GradC++ ToolBox: Руководство администратора

DocumentId:GradSoft-PR-r-09.04.2000-v1.1.0

April 29, 2001

Contents

1 Общие сведения

Пакет Gen включает четыре подпакета:

1. ProgOptions - класс для обработки опций командной строки

2. ServiceOptions - специализация ProgOptions для CORBA сервисов.

3. Logger - класс для организации записи информационных и отладочных сообщений.

4. Threading - набор классов для организации мультипотоковых программ.

Этот пакет разработан компанией GradSoft, последняя версия всегда доступна на нашем сайте: http://www.gradsoft.com.ua

1.1 Объем инсталляции

Пакет Gen может быть установлен как нечто целое "одним движением руки" (рекомендуется). Кроме того, каждый отдельный подпакет и усеченный комплекс, состоящий из трех не использующих CORBA подпакетов (т.е. `ProgOptions + Logger + Threading`) могут быть установлены отдельно.

2 Порядок инсталляции пакета

2.1 Необходимое ПО

1. CORBA ORB (не требуется, если вы не используете ServiceOptions):
   - Unix: omniORB-3.0 или выше, TAO-5.9b или выше, ORBacus 3.3.2 или выше, ORBacus 4.0.2 или выше
   - WindowsNT: поддерживается ORBacus версии 3.x и 4.x или выше.
2. Компилятор C++ :
   - Unix: gcc-2.95.2 или выше, SunProc C++ 4.2
   - Windows NT: Microsoft Visual C версии 6.0 или выше.
3. make:
   - Unix: необходим gnu make
   - WindowsNT: nmake из поставки MSVC++

2.2 Порядок инсталляции под UNIX

- Убедиться, что необходимое ПО установленно и работатет.
- Развернуть архив Gen.tar.gz в избранном Вами каталоге. Будем в дальнейшем называть его `<project_root>`
- Перейти в этот каталог.

- Запустить configure командой `./configure` с нужными опциями (список опций выводтися при запуске `./configure` с опцией `--help`; в частности, можно задать опцию `--prefix=<smth>` для назначения каталога инсталляции либо опцию `--with-corba=<yes/no>` для того, чтобы указать объем компиляции/инсталляции)
- Запустить компиляцию, с помощью команды `gmake`
- Перейти в root режим с помощью команды `su`
- Запустить инсталляцию с помощью команды `gmake install`
- Для деинсталяции пакета можно воспользоваться командой gmake uninstall

2.3 Порядок инсталляции под Windows NT

- Убедиться, что необходимое ПО установленно и работатет. Прописать пути к утилитам `nmake`, `cl` и `xcopy` в переменной среды `PATH`.
- Развернуть архив Gen.tar.gz в избранном Вами каталоге (в дальнейшем этот каталог мы будем называть `<project_root>`)
- Отредактировать файл env_inc.nt.mak в подкаталоге `<project_root>\config`. При редактировании установить значения следующих nmake-переменных:

имя переменной	*описание*
PROJECT_ROOT	`<project_root>` (корневой каталог проекта)
INSTALL_IDL_DIR	каталог, в который будут помещены инсталлированные idl-модули
INSTALL_INC_DIR	каталог, в который будут помещены включаемые файлы
INSTALL_LIB_DIR	каталог, в который будут помещены инсталлированные библиотеки
ORB_DIR	корневой каталог ORB
MSVC_DIR	корневой каталог Microsoft Visual Studio

Примечания:

1. Строго говоря, включаемые файлы будут размещены в подкаталоге `$(INSTALL_INC_DIR)\GradSoft`
2. Для совместного использования Gen с другими продуктами GradSoft, переменные
   INSTALL_IDL_DIR,
   INSTALL_INC_DIR и
   INSTALL_LIB_DIR
   рекомендуется определить следующим образом:
   - выделить специальный каталог-для-инсталляции и сохранить его имя в переменной INSTALL_DIR.

  - определить переменные INSTALL_IDL_DIR, INSTALL_INC_DIR и INSTALL_LIB_DIR через переменную INSTALL_DIR так, как это сделано по умолчанию.

