RenderWare Graphics

Белая книга

Здание RenderWare Графика

Связаться с нами

Критерион Софтвер Лтд.

Для получения общей информации о RenderWare Graphics отправьте электронное письмоinfo@csl.com .

Участники

Команды разработки и документирования RenderWare Graphics.

Оглавление

1. Введение

В этом документе описывается процесс, используемый для сборки RenderWare Graphics SDK из исходного дерева.

Кроме того, читателю предоставляются некоторые подсказки и советы по применению некоторых более сложных методов строительства, используемых командами разработчиков CSL.

Процесс описывается изнутри наружу – начиная с описания инструментов и процессов, используемых для создания одного набора библиотек RenderWare Graphics – затем расширяя его до процесса более высокого уровня, используемого для создания всего SDK – включая несколько наборов библиотек, предварительно скомпилированные примеры и документацию. Также предоставляется краткий обзор использования инструмента GUI Windows для управления процессом сборки RenderWare Graphics.

2. Обзор

Поскольку RenderWare Graphics является многоплатформенным решением, мы внедрили процесс и цепочку инструментов для создания SDK, которые позволяют использовать одну и ту же систему на всех платформах.

Где это целесообразно, мы не делаем никаких предположений о платформе, используемой для создания SDK. В CSL мы используем систему, размещенную на ПК, для создания и поддержки SDK – однако внутренние и внешние разработчики также используют среды разработки, размещенные на Linux (по крайней мере для PlayStation 2). Смотреть *5 Сборка Playstation 2 SDK на Linux* для получения дополнительной информации об использовании Linux в качестве среды сборки для Playstation 2.

Все проекты программирования требуют метода компиляции и связывания различных отдельных файлов, составляющих библиотеку или исполняемый файл. Современные компиляторы обычно поставляются с IDE (интегрированной средой разработки), которая служит для этой цели. (Примерами IDE являются Microsoft Visual Studio или Metrowerks CodeWarrior.) Однако большинство IDE тесно связаны с базовым компилятором, и в этой среде сложно (а в некоторых случаях и невозможно) использовать продукт конкурента.

Мы решили использовать свободно распространяемую утилиту командной строки gnumake в качестве основы процесса сборки RenderWare Graphics; подробности см.

http://www.fsf.org/software/make/make.html

Кроме того, мы также используем ряд функций командной строки Unix (cp, mv, rm и т. д.) в этом процессе. Все инструменты, необходимые для создания RenderWare Graphics, являются либо частью операционной системы хоста, либо поставляются с SDK (brwsdk/bin папка).

- На ПК важно, чтобыrwsdk/binДля успешной сборки папка должна находиться в PATH.

Теоретически использование этих инструментов сборки позволяет нам собирать RenderWare Graphics с использованием любого компилятора и на любой платформе.

В соответствии с модульной природой RenderWare Graphics, процесс сборки также является модульным и построен иерархически. Каждый компонент имеет свой собственный makefile, который, в свою очередь, вызывается makefile более высокого уровня по мере включения все большего количества компонентов в процесс сборки. Иерархия makefile будет описана более подробно далее.

Не желая указывать на очевидное, если у вас есть предыдущая версия RenderWare Graphics и вы изменили содержимое каких-либо папок RenderWare Graphics и хотите сохранить измененные файлы, возможно, будет хорошей идеей переместить или переименовать их, прежде чем приступать к выполнению любых команд, описанных в этом документе, поскольку процесс сборки может перезаписать эти файлы.

3. Файл options.mak

Существует несколько способов, с помощью которых можно построить весь RenderWare Graphics SDK или его отдельные компоненты. Наиболее распространенная форма — когда все настраиваемые пользователем параметры сборки для RenderWare Graphics содержатся в одном файле управления. Этот файл управления может быть

- либо указать в командной строке процесса сборки:

```
gnumake RWOPTIONS=c:/rw/graphics/rwsdk/options.mak
```

- или, в качестве альтернативы, быть явно указано в среде:

```
установить RWOPTIONS=c:/rw/graphics/rwsdk/options.mak

gnumake
```

В любом случае необходимо указать полный путь к управляющему файлу, используя косую черту для разделения путей.

Выбор метода использования полностью зависит от ваших собственных методов разработки. Вы можете выбрать:

- либо поддерживать разные файлы управления для разных сборок и указывать их в командной строке.

```
gnumake RWOPTIONS=c:/rw/graphics/rwsdk/options.d3d8

gnumake RWOPTIONS=c:/rw/graphics/rwsdk/options.sky

и т. д.
```

- или просто используйте один контрольный файл и редактируйте его по мере необходимости.

