Modular Building

splitting the build

Agenda

- Why ?
- Similar Projects
- Proposal

Why?

- Output rpms can be independently updated
  - Smaller updates for endusers
  - Less to build, faster to create updates
- Easier to get involved in development
  - Working on smaller self-contained modules is much less intimidating.
  - Not necessary to build and install the entire tree just to work on one small part of it.
  - e.g. With new *Modular X* fixing memleak in libXcursor was a trivial matter

Similar Projects

- X.org “Modular X”
  - Fine-grained split into approx 100!(?) independently buildable packages
- Mozilla
  - Standalone nss & nspr which can be built outside the mozilla full build
- gcj
  - Desires to be buildable outside the gcc tree to be independently updated outside the gcc schedule

Target Scenario

- *Modular X* an ideal goal
  - Potential developer sees typo in writer menu
  - Grabs openoffice.org-sw source package
  - Sees it requires openoffice-foo-devel, openoffice.org-baz-devel etc. to build
  - installs those packages instead of building all sw dependencies
  - Builds sw nice and fast, sw links against -devel libs. Fixes typo.
  - Gets hooked, fixes redlining and adds grammar checker

Initial proposal

- e.g. Mozilla “standalone” nss
- Uno Runtime Environment
  - We already package a -ure rpm
  - Comprised of various base UDK libaries, e.g. sal
  - Our stable foundation, versioned libs, public apis.
  - Ideal candidate selection
- Buildable -URE
- Buildable -core against -URE
- openoffice.org rpms Require -ure rpms

URE requirements

Modules required to build -ure

- common bootstrapping & packaging modules

```
config_office  dmake  solenv  scp2  instsetoo_native
```

- ure output and buildtools

```
bridges        cli_ure     codemaker  cppu
cppuhelper     cpputools   idlc       io
javaunohelper  jurt        jvmaccess  jvmfwk
offapi         offuh       rdbmaker   registry
remotebridges  ridljar     sal        salhelper
soltools       stlport     stoc       store
udkapi         xml2cmp     ure
```

CORE requirements

- Modules required to build everything else
  - common bootstrapping & packaging modules

```
config_office   dmake   solenv   scp2   instsetoo_native
```

  - Remaining modules not in ure
- Build against ure and ure “devel” output

URE/URE-devel

- Deliverables from URE clearly need to be provided to core
- Package headers, idl files, .rdb files and xml2cmp etc.
  - These are our public headers, so no panic about 3rd parties getting the idea of depending on internal apis
  - Possibly add a pkg-config.pc
- Typically bundled up into an ure-devel package

CORE

- Add support for a separate URE
  - config_office option for ure
    - Pkgconfig handy to find ure bin, include, rdb dirs
  - Add ure-devel bin dir, include dir etc to search paths

Gotchas

- Some differences between ure and core varients
  - some config files (e.g. jvmfwkrc)
    - In particular some of the “ure” modules create two config files, one for ure and one for core
    - Currently package them in ure/ure-devel anyway
    - Ideally clean up separation
  - Library locations
    - Program for core, lib for ure
    - Unify, or link
    - Some search mechanisms work on libpath

Bottom Line

- A “system-ure” is workable at the moment
  - Possible Fedora Core 7
- Much easier if agreed that such a thing is desirable upstream
- If so
  - Get normal rpms requiring the ure rpms
  - Unify or fully separate config (unorc/jvmfwrc)
  - Enable building & packaging ure on it’s own
  - Enable building core against a system ure

References

- Other applications
  - http://wiki.x.org/wiki/ModularizationProposal
  - http://www.mozilla.org/projects/security/pki/nss/buildnss_32.html
- Sample implementation
  - http://people.redhat.com/caolanm/systemure
  - Sample .spec for creating modified ure/ure-devel
  - Patch to use systemure
  - Patch to use .exe location, not .lib location for jvmfwkrc