- Для компиляции пакета перейти в каталог `<project_root>` и воспользоваться командой

  `make build`

- Для инсталляции пакета перейти в каталог `<project_root>` и воспользоваться командой

  `make install`

- Для деинсталляции пакета перейти в каталог `<project_root>` и воспользоваться командой

  `make uninstall`

- Для очистки пакета (для удаления файов, созданных в процессе компиляции) перейти в каталог `<project_root>` и воспользоваться командой

  `make clean`

- Для компиляции, инсталляции, деинсталляции либо очистки одного отдельно взятого подпакета перейти в соответствующий ему каталог (один из четырех:
  `<project_root>\ProgOptions`,
  `<project_root>\CORBA\ServiceOptions`,
  `<project_root>\Logger`
  либо
  `<project_root>\Threading`)
  и воспользоваться командой "make" с подходящей опцией (см. раздел "Детали инсталляции под Windows NT 5)

- Для одновременной компиляции, инсталляции, деинсталляции либо очистки трех подпакетов (ProgOptions + Logger + Threading) перейти в каталог `<project_root>` и воспользоваться командой "make.cut" с подходящей опцией (см. раздел "Детали инсталляции под Windows NT 5)

3 Тестирование

3.1 Общие сведения

Тестовые примеры, входящие в комплект поставки Gen, соответствует примерам использования средств ProgOptions, ServiceOptions и Logger, описанным в руководствах программиста для каждого подпакета.

3.2 Порядок тестирования

3.2.1 ProgOptions

- Прочитать руководство программиста в файле project_root/ProgOption/docs/ProgrammingGuide_rus.pdf
- Перейти в каталог project_root/ProgOptions/demo
- Запустить ProgOptionsDemo поочередно со следующими опциями:

```
--help
--qqq
--zzz
--zzz <arg>
--zz1 <arg1>
--zz3
<arg2>
--<arg3>
```

- Убедиться, что вывод соответствуем ожиданиям

3.2.2 ServiceOptions

- Перейти в каталог `<project_root>/CORBA/ServiceOptions/demo`
- Запустить программу HelloWorldServer с опцией `--help`
- Прочитать help.
- Запустить программу HelloWorldServer с опцией `--ior-stdout`
- Убедиться, что Interoperable Object Reference выведена на стандартный вывод.
- Запустить программу HelloWorldServer с опциями `--ior-file-HelloService hello.ref` .
- Убедиться, что Interoperable Object Reference находится в файле hello.ref.
- Запустить программу HelloWorldServer с опциями:
  - `-OAport 1025`, если вы используете ORBacus
  - `-ORBport 1025`, если вы используете TAO
  - `-ORBpoa_iiop_port 1025`, если вы используете omniORB
- Запустить программу HelloWorldClient с опциями
  - Если вы используете omniORB или ORBacus 4.X: `-ORBInitRef HelloService=corbaloc::127.0.0.1:1025/HelloService`

– Если вы используете TAO или ORBacus 3.X:
`-ORBInitRef HelloService=iiop//:127.0.0.1:1025/HelloService`

Port 1025 должен не быть занят другим приложением; естественно, вы можете указать любой другой номер порта.

- Убедиться, что строка "Hello World" выведена на стандартный вывод
- Проверить правильность настройки NameService
- Запустить программу HelloWorldServer с опцией `--with-naming`
- Запустить программу HelloWorldClient с опцией `--with-naming`
- Убедиться, что строка "Hello World" выведена на стандартный вывод

3.2.3 Logger

- Прочитать руководство программиста в файле `<project_root>/Logger/docs/ProgrammingGuide\_rus.pdf`
- Перейти в каталог `<project_root>/Logger/demo`
- Запустить исполняемый файл
- Убедиться, что файл file.log сгенерирован и его содержание отвечает описанному в ProgrammingGuide_rus.pdf порядку использования пакета

3.2.4 Treading

Поочередно перейти в каталоги

- `<project_root>/Treading/demo/Philosophs` и
- `project_root/Treading/demo/Container` ,

убедиться, что исполняемые файлы созданы и работают.