Имя файла не имеет значения. В CSL мы обычно используемпараметры.makдля общих версий и options.d3d8 и т. д. для конкретных.

Также доступен инструмент Windows GUI для автоматизации генерации и обслуживания файлов options.mak. Это описано далее в разделе Инструмент GUI для создания графики RenderWare.

Образецпараметры.makфайл показан ниже. Этот пример для релиза библиотек D3D8

```
РВТАРГЕТ=d3d8

РВОС=победа

RWGSDK=c:/rw/graphics/rwsdk
```

```
ПЛАГИНЫ=adc anisot collis crowd dmorph dpvs hanim lodatm logo ltmap matfx
mipkl morph patch prtstd prtadv ptank pvs random skin2 spline team toon
userdata

TOOLKITS=2d 2danim anim bezpatch bmp brycntrc charse cmpkey gencpipe
geomcond import intsec ltmap mipmapk pick pitexd png quat ras ray sknsplit
slerp splpvs tiff tilerd toc vcat wing world


CDEBUG=0

RWCOMPILER=visualc

RWLOGO=1

РВСПЛАШ=0

RWMOUSE=0

DXSDK=c:/mssdk

OGLLIBPATH=c:/ogllib

RWDEBUG=0

RWDLL=0
```

Теперь описывается каждый из этих вариантов:

RWTARGET=<цель>

Это определяет целевой драйвер, для которого RenderWare Graphics будет построен. Эта опция выбирает файлы драйвера, специфичные для платформы, и параметры компилятора.

Допустимые значения для <*цель*>являются:

- sky2 – для собственных библиотек PlayStation 2
- xbox – для собственных библиотек Xbox
- opengl – для библиотек OpenGL на многих платформах (используйте RWOS для выбора вариантов)
- d3d8 – для библиотек Direct3D8 на ПК
- gcn – для собственных библиотек GameCube
- null – для драйвера NULL на ПК
- nullsky – для драйвера NULL sky на ПК

- nullxbox – для драйвера NULL Xbox на ПК
- nullgcn – для драйвера NULL GameCube на ПК

Полный список всех возможных вариантов можно получить из содержания rwsdk/makeincl/rwtargetпапка в исходном дереве.

PBOC=<ос>

Это определяет целевую операционную систему. Используется для указания файлов ОС в SDK, а также является параметром в процессе выбора компилятора.

Допустимые значения для <*ос*>являются:

- win – для сборок ПК (d3d8, opengl, null и т.д.)
- xbox – для сборок Xbox
- небо – для сборок PlayStation 2
- gcn – для сборок GameCube

Полный список всех возможных вариантов можно получить из содержания rwsdk/makeincl/rwosпапка в исходном дереве.

RWCOMPILER=<компилятор>

Это определяет компилятор, используемый для сборки SDK.

- Примечание: для любой платформы может поддерживаться несколько компиляторов.

Допустимые значения для <*компилятор*>являются:

для RWOS=win

- cwpc – CodeWarrior для Windows
- intel – компилятор Intel 4
- visualc – компилятор Microsoft Visual C 6
- net – Компилятор Microsoft Visual Studio .NET

для RWOS=небо

- cwsky – CodeWarrior для PlayStation 2

- skygcc – компилятор ee-gcc на ПК или Linux (V1.6b и V2.96)
- sky295 – gcc v2.95 на ПК
- sky2953 – gcc v2.953 на ПК
- skyprodg – серийный номер ps2cc и ps2ld

для RWOS=xbox

- xbox – VisualC 7 для Xbox
- net – Компилятор Microsoft Visual Studio .NET

для RWOS=gcn

- cwdfin – CodeWarrior для GameCube

Полный список всех возможных вариантов можно получить из содержания rwsdk/makeincl/rwos/<os>/rwcmplrпапка в исходном дереве.

RWGSDK=<путь>

Указывает местоположение графики RenderWarerwsdkкаталог.

-

Эта опция не нужна для сборки SDK, но необходима для сборки примеров и другого образца кода.

Установщик RenderWare Graphics создаст переменные средыРВГДИР, содержащий место установки RenderWare Graphics (например,c:/rw/графика), иРВГСДК, содержащий <RWGDIR>/rwsdk.

