



Визитка

*ЮРИЙ ЯЗЕВ, ведущий программист компании GenomeGames, специализируется на разработке графических приложений, БД и всем том, что нужно для MMO-игр*

## Реализация вывода штрих-кода в Oracle Application

Часто возникают потребности в реализации новых решений в условиях морально устаревшего ПО. Подобный случай произошел в компании, где я работал

Это крупная металлургическая корпорация, где любое изменение в наборе ПО тщательно обсуждается и случается довольно редко, поскольку вызывает большой резонанс среди пользователей. Но в определенный момент было решено добавить к выводу отчетов возможность печати штрих-кода. Сердце инфраструктуры компании представляет СУБД Oracle, следовательно, все остальные программные продукты (в данном случае подразумевается ПО для создания отчетов) – это Oracle Reports, Oracle Forms и т.д. Кроме того, в интрасети компании функционирует Oracle Application Server, поэтому отчеты надо было представить в веб.

### Возможные решения

По меньшей мере можно предложить два варианта решения: использовать специальный шрифт, отображающий символы штрих-кода, и растровые изображения, содержащие «забор» разной ширины. Как бы ни был привлекателен второй вариант, его использование связано с большими трудностями, заключающимися в подгонке и составлении результирующего «забора». Чтобы построить штрих-код с помощью этого способа, надо разработать замысловатую программу, которая бы из разных изображений, представляющих прямоугольники разной ширины, составляла ряд, разделенный строго выверенными промежутками. Можно однозначно утверждать, что это не наш выбор.

Если выбрать вариант с использованием специального шрифта, тогда нужно определиться: использовать коммерческое или свободное решение? У каждого варианта есть свои плюсы и минусы, однако второй вариант оказался предпочтительнее не только потому, что он бесплатен.

В случае если вы выберете коммерческое ПО, удобное решение предлагает компания IDAutomation. Она предлагает множество решений подобного рода, так сказать, для любого случая, в том числе для Oracle есть Java-решение. Однако в компании, нуждавшейся в реализации возможности штрих-кодировки, использовалась старая версия Oracle Reports, в которой не было поддержки Java.

Таким образом, наш выбор пал на свободное ПО. На сайте <http://tradepilot.ru> можно прочесть стандарты по штрих-

кодам. Скачать свободно распространяемый шрифт для штрих-кодировки можно с сайта <http://grandzebu.net>. Кроме того, на последнем имеется много дополнительной информации и приложений для штрих-кодировки. Шрифты для штрих-кодов распространяются в обычном для Windows формате шрифтов TrueType (\*.ttf).

В нашем случае понадобились два шрифта и соответственно два файла: Code.128.ttf и Code39.ttf.

Для реализации вывода штрих-кода в создаваемых отчетах Oracle Report на рабочих компьютерах программистов потребовалось только поместить новый шрифт в папку Windows\Fonts (при этом пока опустим необходимость включения в результирующий набор символов старт-, стоп- и контрольный символы). Однако при попытке выложить отчет, содержащий штрих-код в Oracle Application, в веб-интерфейсе вместо штрихов и промежутков мы получили пустые квадратики, знаменующие отсутствие символов данного шрифта.

Установив гарнитуру на все компьютеры организации, имеющие доступ к веб-интерфейсу Oracle Application, мы все равно ни на одном из них не получили желаемый результат. Стало ясно, что гарнитура штрих-кода отсутствует на сервере, где генерируются отчеты для веб-интерфейса.

Поскольку на этом предприятии используется распределенная база данных Oracle, связанная с остальными компаниями внутри корпорации, а сервер установлен в другом городе, доступа к нему мы не имели. Тем не менее, связавшись с администратором удаленной компании, мы узнали, что там используется сервер с HP-UX. Эта новость была не совсем приятной, поскольку, как добавил администратор, во включенной в ОС сборке FreeType шрифты формата \*.ttf не поддерживаются (отсутствует поддержка правильного хинтинга TrueType-шрифтов (<http://ru.wikipedia.org/wiki/FreeType>)).

### Шрифты

Перед нами встала новая задача: преобразовать шрифт из формата ttf в поддерживаемый HP-UX формат Type1. После непродолжительных поисков я остановился на свободно распространяемой утилите ttf2pt1 версии 3.4.4 для Linux, по-

сколько свободных программ подобного рода для Windows найти не удалось.

Чтобы развернуть Linux на своей рабочей Win-машине, установил бесплатную виртуальную машину Oracle Virtual Box 3.2, а под нее – Ubuntu Linux. Затем настала очередь установки утилиты конвертации шрифтов ttf2pt1, ради которой и затевалась вся эта операция с Linux! Для этого перейдя в консоли в папку с этой программой, я ввел команду:

```
make install
```

После того как программа была успешно установлена, можно было использовать ее широкие возможности, которые оказались действительно впечатляющими. Замечу, что это утилита командной строки, и она имеет широкий набор дополнительных флагов (см. рис. 1).