4 Использование

Пользовательские программы, использующие средства ProgOptions, ServiceOptions и Logger должны быть скомпилированы со следующими библиотеками:

Для Unix:

используемый пакет	*статические библиотеки*	*динамические библиотеки*
ProgOptions	libProgOptions.a	libProgOptions.so
ServiceOptions	libProgOptions.a libServiceOptions.a	libProgOptions.so libServiceOptions.so
Logger	libLogger.a	libLogger.so
Threading	libThreading.a	libThreading.so

Для Windows NT:

используемый пакет	*статические библиотеки*	*динамические библиотеки* *)
ProgOptions	ProgOptions.lib	
ServiceOptions	ServiceOptions.lib ProgOptions.lib	
Logger	Logger.lib	
Threading	Threading.lib	

*) будет сделано в следующих версиях

5 Детали инсталляции под Windows NT

5.1 Команды инсталляции

Инсталляция под Windows NT производится командой make, реализованной файлом make.bat, либо командой make.cut, реализованной файлом make.cut.bat.

5.1.1 Команда make (файл make.bat)

1. Дислокация:

   Пять идентичных файлов make.bat находятся в пяти разных местах:

   (a) в корневом каталоге проекта (project_root);

   (b) в каталоге `project_root\ProgOptions`;

   (c) в каталоге `project_root\CORBA\ServiceOptions`;

   (d) в каталоге `project_root\Logger`;

   (e) в каталоге `project_root\Threading`.

   Запуск make.bat из каждого конкретного каталога предполагает определенное ограничение сферы действия команды:

   - `project_root` - действия будут касаться всего пакета Gen,
   - `project_root\ProgOptions` - действия будут касаться только пакета ProgOptions,
   - `project_root\CORBA\ServiceOptions` - действия будут касаться только пакета ServiceOptions,
   - `project_root\Logger` - действия будут касаться только пакета Logger.
   - `project_root\Threading` - действия будут касаться только пакета Threading.

2. Формат команды make:

`make <target>`

где `<target>` принимает одно из четырех значений: build, install, uninstall и clean. Действия, соответствующие каждому из этих значений, определяются по следующей таблице:

значение `<target>`	*описание действий*
build	компиляция пакета/подпакета;
install	инсталляция пакета/подпакета (копирование idl-модулей, включаемых файлов, а также собранных в процессе компиляции библиотек в определенные пользователем каталоги)
uninstall	деинсталляция пакета/подпакета (удаление скопированных файлов)
clean	очистка пакета/подпакета (удаление файлов, автоматически создающихся при компиляции)

3. Особенности использования:

   (a) команда `make` с пустым значением `<target>` имеет смысл и эквивалентна команде `make build`.

   (b) запуск команды

   `make install`

   без предварительного запуска команды

   `make build`

   разрешен (рекомендуется) и приводит к автоматической компиляции той части пакета/подпакета, которая необходима для создания передаваемых пользователю объектов.

5.1.2 Команда make.cut (файл make.cut.bat)

Команда make.cut аналогична команде make, запущенной из корневого каталога за исключением ограниченной сферы действия: запуск make.cut приводит к однородной обработке трех не использующих CORBA подпакетов (ProgOptions, Logger и Threading) вместо четырех.

- Формат команды :

`make.cut <target>`

где `<target>` определяется по таблице:

значение `<target>`	*описание действий*
build	компиляция трех подпакетов;
install	инсталляция трех подпакетов (копирование idl-модулей, включаемых файлов, а также собранных в процессе компиляции библиотек в определенные пользователем каталоги)
uninstall	деинсталляция трех подпакетов (удаление скопированных файлов)
clean	очистка трех подпакетов (удаление файлов, автоматически создающихся при компиляции)

- Особенности использования команды аналогичны описанным выше.

5.2 Зависимости

Пакет ServiceOptions использует средства ProgOptions, и в полном объеме может быть скомпилирован лишь при наличии библиотеки ProgOptions.lib. Генерация ProgOptions.lib происходит в двух случаях:

1. при компиляции ProgOptions в полном объеме
2. при инсталляции ProgOptions без компиляции в полном объеме

Следовательно, при необходимости откомпилировать ServiceOptions отдельно от других проектов, одно из этих действий надо выполнить до начала компиляции. В противном случае, компиляция демо-части ServiceOptions приведет к ошибке типа

```
LINK : fatal error LNK1181: cannot open input file "D:\<project_root>\CORBA\ServiceOpti
ons\..\..\ProgOptions\src\ProgOptions.lib"
```

и будет прервана.

6 Перечень изменения

14.02.2001 - добавлена информация о параметрах ORB в разделе "тестирование ServiceOptions"

09.04.2000 - создание.