ЕслиRWGSDKпеременная окружения не существует по какой-либо причине и явно не указана в файле options.mak, то ее путь строится (локально для сборки gnumake) из <RWGDIR>/rwsdk.

RWDEBUG=<флаг rwdebug>

Указывает, следует ли создавать отладочный набор библиотек RenderWare Graphics. Отладочные библиотеки RenderWare Graphics обеспечивают дополнительную проверку параметров и другие отладочные проверки, такие как утверждения, помогающие устранить ошибки кодирования.

Допустимые варианты для <RWDEBUG*флаг*>являются:

- 1 – собрать отладочный набор библиотек

- 0 – сборка набора библиотек для выпуска

- Этот вариант делает*нет*включить отладочную информацию C в библиотеках. Чтобы сгенерировать отладочную информацию C для пошагового выполнения кода с помощью отладчика, вам нужно будет указать опцию CDEBUG (см. ниже).

RWMETRICS=<флаг rwmetrics>

Указывает, следует ли создавать набор метрик библиотек RenderWare Graphics. Библиотеки метрик RenderWare Graphics предоставляют дополнительные метрики отчетов о производительности для приложения. Метрики могут использоваться для помощи в обнаружении и, следовательно, устранении узких мест производительности в приложении.

Допустимые варианты для<RWMETRICS*флаг*>являются:

- 1 – создать набор библиотек метрик
- 0 – сборка набора библиотек для выпуска

- RWMETRICSдействительно полезно только с релизными сборками библиотеки. Должна быть возможность построитьRWDEBUG+RWMETRICSбиблиотека, но практическое применение этого будет ограничено.

ПЛАГИНЫ=<список плагинов>

Указывает плагины, которые будут построены в SDK. Это список имен каталогов плагинов (изrwsdk/плагин)разделенные пробелами.

- Если этот список не указан, то все плагины brwsdk/плагинкаталог будет создан.

НАБОРЫ ИНСТРУМЕНТОВ=<список инструментов>

Укажите наборы инструментов, которые включены в SDK. Это список имен каталогов наборов инструментов (отrwsdk/инструмент)разделенные пробелами.

- Если этот список пуст, то по умолчанию все инструменты brwsdk/инструментканалог будет создан.

CDEBUG=<c флаг отладки>

Указывает генерацию отладочной информации C в сборке. Допустимые параметры для*<c флаг отладки>*являются:

- 1 – включить отладку C

- 0 – отключить отладку C

КОПТИМИЗИРОВАТЬ=<с флаг оптимизации>

Задает оптимизацию компилятора. Допустимые параметры для *<с флаг оптимизации>* являются:

- 1 – включить оптимизацию компилятора
- 0 – отключить оптимизацию компилятора

Если не указано иное, COPTIMIZE по умолчанию будет иметь значение, противоположное CDEBUG.

RWLOGO=<логотип флаг>

Указывает, должен ли логотип RenderWare Platform отображаться на экране при запуске примеров и просмотрщиков SDK. Используется совместно с плагином логотипа для предоставления быстрого и простого способа включения логотипа в любой пример или демонстрационное приложение.

Допустимые варианты для *<логотип флаг>* являются:

- 1 – Включайте логотип платформы RenderWare в приложения
- 0 – Не включать логотип RenderWare Platform в приложения

РВСПЛАШ=<флаг-всплеск>

Используется при построении примеров для управления отображением заставки RenderWare Graphics на экране при запуске.

Допустимые варианты для *<флаг-заставка>* являются:

- 1 – Включить заставку RenderWare Graphics в примеры
- 0 – Не включайте заставку RenderWare Graphics в примеры

В настоящее время эта опция отключена по умолчанию, поскольку изображение на заставке устарело в связи с использованием последнего логотипа.

RWMOUSE=<флаг мыши>

Для примера кода, который был написан для того, чтобы требовать указатель мыши, мы предоставляем простую эмуляцию для платформ, не имеющих собственной поддержки мыши. Эта опция указывает, нужна ли эта эмуляция мыши для платформы.

