Политика сосуществования библиотек собранных разными компиляторами c++

Ситуация

ОС: МСВС. Имеются gcc-2.95.4, qt-2.3, qt-3.3. Имеется другая версия gcc и набор библиотек (в т.ч. на c++), скомпилированных им. Хочется иметь возможность установить в систему как “родные”, так и сторонние пакеты, так чтобы они не конфликтовали. Хочется собирать сторонние пакеты по спецификациям, минимально отличающимся от “родных”.

Пожелания и конфликты

- gcc-3.4.5
  - не могу назвать его gcc – конфликтует с 2.95;
  - не могу установить его по-умолчанию: libgcc_s.so хочет занять место в /lib, его SONAME такой же, как и у libgss_s.so из 2.95;
- qt-3.3.4
  - хотя он и устанавливается в дерево с полностью конфигурируемым именем, если внести его в ld.so.conf до “родного” qt-3.3.3, падают зависящие от него программы, если после – сторонние программы его не находят – это опять же, следствие одинакового SONAME;
  - пакет нельзя назвать по-умолчанию – если так сделать, он окажется более новой версией qt2 или qt3. я же хочу, чтобы они стояли одновременно;

Резюме

- Имена сторонних пакетов не должны говорить о том, что они – обновления “родных”.
- Имена файлов в сторонних пакетах не должны совпадать с именами файлов в “родных”. Нужна возможность выбора, на файлы какого пакета ссылаются имена программ – механизм, известный как alternatives.
- ld.so.conf не должен изменяться сторонними пакетами. Однако, информация предназначенная для ld.so.conf должна где-то собираться.

Решения

Все расширения объединил в пакет altenvironment. От него должны зависеть (BuildRequires, Requires) все сторонние пакеты.

Имена пакетов

Имена пакетов изменяются добавлением тэга, определяющего конфигурацию, в которой

собран пакет.

До сих пор это был суффикс -mcbc (gcc-mcbc, gcc-mcbc-c++). С суффиксом сложность – куда его вставлять в многосложные названия (сейчас – после первого слова). А кроме того, логичнее иметь все сторонние пакеты кучкой в общем репозитарии (хотя, несущественно).

Сейчас мне кажется, что логичнее, чтобы это был префикс, определяющий конфигурацию и “стороннесть” (nic-gcc, nic-gcc-c++).

Имена исполняемых файлов

Если исполняемые файлы находятся в разделяемых каталогах (/bin, /usr/bin) их имена должны включать в себя тэг, аналогичный имени пакета. В остальных случаях смысла в этом нет.

Если требуется вызов программ из разных пакетов по одному имени, следует использовать alternatives. Однако, возникает сложность с размещением символьной ссылки, если исходный пакет не рассчитан на alternatives. Так, например, в случае gcc (2.95) и gcc-3.4.5-mcbc, пришлось делать ссылку /bin/gcc.

Пути поиска динамических библиотек и ld.so.conf

ld.so.conf не модифицировать. В ld.so.conf.d не писать. ldconfig в post и postun не вызывать так. Однако, собирать всю означенную информацию в симметричном месте, соответсвующем конфигурации (/etc/altenvironment/3.4.5-mcbc). Вызывать ldconfig с этой информацией, но без обновления ld.so.cache. Здесь – очень странное место: я бы предпочёл обновлять свой, сторонний кэш, а потом подсовывать его ld.so при запуске сторонней программы. Первая половина этого процесса (другой кэш) возможна, вторая – нет): Почему – загадка. Далее, DT_RUNPATH vs. LD_LIBRARY_PATH. Соблазнительно использовать первое. Однако на эти грабли мы уже наступали – получается очень жесткая, неконфигурируемая структура. Одного моего опыта, конечно, мало. Поэтому ссылаюсь на мнение разработчиков Fedora: Beware of Rpath. Итак, LD_LIBRARY_PATH. Устанавливается скриптом, сейчас он называется /usr/bin/run-3.4.5-mcbc. Объединяет все пути найденные в /etc/altenvironment/3.4.5-mcbc/*.conf. Если этот скрипт запускать по имени “$0", то он после установки LD_LIBRARY_PATH запускает программу "$0-3.4.5-mcbc”. Соответственно, скрипт /usr/bin/link-3.4.5-mcbc переименовывает переданные ему в качестве параметров бинарники в *-3.4.5-mcbc и создает на их месте символьную ссылку на run-3.4.5-mcbc.