

ModCBroker: Руководство Администратора

DocumentId:GradSoft-Cbroker-AG503-79-10.05.2000-3.3.0

Руслан Шевченко, Александр Яновец, Юлия Прокопенко

October 15, 2008

Contents

1	Введение	2
2	Компиляция и Инсталляция <i>ModCBroker</i>	2
2.1	Для UNIX	2
2.1.1	Необходимое ПО:	2
2.1.2	Компиляция ModCbroker как DSO модуля	2
2.1.3	Параметры configure	3
2.2	Для Windows.NT	4
2.2.1	Необходимое ПО	4
2.2.2	Процесс инсталляции:	4
2.3	Компиляция клиентской библиотеки <i>clcbroker</i>	5
2.3.1	Для Unix	5
2.3.2	Для Windows NT	6
3	Тестирование	6
3.0.3	Замечание о режиме выполнения потоков ORB	6
4	Использование	7
5	Настройка	7
5.1	Перечень директив	7
5.2	Типичные примеры	9
5.2.1	Обращение к сервлету для выдачи контента	9
5.2.2	Использование сервлетов для авторизации	10
5.2.3	Использование фильтров	10
6	Интеграция с tcl [3] [2]	10
7	Интеграция с mod_ssl [1]	10
7.0.4	Для Unix	10
7.0.5	Для Windows NT	11
8	Перечень изменений	11

1 Введение

ModCBroker – модуль Apache, обеспечивающий трансляцию WWW запросов в CORBA запросы. Модуль разрабатывается и поддерживается компанией GradSoft, дистрибутив ModCBroker в исходных кодах доступен по адресу <http://www.gradsoft.kiev.ua>. Версия ModCBroker, для которого написано это руководство: 3.2.0

2 Компиляция и Инсталляция *ModCBroker*

2.1 Для UNIX

2.1.1 Необходимое ПО:

1. Операционная система: Sun Solaris 2.6 или выше, Linux с ядром 2.2.0 или выше, FreeBSD версии 4.7 или выше. В принципе возможен перенос на любую posix-совместимую платформу.
2. ORB - один из ORBacus-4.1, omniORB, TAO, MICO
3. Apache - 2.2.3 или выше.
4. Компилятор C++, совместимый с используемым брокером объектных запросов
5. Make - GNU make версии 3.76 или выше.

2.1.2 Компиляция ModCBroker как DSO модуля

Необходимые предусловия:

- ORB должна быть скомпонована в виде набора разделяемых библиотек.
- Apache должен быть скомпилирован с поддержкой `mod_so` и установлен вместе с набором включаемых файлов.
- Если ORB требует специфических опций компилятора или компоновщика, несовместимых с опциями компиляции Apache по умолчанию, Вам возможно перед инсталляцией придется перекомпилировать Apache, установив эти опции в параметрах `EXTRA_CFLAGS` и `LD_FLAGS_SHLIB` конфигурации Apache.

В частности такая ситуация возникает при использовании комбинации FreeBSD 4.x и любой многопоточной ORB: вы должны перекомпилировать Apache указав при компиляции:

```
EXTRA_CFLAGS=-pthread
LD_SHLIB=gcc
LD_FLAGS_SHLIB=-shared -pthread
```

Процесс инсталляции:

1. Развернуть архив `ModCBroker.tar.gz` с помощью последовательности команд:
 - `gzip -d ModCBroker.tar.gz`
 - `tar -xvf ModCBroker.tar`
2. Перейти в образовавшуюся директорию `ModCBroker`
3. Запустить `./configure`
При необходимости, указать опции `configure`, описанные в разделе 2.1.3
4. Запустить `gmake`
5. Запустить `gmake install` После этого DSO модуль с именем `mod_cbroker.so` будет скопирован в каталог `<prefix>/libexec/apache2 //`
6. Добавить в конфигурацию Apache (`httpd.conf`) строчку

```
LoadModule cbroker_module <prefix>/libexec/apache2/mod_cbroker.so.
```
7. Если необходимо, добавить туда-же строку `AddModule mod_cbroker`
8. Перезапустить Apache

После этого модуль загружается при старте Apache и вы можете переходить к тестированию.

