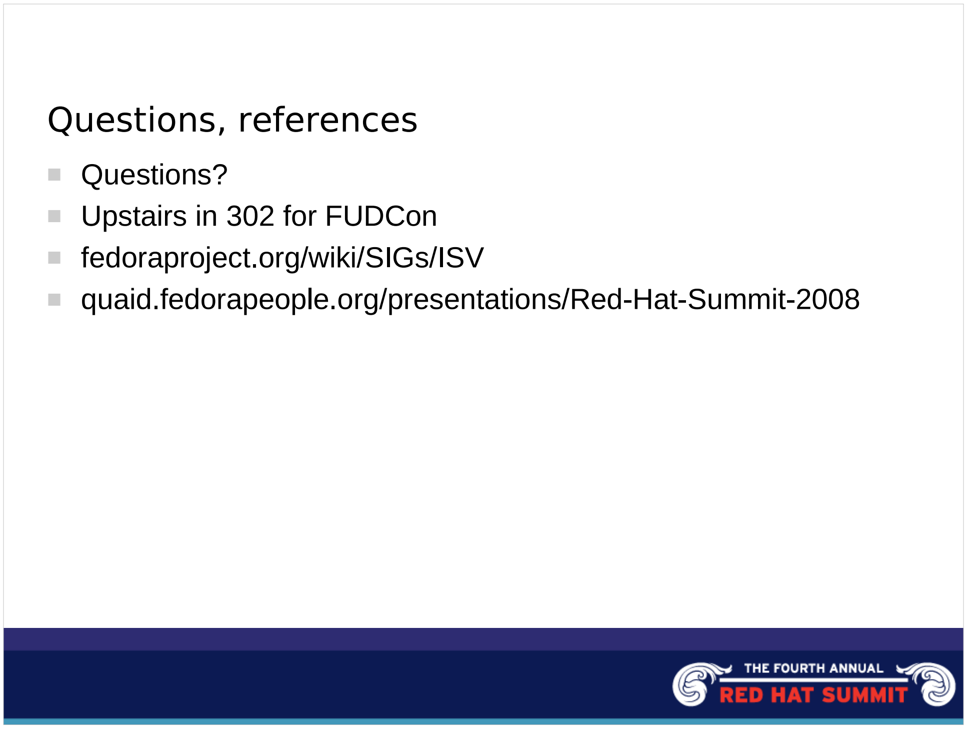


Why do all this work?

- Initial work == front load effort that pays off many times over the years

THE FOURTH ANNUAL
RED HAT SUMMIT

The initial work can be a bit more than you know; it might be easier. It front loads effort for a many-fold return later.



Questions, references

- Questions?
- Upstairs in 302 for FUDCon
- fedoraproject.org/wiki/SIGs/ISV
- quaid.fedorapeople.org/presentations/Red-Hat-Summit-2008

THE FOURTH ANNUAL
RED HAT SUMMIT