В моем случае понабилось использовать только два. Таким образом, команда для конвертирования шрифта из формата ttf в формат Type1 выглядит следующим образом:

```
ttf2pt1 -a -b Code_128.ttf Code128
```

Где флаг -a означает включение из источника всех символов для конвертации, а флаг -b означает, что нужно создать двоичный файл с расширением \*.pfb и закодировать информацию в него. Кроме файла pfb, нужен файл формата afm, в котором хранятся метрики символов, он создается по умолчанию.

В итоге, получив эти два файла, мы отправили их администратору сервера, чтобы он подключил их в систему.

## Неожиданные проблемы и их решения

После этого удалось получить вывод шрифта штрих-кода в отчетах в веб-интерфейсе. Тем не менее, хотя штрих-код имелся, символы были закодированы некорректно. Чтобы исправить эту проблему, я написал на PL/SQL хранимую в БД Oracle функцию для кодирования набора цифр в правильный штрих-код.

Если коротко описать действия, выполняемые этой функцией, то она вставляет «старт»- и «стоп»-символы в строку, состоящую из переданных в качестве параметра цифр, подсчитывает контрольный знак этих цифр и вставляет его перед символом «стоп». На выходе функция выдает результирующую строку – закодированную последовательность цифр, которую можно вывести в качестве штрих-кода. За более подробной информацией о представлении и кодировании штрих-кода обращайтесь по указанным выше ссылкам. Приведу исходный код этой функции (см. листинг 1):

Листинг 1. Функция для преобразования набора цифр в формат, пригодный для вывода штрих-кода по стандарту Code 128

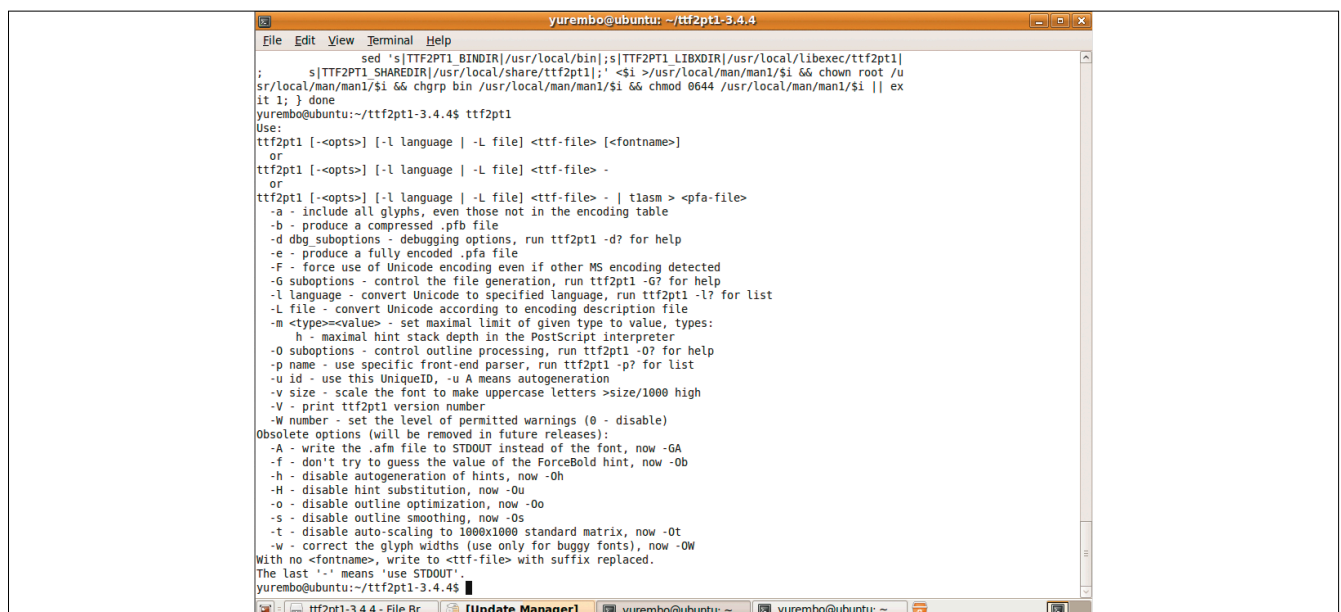
```
FUNCTION GET_BARCODE -- by yurembo
(num IN varchar2)
RETURN varchar2
IS
type array_numbers is table of integer;--Массив
--Инициализация массива
nums array_numbers:= array_numbers();

type string_array is table of varchar2(255);
mas string_array := string_array();

result varchar2(255);--Возвращаемый результат
--Преобразованное в строку число - параметр
chnum varchar2(255);
symbolcount Integer;--Количество символов
controlsymbol varchar2(255);--Контрольный символ
--Стартовый код для таблицы C
start_code constant integer := 108;
--Завершающий символ
stop_code constant integer := 109;
divider constant integer := 103;--Делитель

s varchar2(10);--Промежуточное значение
--В процессе программы изменяется (см. комментарии
--ниже)
j Integer;
--Индикатор: четное или нечетное количество
--символов: если false, то количество символов
--четное, если true, то нечетное
b boolean := false;
```