2.1.3 Параметры `configure`

- путь к установленным заголовочным файлам Apache
`--with-apache-headers=<path-to-apache-headers>`. По умолчанию, `configure` ищет эти файлы в каталогах
 - `/usr/local/include/apache2,`
 - `/usr/include/apache,`
 - `/usr/local/include/apache,`
 - `/usr/local/apache/include,`
 - `/usr/local/etc/apache/include.`
- тип используемой ORB и путь к префиксу инсталляции используемой ORB:
 - `--with-tao=<tao-root>` - для TAO
 - `--with-ob=<ob-prefix>` и `--with-jtc=<jtc-prefix>` - для ORbus (IONA)

- `--with-obe=<obe-prefix>`, `--with-jtc=<jtc-prefix>` и `--ansiicpp`
- для ORbix/E (IONA)
- `--with-omni=<omni-root>` - для omniORB
- `--with-visi=<visi-prefix>` | - для VisiBroker (Borland)

По умолчанию `./configure` пытается определить конфигурацию типичной установки ORB, так что вам эти опции возможно не понадобятся.

Список всех опций конфигурации можно посмотреть с помощью команды

```
./configure --help
```

Вы можете обратиться к [4], для подробного описания механизма выбора ORB. Если что-то идет не так, можете написать нам.

2.2 Для Windows.NT

2.2.1 Необходимое ПО

1. ORB - ORBacus-4.0.x
2. Проинсталлированный Apache, версии 2.2.3 или выше.
3. Компилятор C++ Microsoft Visual C версии 6.0 или выше.

2.2.2 Процесс инсталляции:

1. Развернуть архив `ModCBroker.zip`.
2. Перейти в образовавшуюся директорию `ModCBroker/msvc.gen`
3. Определить значения следующих `make`-переменных в файле `env_inc.mak`:
 - `PROJECT_ROOT` - абсолютный путь к `ModCBroker`
 - `TARGET_ORB` - тип используемой ORB, т. е.
 - `ORBACUS` - для `Orbacus`
 - `TAO` - для `TAO`
 - `OMNI` - для `OmniIDL`
 - `ORB_DIR` - где инсталлирована данная ORB.
 - `APACHE_DIR` - директория, где находится проинсталлированный Apache.
 - `APACHE_MODULE_DIR` - директория, где находится библиотека. Обычно - директория `modules` в корневой директории проинсталлированного Apache. Это значение установлено по умолчанию.
 - `INSTALL_DIR` - абсолютный путь к целевой директории инсталляции
 - `INSTALL_IDL_DIR` - директория, куда будут инсталлированы IDL файлы. Установлена по умолчанию.

- `INSTALL_INC_DIR` - директория, куда будут инсталлированы включаемые файлы.
- `INSTALL_IDL_DIR` - директория, куда будут инсталлированы IDL файлы. Установлена по умолчанию.
- `INSTALL_LIB_DIR` - директория, куда будут инсталлированы библиотеки.
- `INSTALL_IDL_DIR` - директория, куда будут инсталлированы IDL файлы. Установлена по умолчанию.
- `MSVC_DIR` - директория, где находится MS Visual C++

4. Зайти в директорию `ModCBroker/msvc.gen`
5. Запустить компиляцию, запустив `make.bat`
6. Инсталлировать, набрав `make install`
7. Инсталлировать библиотеку, набрав `make install-dll`
8. Вписать в конфигурацию Apache (`httpd.conf`) строчку

```
LoadModule cbroker_module modules/mod_cbroker.so
```

Данная директива должна идти последней в списке директив `LoadModule`

2.3 Компиляция клиентской библиотеки *clcbroker*

Так как клиентская библиотека состоит просто из генерируемых стаб файлов, то ее можно собрать на любом компьютере с инсталлированной ORB просто взяв файл `idl/HTTP.idl`, сгенерировать объектные стабы и скелетоны с помощью `idl` компилятора, и либо добавить их в свой проект, либо собрать в библиотеку `libclcbroker.so` (`clcbroker` для UNIX)

В процессе компиляции *ModCBroker* клиентская библиотека также генерируется, поэтому если у вас клиент и сервер находятся на одном и том-же компьютере, то никаких дополнительных действий производить не нужно.