Допустимые варианты для *<флаг мыши>* являются:

- 1 – платформа не имеет мыши – эмулируйте при необходимости
- 0 – платформа изначально поддерживает мышь – эмуляция не требуется

RWDLL=<флаг генерации DLL>

Процесс сборки RenderWare Graphics по умолчанию генерирует набор библиотек, с которыми затем могут быть связаны приложения. В некоторых случаях, например, экспортеры RenderWare Graphics, DLL, содержащая функциональность RenderWare Graphics, более полезна. Поэтому для *НОЛЬ, D3D8, D3D9* и *OpenGL* только цели, дополнительная опция make, RWDLL, доступна для генерации обычных библиотек RenderWare Graphics, а также DLL, содержащей большинство этих библиотек. Эта DLL называется rwg<RWTARGET>.dll и помещается в rwsdk/dll/<RWTARGET>/debug|release папка.

Обратите внимание, что перед запуском сборки DLL рекомендуется выполнить полную очистку библиотек цели, поскольку исходный код RenderWare Graphics собирается с использованием многопоточных настроек компилятора DLL.

Чтобы использовать RenderWare Graphics DLL во внешнем приложении, определите _RWDLL (обратите внимание на начальное подчеркивание) необходимо поместить в настройки сборки приложения.

Допустимые варианты для *<флаг генерации dll>* являются:

- 1 – .дллгенерируется после создания обычных библиотек.
- 0 – создаются только обычные библиотеки.

DXSDK=<путь>

Для сборок Direct3D 8 PC это указывает каталог, где можно найти Microsoft DirectX SDK. Обычно это:

```
DXSDK=c:/mssdk
```

OGLLIBPATH=<путь>

Для сборок OpenGL для ПК этот параметр указывает каталог, в котором может быть найден файл opengl32.lib.

PS2_DRIVE=<письмо водителя>

Для сборок PlayStation 2, если это определено, это будет добавлено с двоеточием к ссылкам на библиотеки Sony и файлы toolchain в \usr. (Сборки, размещенные на Linux, не должны определять это). Это может быть:

```
PS2_DRIVE=c
```

IOPPATH=<путь>

Для сборок PlayStation 2 это будет добавлено в примерах, созданных с использованием скелета и файловой системы хоста, к ссылкам на modules/*.{img,irx}. Обычно это будет:

```
IOPPATH=/usr/local/sce/iop
```

XBOXSDK=<путь>

Для сборок Xbox это указывает путь к Microsoft Xbox SDK.

Путь для этой опции должен быть указан в краткой форме, т.е.

```
XBOXSDK=c:/progra~1/micros~4/
```

4. Иерархия Makefile

Файлы makefile, используемые в процессе сборки, описаны в следующей древовидной схеме. Пожалуйста, обратитесь к каждому отдельному файлу makefile для получения более подробной информации о том, как они работают. Примечание: предполагается знакомство с gnumake для понимания этих файлов makefile.

| makefile	Файл сборки верхнего уровня. Этот файл сборки управляет компонентами, включенными в SDK. Плагины, инструменты, демонстрации, примеры и просмотрщики, составляющие SDK, определяются здесь. Обычно этот файл сборки используется вместе с одним из главных файлов управления ниже для создания полного дерева SDK. Он работает, создавая файл options.xxx для сборки, а затем вызывая файлы сборки нижнего уровня для каждого из компонентов по очереди с этим набором параметров.
| makefile.gcn	Главный make-файл для GameCube SDK
|	
| makefile.sky	Главный make-файл для PlayStation 2 SDK
|	
| makefile.win	Главный makefile для ПК SDK
|	
| makefile.xbox	Главный make-файл для Xbox SDK
+ - - - примеры	Каждый пример SDK имеет свой собственный makefile, который определяет, как должен быть построен пример. Каждый из примеров может быть построен изолированно.
| |	
| \---один из примеров	
| | makefile	

+ - - - рвсдк | | make-файл	Это главный makefile, используемый для сборки SDK. Для разработчиков, которые регулярно перестраивают одну версию RenderWare Graphics, это будет makefile, который обычно используется. Он вызывает makefile подкаталогов для сборки ядра, плагинов и инструментов.
| + - - - makeincl	В этом каталоге содержатся все файлы, включенные в make-файлы rwsdk.
| | | makeaplg	Этот файл включен во все файлы сборки архивных плагинов. Он содержит набор опций, которые являются общими для всех архивных плагинов.
| | | мейккор	Этот файл включен в основной makefile.
| | | сделалдll	Этот файл содержит опции для построения объединенной DLL, содержащей большинство библиотек RenderWare Graphics. Это возможно только для целей NULL в настоящее время.
| | | сделатьвыбор	Общие параметры makefile, общие для всех компонентов SDK.
| | | makeplug	Этот файл включен во все файлы плагинов makefiles. Он содержит набор опций, которые являются общими для всех плагины.
| | | maketool	Этот файл включен во все makefiles набора инструментов. Он содержит набор опций, которые являются общими для всех наборов инструментов.
| | | makewrld	Этот файл включен в makefile плагина world.
| | + - - - рвос	В этом каталоге содержатся параметры сборки, специфичные для конкретной ОС.
| | | + - - - гкн	
| | | | + - - - rwcmplr	Этот каталог содержит параметры сборки, специфичные для компилятора.
| | | | | + - - - cwdfin	
| | | + - - - небо	
| | | | + - - - rwcmplr	Этот каталог содержит параметры сборки, специфичные для компилятора.
| | | | | + - - - cwsky	
| | | | | + - - - skygcc	
| | | + - - - победа	