Рисунок 1. Описание флагов утилиты



```

        str varchar2(255) := '';
BEGIN
    chnum := TO_CHAR(num);
    symbolcount := Length(chnum);
    --Нечетное количество символов?
    if mod(symbolcount, 2) != 0 then
        chnum := '0' || chnum; --Тогда добавляем лидирующий ноль
        b := true; --Помечаем, что количество символов нечетное
    end if;

    s := '';
    j := 1; --Счетчик для массива чисел
    for i in 1 .. Length(chnum) loop
        s := s || substr(chnum, i, 1);
        if (Length(s) = 2) or ((b = true) and (Length(s) = 1)) then
            nums.extend;
            nums(j) := TO_Number(s) * j;

            mas.extend;
            mas(j) := chr(TO_Number(s));
            -----
            if b = true then
                b := false;
            end if;
            -----
            j := j + 1;
            s := '';
        end if;
    end loop;

    j := start_code; --Сумма
    for i in nums.first .. nums.last loop
        j := j + nums(i);
    end loop;

    controlsymbol:= mod(j, divider); --Контрольный символ

    for i in mas.first .. mas.last loop
        str := str || mas(i);
    end loop;

    chnum := chr(start_code) || str || chr(controlsymbol)
    || chr(stop_code); --Все объединяем в одну строку

    --chnum := start_code||' '||chr(start_code);

    result := chnum;
    RETURN result;
    --EXCEPTION
    -- WHEN other THEN
    --     statements ;
END;
```

Добавив эту функцию в БД, вместо корректного вывода мы получили не очень впечатляющий результат: вместо некоторых символов отображались квадратики, обозначающие отсутствие данного символа в используемом шрифте. Эта проблема привела нас в замешательство.

Потратив много человеко-часов на ее решение, нам только удалось выяснить причину этого инцидента. По определенным причинам мы вынуждены использовать Oracle Report 6.0, выпущенный в 1999 году, который поддерживает для вывода только первые 128 символов из таблицы кодировки (ASCII), но он не выводил верхние 128 символов, что было для нас необходимо. Затем, исследовав шрифт Code 128, я убедился, что проблема действительно в этом, и часть символов выходит за границу первых 128 символов.

## Надежда умирает последней

Уже почти потеряв надежду на успешный результат, я продолжил исследования, заключающиеся в поиске и реализации другого шрифта штрих-кода. Результатом этих изысканий стало мое предложение об использовании шрифта

Code 39, который вмещал все символы в первую половину таблицы кодировки. Получив одобрение своего решения, я приступил к его реализации.

Важно, чтобы ваша модель сканера поддерживала выбранный шрифт штрих-кода. Замечу, что современные сканеры поддерживают подавляющее большинство всевозможных шрифтов.

Уже зная путь к решению, мне не составило труда добиться его. Во-первых, на языке PL/SQL я написал хранимую процедуру для преобразования набора символов в закодированный штрих-код, в случае со шрифтом Code 39, она получилась короче (см. листинг 2):

Листинг 2. Функция для преобразования набора цифр в формат, пригодный для вывода штрих-кода по стандарту Code 39

```

FUNCTION GET_BARCODE39(num in varchar2)
return varchar2 is -- by yurembo
    result varchar2(255); --Возвращаемый результат
    --Преобразованное в строку число - параметр
    chnum varchar2(255);
    controlsymbol varchar2(255); --Контрольный символ
    symbol constant char := '*'; --Старт/стоп-символ
    divider constant integer := 43; --Делитель
    summa Integer := 0;

begin
    chnum := TO_CHAR(num);

    for i in 1 .. Length(chnum) loop
        summa := summa + to_number(substr(chnum, i, 1));
    end loop;

    if mod(summa, divider) != 0 then
        controlsymbol := chr(mod(summa, divider));
    else
        controlsymbol := '0';
    end if;

    result := symbol || chnum || controlsymbol || symbol;

return result;
end;
```

Затем в Linux командой:

```
ttf2pt1 -a -b Code39.ttf Code39
```

я преобразовал шрифт в формат, пригодный для сервера HP-UX, и отправил результат системному администратору.

Для экспериментов я применил следующий SQL-запрос, использующий мои функции:

```

select ACCOUNTING_EVENT_ID, EVENT_TYPE_CODE, ␣
       ACCOUNTING_DATE, get_barcode(ACCOUNTING_EVENT_ID) F, ␣
       get_barcode39(ACCOUNTING_EVENT_ID) V
from ap_accounting_events_f1
```

После того как системный администратор подключил шрифт, мы смогли наблюдать корректный вывод штрих-кода в наших отчетах!

\*\*\*

Таким образом, выбрав свободное решение с помощью разработанного на его основе приложения, удалось достичь приемлемого результата, отвечающего требованиям предприятия. И теперь, надеюсь, после изложения варианта решения на бумаге им смогут воспользоваться на других предприятиях, где встанет вопрос реализации вывода штрих-кода в условиях устаревшего ПО. **EOF**