2.3.1 Для Unix

1. Развернуть инсталляцию *ModCBroker*
2. Отконфигурировать, сказав `./configure --without-apache`
3. Откомпилировать, командой `make client`
4. Инсталлировать клиентскую библиотеку, командой `make install-client`

2.3.2 Для Windows NT

1. Развернуть инсталляцию *ModCBroker*
2. Откомпилировать, командой `make client`
3. Инсталлировать командой `make install-client`

3 Тестирование

1. После инсталляции *ModCBroker* убедитесь, что Apache работает и, впишите в конфигурацию (`httpd.conf`) следующий блок:

```
CbrokerORBArgs -ORBInitRef NameService=corbaloc::you-host-name:10000/NameService
CbrokerLocation /cbroker
```

2. Перезапустите `httpd`
3. Запустите CORBA NameService на 10000 порт. (для ORBACUS команда: `nameserv -OApport 10000`)
4. Зайдите в каталог `mod-cbroker-dir/demo/HelloWorld`
5. Запустите `HelloWorldServlet`, указав ему имя NameService командой:

```
./HelloWorldServlet -ORBInitRef \
    NameService=corbaloc::you-host-name:10000/NameService
Для TAO 1.1
./HelloWorldServlet -ORBInitRef \
    NameService=iioploc://you-host-name:10000/NameService
```

6. Наберите в WWW браузере следующий URL: `http://you.host.name/cbroker/Hello/Hello`

3.0.3 Замечание о режиме выполнения потоков ORB

В некоторых ORB (к примеру - ORBacus), режимы выполнения потоков по умолчанию (`threading mode`) не поддерживают обратные вызовы (т. е. `callback` функции). То есть вам необходимо самостоятельно установить режим выполнения потоков в конфигурационных параметрах ORB, в противном случае ваш клиент остановится во время первого вызова метода `HTTPStream`

К примеру, для ORBacus директива `CbrokerORBArgs` в конфигурационном файле `httpd.conf` в нашем примере должна выглядеть следующим образом:

```
CbrokerORBArgs -ORBInitRef NameService=corbaloc::you-host-name:10000/NameService \
    -ORBreactive -OAreactive
```

В то время, как для всех остальных ORB режимы потоков остаются по умолчанию:

```
CbrokerORBArgs -ORBInitRef NameService=corbaloc::you-host-name:10000/NameService \
```

4 Использование

1. Убедиться что в конфигурации Apache2 загружается `mod_cbroker.so`
2. Установите параметры конфигурации в `httpd.conf` (см. ??)
3. Пользовательские программы, работающие back-end-ом для `ModCBroker` должны быть скомпилированы с библиотекой `libclcbroker.so` либо `lib-clcbroker.a` (для UNIX), `clcbroker.lib` (для Windows NT)

5 Настройка

Настройка `ModCBroker` заключается в установке обработчика (вы можете установить обработку по директории), в настройке авторизации `apache` и, при необходимости, настройке авторизационных сервлетов, фильтров и интерфейсов со скрипт-языками.

5.1 Перечень директив

1. `CbrokerORBArgs <arguments for ORB>`
 - Устанавливает параметры инициализации ORB.
 - Значение по умолчанию: `-ORBconfig /etc/orb.cf`
2. `CbrokerLocation`
 - Устанавливает виртуальную директорию для `ModCbroker`
 - Значение по умолчанию: `/cbroker`
3. `CbrokerDefaultServlet`
 - Устанавливает имя сервлета по умолчанию: мы вызываем этот сервлет в том случае, когда в запросе элемент пути, соответствующий сервлету пуст.
 - Значение по умолчанию: `index`
4. `CbrokerDefaultHandler`
 - Устанавливает имя обработчика по умолчанию: мы вызываем этот обработчик в том случае, когда соответствующий элемент пути отсутствует.
 - Значение по умолчанию: `index`
5. `CbrokerAddAuthByLocation location servletName handlerName`
 - Устанавливает авторизационный обработчик для директории. Параметры:
 - `location` - директория, которую мы защищаем.
 - `servletName` - имя авторизационного сервлета.