| | | | | + - - - rwcmplr — Этот каталог содержит параметры сборки, специфичные для компилятора.

| | | | | | + - - - интел

| | | | | | + - - - визуал

| | | + - - - иксбокс — Этот каталог содержит параметры сборки, специфичные для компилятора.

| | | | + - - - rwcmplr

| | | | | + - - - иксбокс

| | + - - - rwtarget — Этот каталог содержит специфичные для цели параметры сборки. Файлы здесь обычно определяют дополнительные файлы драйверов, необходимые в основной библиотеке по общему набору.

| | | + - - - д3д8

| | | + - - - по умолчанию — Параметры общие для всех целей.

| | | + - - - гкн

| | | + - - - ноль

| | | + - - - nullgcn

| | | + - - - нулевой

| | | + - - - nullxbox

| | | + - - - opengl

| | | + - - - труба — Вариант rwtarget для файлов, специфичных для конвейера.

| | | | + - - - стр2

| | | + - - - небо2

| | | + - - - иксбокс

| + - - - плагин — Каталог плагинов состоит из ряда подкаталогов – по одному на плагин. Каждый плагин имеет свой собственный makefile, который может быть использован для сборки плагина в изоляции.

| | \---один изплагинов

| | | makefile — Makefile для плагина.

| + - - - источник — Файлы ядра RenderWare Graphics.

| | makefile — Makefile для файлов ядра RenderWare Graphics.

|

| + - - - инструмент — Каталог инструментария состоит из ряда подкаталогов – по одному на инструментарий. Каждый инструментарий имеет свой собственный makefile, который может быть использован для сборки инструментария в изоляции.

| | |

| | \---один из наборов инструментов

| | | makefile — Makefile для набора инструментов.

| \---мир — Плагин world представляет собой особый случай и имеет отдельный make-файл.

| | makefile — Небольшой makefile-оболочка, инкапсулирующий makewrld.

```
|
+ - - - поделился                    Общее дерево содержит файлы, общие для
                                     демонстраций, примеров, инструментов и средств
                                     просмотра.

|      |      сделатьвыбор           Общие параметры makefile для дерева
                                     приложений. Этот файл включен во все makefile
                                     приложений.

|      |      maketarg               определяет параметры сборки для
                                                                цель
                                     d3d8, opengl, sky2 и т.д.

|      + - - - cwcommon              Общие файлы CodeWarrior. Общие файлы для
|      + - - - демоком               демонстраций. Общие файлы, включенные из
|      + - - - makeincl              makefiles приложений.

|      |      + - - - рвос           Общие файлы для каждой ОС.
|      |      |      + - - - гкн
|      |      |      | + - - - rwcmplr | | + - - -   Общие файлы для каждого компилятора.
|      |      |      cwdfin
|      |      |      + - - - небо
|      |      |      | + - - - rwcmplr | | + - - -   Общие файлы для каждого компилятора.
|      |      |      cwsky | | + - - - skygcc |
|      |      |      | + - - -
|      |      |
skyprodg
| |          |      + - - - победа
| |          |      | + - - - rwcmplr | | + - - -   Общие файлы для каждого компилятора.
| |          |      intel | | + - - - visualc \---
| |          |      xbox
| |          |
|      |      |            + - - - rwcmplr          Общие файлы для каждого компилятора.
|      |      |                  \---xbox
|      + - - - скел                  Исходные файлы скелетной структуры для демонстраций,
                                     примеров, инструментов и средств просмотра.

|      |      + - - - небо
|      |      |      + - - - мпег
|      |      + - - - победа
|      |      \---xbox
|      |
|      \---sncommon                  Общие файлы SN-Systems.

+ - - - инструменты                  Каталог инструментов SDK содержит инструменты,
                                     размещенные на ПК. Они созданы с помощью одной
                                     из библиотек PC SDK.

|      |
|      \---один из инструментов
|             | makefile             Каждый инструмент имеет свой собственный make-файл.
|      |      + - - - гкн

\---зрители                          Каждый просмотрщик SDK имеет свой собственный makefile,
                                     который определяет, как просмотрщик должен быть построен.
                                     Каждый из просмотрщиков может быть
```