- handlerName - имя авторизационного обработчика.
 - Значение по умолчанию: нет.
6. CbrokerPrefix
 - Значение тега начала вызова сервлета в фильтре CBROKER.
 - Значение по умолчанию: <?cbroker
 7. CbrokerSuffix
 - Значение тега конца вызова сервлета в фильтре CBROKER.
 - Значение по умолчанию: ?>
 8. CbrokerPrefix0
 - Значение тега начала вызова сервлета в фильтре CBROKER0.
 - Значение по умолчанию: <?cbroker0
 9. CbrokerSuffix0
 - Значение тега конца вызова сервлета в фильтре CBROKER0
 - Значение по умолчанию: ?>
 10. CbrokerFormAuth on|off
 - Используется внутри директории, когда мы применяем form-авторизацию вместо стандартной авторизации http (Basic или Digest). В этом случае когда ресурс требует авторизации – вместо кода ошибки HTTP_UNAUTHORIZED (по которому браузер спрашивает у пользователя логин и пароль) mod_cbroker возвращает перенаправление на форму ввода логина и пароля (заданную пользователем или встроенную). После ввода логина и пароля, авторизационные значения сессии в зашифрованном виде ассоциируются с сессией клиента при помощи cookie.
 - Значение по умолчанию: off
 11. CbrokerFormAuthCookieName
 - Имя cookie, в которой храняться авторизационные значения сессии.
 - Значение по умолчанию: sac
 12. CbrokerFormAuthKey Ключ, в соответствии с которым кодируется выражения login:password (в соответствии с алгоритмом blowfish)
 13. CbrokerFormAuthLoginUrl
 - URL (относительно корня хоста) по которому показывается форма входа.

- Значение по умолчанию: /cbroker-login

14. CbrokerFormAuthLogoutUrl

- URL (относительно корня хоста), обращение к которому означает выход клиента из программы.
- Значение по умолчанию: /cbroker-logout

15. CbrokerFormAuthAfterLogoutUrl

- URL (относительно корня хоста), на который перенаправляется клиент после выхода.
- Значение по умолчанию: /

16. CbrokerFormAuthLoginFormTemplate

- имя файла, в котором находится форма ввода логина/пароля при авторизации с помощью формы.
- Значение по умолчанию: нет (показывается встроенная форма).
- Форма должна иметь три поля ввода: username, password и backurl (url перехода). Также в тексте html формы допустимо использование переменных \$(LOGIN_URL) и \$(BACK_URL) которые замеются на значения cbrokerFormAuthLoginUrl и backurl соответственно.

5.2 Типичные примеры

5.2.1 Обращение к сервлету для выдачи контента

```
CbrokerORBArgs -ORBInitRef NameService=corbaloc::www.name.com:10000/NameService
CbrokerLocation /cbroker
```

И если вам нужна авторизация, то для выделенного ресурса должна быть настроена авторизация этого ресурса, например так:

```
<Location /cbroker>
AuthType Basic
AuthName authorization
require valid-user
</Location>
```

Этот пример говорит, что

1. apache вызывает обработчик ModCBroker когда мы обращаемся к виртуальной директории cbroker. (т. е. когда URL запроса начинается с /cbroker)
2. Встроенной ORB передается строка параметров инициализации:

```
-ORBInitRef NameService=corbaloc::www.name.com:10000/NameService
```

3. для доступа требуется авторизация. Заметим, что авторизация производится в пользовательском сервлете, хотя в конфигурации об этом ничего не сказано.

5.2.2 Использование сервлетов для авторизации

Допустим, Вы хотите определить с помощью CORBA сервлетов доступ к определенному набору ресурсов (Web страницам либо приложению).

В таком случае надо воспользоваться директивой `CbokerAddAuthByLocation`, не забыв при этом указать Apache на необходимость авторизации этого ресурса.

```
CbokerAuthLocation MyProtectedArea AuthServlet AuthHandler
<Location MyProtectedArea>
AuthType Basic
AuthName MyResources
require valid-user
</Location>
```

При отсутствии директив `AuthType` Apache будет генерировать внутреннюю ошибку.

5.2.3 Использование фильтров

`ModCboker` также определяет два фильтра: `CBROKER` и `CBROKER0`. Вы можете указать использование этих фильтров при обработке файлов определенных типов с помощью команды `AddOutputFilterByType`. Например:

```
AddOutputFilterByType CBROKER cbhtml
```

6 Интеграция с tcl [3] [2]

Также можно использовать `cbroker` совместно со скрипт-языком. Как пример, вы можете найти в дистрибутиве, в поддиректории `add_on/tcl` расширение `tcl`, которое должно загружаться из `ssh`

Его надо откомпилировать как расширение `Tcl`, соответствующее стандарту TEA, и загрузить результирующую разделяемую библиотеку в секции инициализации интерпретатора. После этого можно будет пользоваться командой `cbroker` в скрипт-языках.

7 Интеграция с mod_ssl [1]

7.0.4 Для Unix

Начиная с Apache-2.x ничего особенного делать не надо.

1. Установите `mod_ssl`

2. Откомпилируйте и установите `mod_cbroker`, как обычный DSO модуль. при установке он должен быть скомпилирован с `mod-ssl`

7.0.5 Для Windows NT

Во первых, необходимо иметь проинсталлированный apache (вер. 2.0.44 и выше).

Перед компиляцией `mod_cbroker` в файле `env_inc.mak` к переменной `CFLAGS` добавьте параметр `/DEAPI`.

Далее следуйте общим инструкциям по инсталляции.

8 Перечень изменений

- 15.10.2008 - добавлено описание опций для авторизаци с помощью формы.
- 13.06.2007 - пересмотрено для ModCbroker-3.2.0, убрана устаревшая информация.
- 23.05.2003 - пересмотрено для ModCbroker-3.1.1, убрана устаревшая информация.
- 04.02.2003 - пересмотрено для ModCbroker-3.1.0
- 07.06.2002 - просмотр.
- 21.05.2002 - пересмотрено для ModCbroker-3.0, убрано описание статической компиляции.
- 18.03.2002 - добавлена матрица поддежки и контрибуция Ричарда Боуска.
- 15.02.2002 - добавлено замечание о ТАО-1.1
- 09.01.2002 - просмотр.
- 16.08.2001 - добавлено замечание о инсталляции Apache на FreeBSD используя коллекцию портированного ПО (ports collection)
- 21.06.2001 - изменен порядок инсталляции для Windows NT.
- 02.06.2001 - изменения структуры.
- 25.05.2001 - добавлено описание компиляции DSO модуля.
- 22.03.2001 - добавлены инструкции по интеграции с `mod_ssl`.
- 17.03.2001 - просмотр.
- 04.02.2001 - добавлено замечание о режиме выполнения потоков.
- 30.01.2001 - добавлены формальные атрибуты документа

- 28.11.2000 - добавлено указание на необходимость предварительного конфигурирования Apache.
- 10.05.2000 - первая редакция

References

- [1] Ralf. S. Engelschall Ben Laurie. *mod_ssl home page*, 1995-2001. <http://www.modssl.org>.
- [2] Apache Software Foundation NetCetera AG. *websh home page*, 2000-2002. <http://tcl.apache.org/websh>.
- [3] J. K. Osterhout. Tcl: An embeddable command language. pages 133–146, 1990.
- [4] Ruslan Shevchenko. *corbaconf: autoconf-based package for CORBA based applications*, 2000. <http://corbaconf.kiev.ua>.