построен в изоляции

```
|
\---один из зрителей
        | makefile
```

5. Инструменты, необходимые для сборки RenderWare

Графика

RenderWare Graphics поставляется со многими необходимыми инструментами сборки — их можно найти в rwsdk/binкаталог.

Для целей сборки RenderWare Graphics предполагается, что на сборочной машине присутствуют следующие дополнительные инструменты.

Компилятор, компоновщик и ассемблер

Они должны быть настроены для поддержки параметров сборки командной строки. Список поддерживаемых компиляторов см. в разделе "makeincl/rwos/<os>/rwcmplr'каталог в древовидной структуре выше. Обычно вашПУТЬ, ВКЛЮЧИТЬиЛИБДля этих инструментов следует соответствующим образом настроить переменные среды.

Библиотеки и заголовочные файлы для конкретных платформ

Это компоненты SDK, поставляемые производителем платформы. Предполагается, что они устанавливаются с использованием конфигурации по умолчанию.

Сборка Playstation 2 SDK на Linux

Поскольку установщик Windows, поставляющий дистрибутив RenderWare Graphics, не сохраняет разрешения Unix, для некоторых инструментов необходимо восстановить разрешения перед сборкой SDK.

Следующие команды, введенные в каталоге rwsdk, восстановят разрешения инструментов:

chmod a+x bin/sed

chmod a+x bin/mkdir

chmod a+x buildtools/findsyms/findsyms

chmod a+x buildtools/incgen/incgen

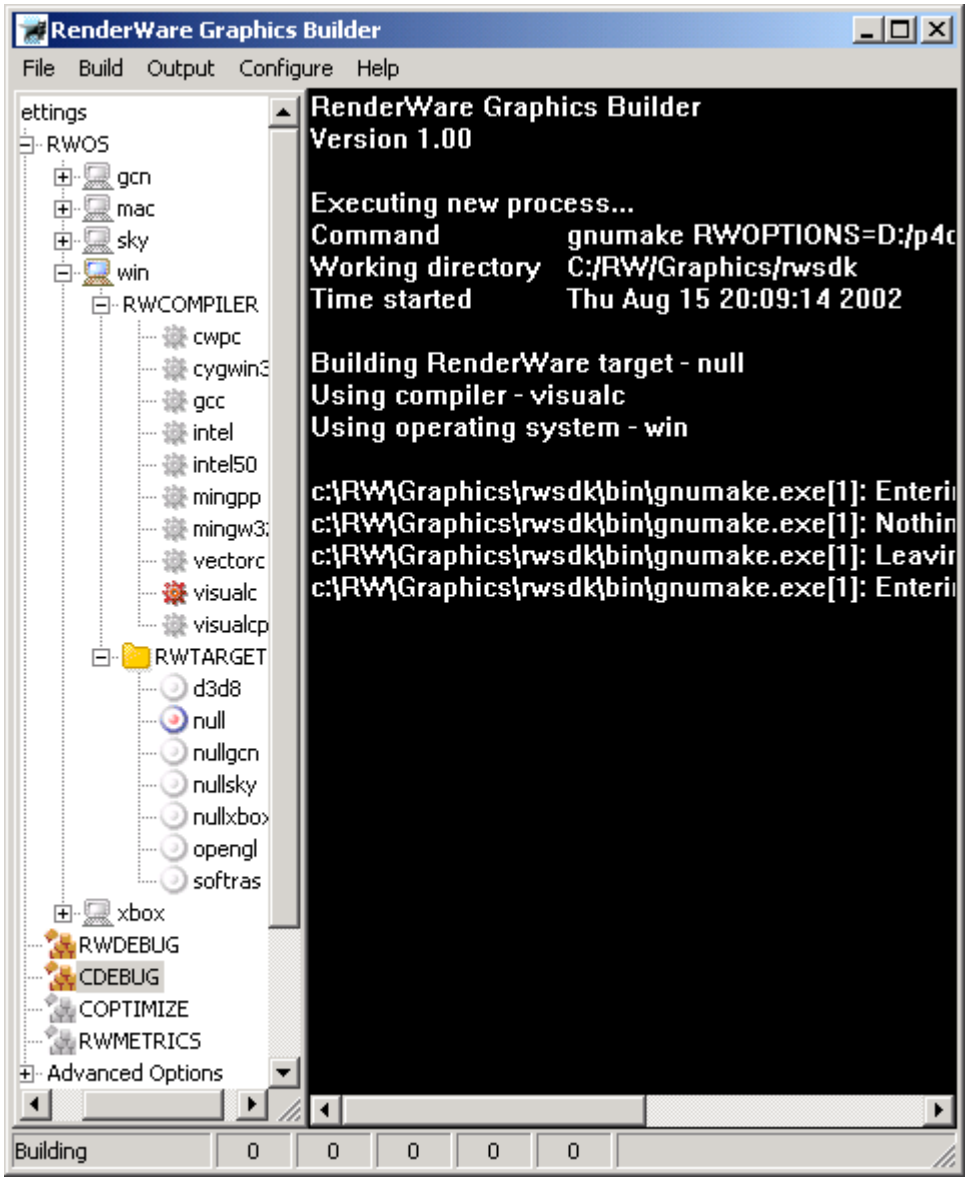
chmod a+x buildtools/inline/inline

6. Графический инструмент для создания RenderWare

Графика

Инструмент GUI для Windows (совместимый с Windows 2000 и выше) поставляется с исходным дистрибутивом RenderWare Graphics. Это графический интерфейс для управления файлами options.mak и сборки RenderWare Graphics SDK.

Этот инструмент позволяет вам создавать SDK, документацию, отдельные плагины или наборы инструментов, а также примеры и просмотрщики, используя те же параметры сборки одним нажатием кнопки. Исходный код примеров и просмотрщиков поставляется с бинарным дистрибутивом RenderWare Graphics.



Для получения более подробной информации обратитесь к руководству пользователя инструмента сборки, buildtool.pdf,вдокументы/инструментыв каталоге RenderWare Graphics.

7. Строительные компоненты RenderWare Graphics SDK

Существуют альтернативные методы создания частей или всего RenderWare Graphics SDK.

- Совет: быстрый способ сгенерировать только библиотеки и заголовки SDK (избегая при этом создания каких-либо примеров, инструментов или средств просмотра) — вызвать командную строку.

  gnumake RWOS=win RWCOMPILER=visualc RWTARGET=d3d8 sdk-libs

  Дополнительную информацию и примеры можно найти в файлах makefile.xbox, makefile.win, makefile.gcn и makefile.sky (PS2).

  ПРИМЕЧАНИЕ: существуют зависимости между компонентами SDK, например, примеры SDK требуют последнюю версию библиотек SDK для успешной сборки. Рекомендуется, чтобы пользователь приобрел расширенный опыт сборки RenderWare Graphics таким образом, поскольку могут возникнуть проблемы с зависимостями сборки. Однако любые проблемы должны быть решены, если пользователь выполнит процесс, описанный в разделе Файл options.mak, или ярлык, описанный ниже.

Обычно разработчики работают только над одной "разновидностью" библиотек в любой момент времени. Эти инструкции описывают, как построить один набор библиотек.

1. Построитьпараметры.makфайл, содержащий параметры для необходимой платформы (см. выше).

- Совет: быстрый способ созданияпараметры.makфайл, содержащий стандартные параметры, должен использовать make-файл верхнего уровня следующим образом

  gnumake RWOS=win RWCOMPILER=visualc RWTARGET=d3d8 options.d3d8

  Подставьте любое из поддерживаемых значений для платформы вместо win, visualc и d3d8 выше. Полученный файл теперь можно переименовать как параметры.maki используется в этом процессе.

2.Укажите путь к этомупараметры.makфайл в среде с использованием RWOPTIONS=c:/mypath/options.mak.

3.Убедитесь, чторwsdk/binкаталог инструментов включен в вашПУТЬпеременная окружения.

4. Перезагрузите компьютер, чтобы убедиться, что переменные среды приняты.

5.компакт-диск /rwdistribution/rwsdk.

6. Введите ''gnumake'.

Theпараметры.makФайл можно редактировать вручную для создания вариаций библиотеки или для создания отдельной библиотеки.

Использование файлов .mcp

Файлы проекта CodeWarrior (MCP) предоставляются с демонстрациями, примерами и просмотрщиками RenderWare Graphics. Это позволяет пользователям CodeWarrior IDE создавать приложения RenderWare Graphics из своей любимой среды разработки и иллюстрирует подходящую структуру для создания новых приложений RenderWare Graphics с использованием CodeWarrior. Цели предоставляются для стандартных конфигураций отладки, выпуска и метрик аналогично другим доступным методам сборки.

Пользователям рекомендуется иметь самую последнюю версию MetroWerks CodeWarrior IDE для своих платформ разработки.

8. Создание полного SDK

Если вы хотите воспроизвести весь процесс сборки SDK, самый простой способ — использовать один из главных файлов управления сборкой верхнего уровня.

например

```
gnumake –i –f makefile.sky
```

создаст PlayStation 2 SDK. –яопция будет игнорировать любые возникающие ошибки, позволяя процессу сборки дойти до завершения.

Изучение этих файлов основной сборки предоставит более подробную информацию о происходящем.

9. Распространенные проблемы сборки

Общие проблемы сборки

Общих проблем сборки можно избежать, если:

- Очистка папок перед пересборкой (удаление неиспользуемых или старых исходников и библиотек)
- Обновление путей таким образом, чтобы были найдены все новейшие платформенно-специфичные SDK
- Убедитесь, что в пути включены самые современные инструменты

конфликты версий cygwin

В процессе сборки используется свободно доступныйcygwinНабор утилит командной строки. Если они уже установлены на машине сборки, то возможен конфликт номеров версий между установленной версией и той, которая поставляется с SDK.

Эти конфликты обычно можно разрешить путем переименования или удаления файла. rwsdk/bin/cygwin1.dllиз дистрибуции.

Версии и обновления SDK для конкретных платформ

Версии комплектов разработки программного обеспечения, предоставляемых для аппаратного обеспечения разработки, должны быть максимально актуальными, чтобы избежать проблем совместимости при сборке с помощью RenderWare Graphics. Текущие версии этих платформенно-специфичных SDK, на которых были построены инструменты разработки RenderWare Graphics, можно найти вжурнал изменений.txtфайл в корневом каталоге RenderWare Graphics.

Не удалось найти SDK платформы

Если компоненты SDK, специфичные для платформы, не были установлены в местоположении по умолчанию, то процесс сборки не сможет их найти. Это можно решить следующим образом:

- указав правильный путь в вашемпараметры.makфайл
- при сборке всего SDK, редактирование makefile верхнего уровня для указания правильного местоположения

Сборка PC SDK ищет компилятор CodeWarrior, однако он не используется

Чтобы предоставить набор библиотек, работающих со всеми компиляторами ПК, RW SDK собран с компилятором Microsoft Visual C/C++. Это создает библиотеки, совместимые со всеми компиляторами — за одним исключением. PNG toolkit, который использует общедоступный код, необходимо пересобрать с CodeWarrior, чтобы предоставить библиотеку, совместимую с компилятором CodeWarrior. Если поддержка CodeWarrior не требуется, то любые такие ошибки можно смело игнорировать.

10. Расширенные возможности

В этом разделе описываются некоторые из более продвинутых методов, которые используются командой разработчиков CSL. Они предоставляются только для информации и могут не поддерживаться во всех конфигурациях.

Пересборка одного плагина

Как было кратко упомянуто выше, можно перестроить отдельный плагин или инструмент в отрыве от основной сборки SDK. Для этого вам необходимо сделать следующее:

1. Создайте версию SDK, которая в настоящее время используется для разработки.

2. компакт-дискrwsdk/plugin/anyplugin.

3. Типgnumake.

- Для этого потребуется, чтобыпараметры.makфайл указывается с помощьюВАРИАНТЫ переменная окружения.

. ipp-файлы

В файлах сборки SDK есть дополнительное правило, которое поддерживает генерацию файлов с расширением .иппрасширение. Эти файлы содержат результат препроцессорной стадии компилятора.

Иногда бывает полезно проверить предварительно обработанную/расширенную версию файла, чтобы иметь возможность отладки разделов кода, состоящих из макросов.

Простой пример создания .иппфайл показан ниже:

```
cd rwsdk/plugin/spline
gnumake rpspline.ipp
```

Файлrpspline.ippсодержит предварительно обработанную версиюrpspline.c